UNIVERSITY OF GHANA

COLLEGE OF BASIC & APPLIED SCIENCE

DYNAMIC ROUTE MAINTENANCE SCHEME FOR AODV ROUTING PROTOCOL USING JOINTNODES

BY

OTCHERE, ISAAC NYAMEAMAH
(10508872)

THIS THESIS IS SUBMITTED TO THE UNIVERSITY OF GHANA, LEGON IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF MPHIL COMPUTER SCIENCE

JUNE, 2018

DECLARATION

I hereby declare that this thesis is my own original research work undertaken at the Department of Computer Science, University of Ghana, Legon under the guidance of my thesis supervisor except for other people's work which have been duly cited and acknowledged.

STUDENT

Name: Otchere, Isaac Nyameamah

Signature: .....

Date: .....

SUPERVISOR

Name: Dr. Jamal-Deen Abdulai

Signature: .....

Date: .....

CO-SUPERVISOR

Name: Dr. Ferdinand A. Katsriku

Signature: .....

Date: .....

DEDICATION

To my family.

ACKNOWLEDGEMENT

I wish to express my profound gratitude to my creator, God Almighty, granting me wisdom and knowledge, and guiding and protecting me throughout the writing of this thesis successfully.

I also wish to extend my profound gratitude to my supervisors Dr. Jamal-Deen Abdulai and Dr. Ferdinand A. Katsriku for their support and guidance throughout my studies and thesis work. May the Almighty Lord bless and replenish all that they have lost in the course of executing their duties.

My next gratitude goes to the entire staff of the Department of Computer Science for their indispensable help and support in making my thesis a success.

Finally, to Agbesi Collinson Colin, Akama Samuel, Asiedu Bismark and all my colleagues in the Department who helped in diverse ways to make this work a success, I say God richly bless you and grant you all the desires of your hearts.

ABSTRACT

Ad hoc On-demand Distance Vector (AODV) is one of the widely deployed routing protocol for mobile ad hoc network (MANET). Through route discovery process, AODV establishes a route timely for data transmission without prior knowledge of the network topology. However, how to maintain this route while data transmission is ongoing is a problem due to the mobility and the limited resource nature of the nodes in this type of network. Frequent changes in an ad-hoc network topology require a routing protocol, which is more adaptive to topological variations. This thesis proposes a dynamic route maintenance scheme which is more adaptive to the topological variations in MANET.

In this route maintenance scheme, MANET benefits from a carefully selected neighbour, herein known as *JointNode,* to form an alternative link for the individual node-to-node links within a route in AODV routes maintenance phase. Nodes use the one-hop hello messages to build this neighbour knowledge. The scheme identifies a local *JointNode* as a backup link to dynamically resort to as an alternative or imperative options. Nodes can therefore configure an already established route to become a refined route dynamically without necessarily resorting to the route rediscovery process.

The thesis demonstrates through network simulation using NS-2 that AODV routing protocol can benefit from the proposed route maintenance scheme. The simulation result shows a significant increase in the network throughput and packet delivery ratio and a decrease in both delay and nominal route overhead.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1. Background of the study

Mobile computers and smart devices have become very pervasive and prevalent in society. In recent times, these devices have become so convenient to be carried with us as we move around with wireless networks providing an integral network for them to stay connected everywhere and anytime. Wireless communication provides a mobile supporting network. It has a number of advantages that includes cost efficiency and instantaneous setup anywhere and anytime over the wired counterpart. This has almost relegated wired networks to providing only a backbone service for these mobile supporting networks.

Wireless communication networks comprise of a variety of technologies and various deployment approaches. There are two broad approaches for wireless deployment: infrastructure-based wireless networks and infrastructureless-based wireless network. The infrastructure-based wireless network, in its simplest form, has a fixed Access Point (AP) with associated mobile stations. Examples of such wireless network include WiFi network, the cellular based network like the GSM, etc. In this type of wireless deployment, all communications are routed via the centralised AP, making each mobile station just one hop away from the AP. No other node exists between a mobile station and the AP. The APs are usually well planned and resourced. AP may also serve as a backbone via which the mobile stations can connect to the wider network.

Practically, it may not be possible to have such a well-planned wireless coverage everywhere or every time. For instance, it is practically impossible to build such a

network in a battlefield. Again, in disaster circumstances, these installations can be destroyed. Yet, communication in such situations mentioned above are very crucial. The solution to the afore mentioned challenges of the infrastructure-based network is infractructureless-based wireless network. It offers a decentralized characteristic network. It is popularly known to as Mobile Ad hoc NETwork (MANET). Mobile Ad hoc NETwork (MANET) applications in recent past includes; battlefield, disaster/crisis management, community, education, health, coordination/collaborative work, multimedia applications, etc.

MANET is a self-organizing and self-configuring "multi-hop" network which does not require any fixed installations. Mobile stations/nodes are dynamically and arbitrarily located and require that packets are relayed to other nodes in order to deliver data across the network. This network, therefore, consist of several mobile nodes acting as terminals and routers at the same time and communication between two nodes (i.e. source and destination nodes) may be direct or through other nodes in the network acting as routers. Thus, a source-destination route may be defined by sets of links formed by nodes between the two communicating pairs. This type of network requires no predefine structure and no centralised network administration. This decentralization makes the networks more flexible and more robust.

Unlike a well-planned and resourced AP networks that have much larger coverage, individual handheld mobile nodes may not be such resourced because of form factor. The individual nodes may be limited to some constraints; limited bandwidth, wireless radio transmission range, nodes are mobile with no fixed motion and boundary, nodes' energy are only limited to battery power, etc. These limitations place some restrictions on the use of the resources which make finding an efficient routing protocol for MANET very challenging. The basic routing protocols need to be modified to meet

the unique characteristics of mobility and the other constraints to solve the problem of routing in MANET. A good number of research has proposed routing protocols for MANETs (Johnson, Hu, & Maltz, 2007; Perkins & Royer, 2003; Haas & Pearlman, 1998; Perkins & Bhagwat, 1994).

Among the routing protocols is Ad-hoc On-demand Distance Vector (AODV), which is one of the most popular MANET routing protocols (Cerri & Ghioni, 2008). As stated in the RFC3561, "AODV can handle low, moderate, and relatively high mobility rates, as well as a variety of data traffic levels" (Perkins & Royer, 2003, p. 6). Broadcasting remains the fundamental and effective way of disseminating route discovery information and other network services in MANET. Flooding is the broadcast scheme used in AODV. Meanwhile, flooding can be extremely costing. It can lead to a serious network problem, the *broadcast storm* (Sooriyaarachchi, Fernando, & Gamage, 2016; Abdulai, Ould-Khaoua, Mackenzie, & Mohammed, 2008).

AODV is intended to decrease the dissemination of control traffic and eliminate overhead on data traffic keeping in mind the goal of enhancing the network's scalability and performance by broadcasting request packet only on-demand (Perkins & Royer, 1999). Yet the dissemination of the control packet at the route discovery phase poses a great challenge especially in an environment where the route topology changes rapidly. A lot of efforts has been proposed to minimised these disseminations. The efforts are geared in the following basic aspects of the AODV protocol: (i) the way the discovery process blindly floods the network with the route request messages, e.g. the probabilistic based broadcasting (Agarwal, Govil, & Sinha, 2016; Abdulai et al., 2008) and (ii) the frequent route rediscovery as a result of route breaks (Sharmaa, Patraa, & Kuma, 2016; Abhilash, Perur, & Iyer, 2002). This thesis belongs to those

that work on the later. It seeks to establish a long-live source-destination route by preventing frequent route breaks through a route maintenance scheme.

Routes are rediscovered whenever they are lost prematurely, that is before a source finishes sending its packets to the destination. The cost of route breaks can be significant – cost for detecting the disconnection and the cost for re-establishing it again. Route stability is a major factor that affects network performance. Apart from the network being flooded with rediscovery packets which may create a contention problem, link break in AODV also increases the number of packet drops and delays (Bisengar, Ouadoudi, Nourddine, Mohamed, & Mohamed, 2012).

Many approaches have been proposed to solve or minimise this problem of frequent route breaks, yet none of them guarantees the lowest bound. Several attempts like caching of learned routes (Kawish, Aslam, & Khan, 2008), use of location information (Raich & Vidhate, 2013; Latiff, Ali, Chia-Ching, & Fisal, 2005), and the use of virtual backbones (Wu, 2011; Lai, Lin, & Yuh-Chung, 2007; Lee & Gerla, 2000) have been proposed to maintain source-destination routes that are found. As a means of maintaining source-destination routes, frequency of route search is minimised. For a network like MANET whose topology changes frequently, route rediscovery will heavily increase the overall overhead. Therefore, establishing a long-live source-destination route is not a bad idea. It will minimise the number of route reestablishment.

This thesis therefore seeks to reduce frequent route breaks and hence reduce route rediscoveries using backup routing. To establish a long-live source-destination route, virtual backbones create a local redundant information for re-establishing the individual links locally whenever the existing link goes bad. The unnecessary frequent global dissemination of control packet for route re-establishment must be avoided.

Current implementation of local backup repair process (Sharmaa et al., 2016; Lai et al., 2007; Lee & Gerla, 2000) is time and traffic demanding, and at times wrong and stale neighbours are selected to repair route which eventually fails. Moreover, Perkins & Royer (2003) stated that failure of route repair is even more severe.

It is proposed in this thesis that the hop-to-hop linked nodes identify neighbour nodes that they share in their 1-hop neighbour (herein referred to as *JointNode*) and keep them as backup routes. When a node notices a link failure, they can select from the pool of that link's *JointNodes* to locally repair the link without waiting for rediscovery of this route entirely. The motivation for this thesis work is the fact that the proposed *JointNode* sets come in handy to quicken a local repair process and, as well, it can pre-determine whether such local repair exists in advance. Should this *JointNode* set be empty for a given node-to-node link, there is no need to waste time for local route repair than to signal source node to rediscover the route as local repair is likely to fail. Thus, these local *JointNodes* increase the redundancy of the network links to speed up link repairs to prolong the lifetime of a discovered route and, overall, improves the performance and efficiency of the basic AODV protocol.

1.2. Statement of the Problem

Ad hoc on-demand networks source-destination route should not break frequently for it to guarantee a Quality of Experience (QoE) for their applications. To achieve a long-life route and end-to-end QoE for ad hoc applications, the issue of premature route break should be carefully addressed while developing routing schemes. It is expensive to find a route between a communicating pair of nodes in MANETs in terms of bandwidth and packet latency (Abdulai et al., 2008). So node-to-node link failure should not render the entire route unusable should there be an alternative way to repair the route. Therefore, route failure and rediscovery caused by nodes' mobility during

the lifetime of a given source-destination nodes should be minimized if it cannot be completely eliminated.

Mobile nodes have non-uniform mobility and make the topology dynamic. Yet active route of on-demand protocols should be able to dynamically adjust itself accordingly. To achieve this, individual node-to-node link should be able to manage the links and adjust them when need arises without the active involvement of the source and destination nodes. The intermediate nodes should be able to reconstruct when broken paths so that the source and destination nodes will not notice the occurrence of communication link failure. Perkins & Royer (2003) believe repairing a link locally may be less time consuming than to do a global search beside the less demand of traffic. They estimated that it will take two times the time to travel between two nodes plus the queue and process time which are far less than a global search that takes the network traverse time. Thus, apart from the demand of resources like bandwidth in discovering a new route, it also introduces significant packet delivery latency, which is the time required to transmit packets from source to destination nodes.

Unfortunately, the current implementation of route maintenance in AODV routing protocol degrades the overall network throughput and packet delivery latency. The route maintenance process is associated with a large amount of communication overheads resulting from the exchanged of large volume of network control packets. This work proposes a novel algorithm which reduces the route failures resulting from link breaks, thereby increasing the life-span of routes. This is achieved by merging broken links using a special node called *JointNode*. Although similar approaches have been proposed by (Sharmaa et al., 2016; Lai et al., 2007; Abhilash et al., 2002; Lee &

Gerla, 2000), their methods are associated with increased end-to-end delay and increased use of stale neighbours to repair a broken link.

Using already existing mechanisms in AODV to improve the performance of the routing protocol during active route lifetime is prime. The study uses the hello messages of AODV to identify a neighbour node that is shared by two nodes forming a link, herein referred to as *JointNode,* and create an alternative route table which contains these *JointNodes* that are known to link the two nodes on an active route and therefore use such node as a bridge to re-join that link should it fail. This study is similar to AODV modified route maintenance scheme, MRM (Sharmaa et al., 2016) and AODV Adaptive Backup Routing, ABR (Lai et al., 2007). However, it differs from ABR as upon route break does not have to trigger rediscovery but rather repair by selecting from the *JointNode* set to merge the failed link again. It also differs from MRM in the way they select members of the backup link nodes (or *JointNode* set as called herein). In addition, this thesis also suggests a way to proactively change a link to make other gains such as hop-gain, anticipated link failure, energy, traffic congestion, etc. This proactive link management scheme improves the latency incurred due to route breaks and rediscovery operations and thus prolong the route life time. Although, the proposed algorithm optimizes the overhead associated with the route maintenance phase throughout the lifetime of the route, it does not require a complete overhaul of the existing AODV which makes it practically applicable.

1.3. Aim

The aim of this study is to minimize route breaks and prolong the lifetime of AODV active route to improve network performance metrics for a decentralised ad hoc network. To enable a typical non-fixed topology network to fully adapt to the continuous dynamic topology of MANET, route maintenance algorithm must be

designed to dynamically stay connected, even as topology changes. In order to improve the performance of communication links across such a dynamic network, the communication links must be able to adjust to the dynamics of the network topology. To do so, the traditional "break and rediscover" phase of the route is modified. As the nodes roam from one area to another, communication links break due to the limitations of the wireless coverage ranges and also some nodes which could not reach each other directly may also come into direct coverage. This study designs a route maintenance scheme which hide these effects from the communicating parties by modifying the communication links locally.

1.3.1. Specific Objectives

Specifically, the study seeks to:

i. Identify and maintain *JointNodes*.

ii. Repair a broken link within an active route locally by selecting a best node from the *JointNode set* to bridge lost links.

iii. Eliminate redundant link or links in a given active route whenever an upstream node notice that it can reach out to a downstream node directly without the need of its intermediate neighbour or neighbours.

1.4. Justification of the Study

Most route discovery algorithms of MANETs are established to favour nodes that are far apart. The algorithms ensure either of the following conditions: the established route is (i) the shortest path to the destination, (ii) the least hop count, (iii) makes the minimum of (re)broadcast in discovering the route. In either way of choosing best routes, nodes selected are likely to be sitting at the fringes of each other's coverage range. These individual nodes sitting at the fringes are likely to fade due to mobility.

However, finding a route between two nodes in MANET is a costly activity. It is therefore in the right sense to safeguard a return route to avoid repeating this expensive operation frequently. Protecting routes from failing is very important.

MANETs have highly dynamic topology as the motion of the individual nodes in the network has no definite pattern. Besides, nodes distribution is often random and a node can leave or join the network at any point in time. Therefore, a maintenance mechanism that does not require the global network information but rather reflects the local topological characteristics of a given node-to-node link can dynamically adapt to the local changes is recommended. Taking cognizance of the fact that route search cost largely affects network, the recommendations and suggestions from this research when implemented will help AODV and all matching ad hoc routing protocols to achieve an improved quality of the network. This will as well provide QoE to the ever increasing mobile devices and MANET applications.

The results of the study also overcome the problems of unnecessary delays as a result of the route maintenance. It is a quick mechanism to heal from a broken link (whenever next hop is unreachable) in the route to restore data transmission.

1.5. Significance of the Study

The findings of this study will contribute to the development of MANET considering that it plays an important role in communications and technologies today. The increasing demand in MANET applications in recent past and the pervasiveness of mobile and smart devices justify the need for more effective routing techniques in the ad hoc environment. Thus, applications that demand various quality of services that are finding their way to this non-traditional network can seamlessly be handled.

For researchers, the study will help uncover the critical area in ad hoc on-demand protocol where many research have been unable to explore. The study may arrive at a new way of maintaining source destination route in AODV route maintenance phase. The research work will also serve as a reference material to the academia for future research work.

1.6. Scope of the study

This study proposes a route maintenance scheme for ad hoc on-demand protocol, specifically AODV. Research over the years mostly focused on the route discovery process and how the routing could return the destination with the minimum committed overhead. Moreover, those that are committed to how a route can be maintained mostly use an inaccurate neighbour to repair lost links. More specifically, this study focuses on how an active route can be maintained when a link within the route fails without having to abandon this route entirely to seek for new fresh route. How AODV returns a route on a source-destination route request is not considered here in this study. Besides the study will not look into what caused the link failure though it is assumed to be the nodes' mobility. This study shall be limited to simulation using Network Simulator 2 (NS2).

1.7. Limitations of the Study

Due to financial and time constraints, this study could only be tested using computer simulation. Testbeds needed to help evaluate the proposed scheme experimentally could not be purchased due to lack of fund. Besides it would demand a great deal of time to configure a large set of these testbeds for such a large set of network scenarios.

Moreover, the number of nodes and the maximum speeds of the nodes used were limited 200 nodes and 35 ms^{-1} respectively. Running the NS-2 simulator for a large

number of nodes and a relative high mobility is computationally intensive. Due to limited hardware resources, the maximum number of mobile nodes that could be ran successfully for the testing of the proposed system was 200 nodes. It could not support a relatively large network size and a very high mobility.

1.8. Organisation of the Study

This thesis is organized as follows. In Chapter two, we present the theoretical background of MANET, AODV and the various proposed route maintenance schemes of AODV. Chapter three details the proposed approach of route maintenance and simulation model. The results and analyses of the simulation are discussed in Chapter four and finally, Chapter five concludes the thesis and presents discussions on future research directions.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

This chapter presents the theoretical framework of the study. Mobile ad hoc network and AODV routing protocol with the various route maintenance schemes are reviewed.

2.2. Mobile Ad-hoc Network

Ad-hoc network, a technology that was initially developed for military applications, like battlefield communication in an unknown territory where an infrastructure network is virtually impossible to have or maintain, now provides a relief to many modern internetwork demand. For its ease and simplicity of deployment, a decentralised wireless network, MANET, has become very demanding in modern times. This system requires no well-planned and positioning of base station like centralised networks. The challenge in MANET, however, is routing owing to the fact that entities of MANET are free to move randomly changing the network topology rapidly and unpredictably.

A routing protocol sits in the network layer and is in charge of choosing which path an incoming packet ought to take. Thus, it is a set of rules that find the pathway for packets in transmission from the originating node to their destination. Because of the dynamic nature of MANET caused by the mobility of its nodes traditional routing protocols intended for wired networks cannot be used for such ad hoc network (Ahmed & Khalifa, 2017). The authors believe that in ad hoc networks a node has to know at least the reachability information to its neighbours for deciding a packet route and must adapt to the network topology which can change quite often. Again,

discovering route to destination can also be a large search when the number of nodes in the network is substantial. The search frequency will also be more should mobility be high as node to node link connectivity loss will occur frequently. Besides, nodes usually have limited resources; they are usually battery powered (energy), limited processor, storage constraint, etc.

In MANET, mobile nodes communications are routed from one node to another. To each other, packets are forwarded from a source through intermediate nodes to a destination. Thus, nodes in MANET may act as (i) end-points (source or destination node) and sometimes act as (ii) routers to forward packets in this multi-hop environment. The core mandate of all MANET protocols is to design a dynamic routing protocol for this multi-hop environment that can effectively set up routes for data packets to be sent between nodes with less communication overhead while guaranteeing high throughput and low end-to-end delay (Akansha & Sharma, 2016). Because MANET topology can change rapidly and unpredictably, the protocols must make them dynamically autonomous so that mobile nodes can organize themselves arbitrarily via the wireless links (Abdulai et al., 2008; Hoebeke, Moerman, Dhoedt, & Demeester, 2004). The low resource nature of the network entities also imposes severe restrictions which demand that they are efficiently and optimally utilized. Due to this, MANET routing protocols have therefore attracted a lot of academic attention that is aimed at achieving routing stability for such network.

2.2.1. MANET Applications

Due to the proliferation of small and inexpensive wireless communicating devices, both the academia and industry have now taken advantage of the MANET field. Today, some day-to-day applications, are deployed in this specialised network.

Several comprehensive overviews in the application field have been written in Alslaim, Alaqel, & Zaghloul (2014) and Hoebeke, et. al., (2004). Typical MANET applications include the following:

Military applications

Nowadays it is common to find military equipment containing some sort of wireless connectivity. With rapidly deployment and establishment of networks done via MANET, real-time data, voice and video communications are enabled. The military takes advantage of the feasibility of ad hoc network technology to communicate between the soldiers, vehicles and equipment for reconnaissance, intelligence and other tactical operations in battlefield (Kant, et al.; Ball, Qela, & Wesolkowski, 2011).

The potentialities of connecting moving smart objects to the internet using MANET technologies has become another area of prospect. For instance, the concepts of MANET are very crucial multi-UAVs (unmanned aerial vehicles) systems which are now noted to have advantages of scalability, survivability, cost and speed to complete a mission over a single large UAV. In this area, vehicular ad-hoc networks (VANET) and flying ad-hoc networks (FANET) have emerged. They are specialized MANETs (Ball, Qela, & Wesolkowski, 2011). They focus on the multi-UAV systems with VANETs focusing on land moving vehicles nodes and with FANETs as a specialised form of MANET that addresses the concept of flying mobile nodes (Bekmezci, Ozgur, & Samil, 2013).

Emergency services:

Emergency and rescue operations may take place at where established infrastructure-based communication networks are damaged by a natural disaster (earthquake, fire, flood, etc.) or may just be unavailable. In such situations a rapid

deployment of a communication network is needed to coordinate the activities of the rescue team. Immediate deployment of MANETs in these scenarios may assist rapid activity coordination.

Commercial and civil applications:

MANETs can be used to create an instant and temporary autonomously links for multimedia network using notebook and handheld computers to distribute and share information among participants in a conference or classroom. In fact, our day-to-day applications such as web services are sometimes deployable in this special network via gateway (Zhuang, Liu, & Liu, 2009; Truong & Dustdar, 2012; Li & Yang, 2016). Another appropriate local level application can be in smart home networks where devices can communicate directly to exchange information. Personal Area Network (PAN) uses a short-range MANET to simplify the intercommunication between the various mobile devices (such as a laptop, a cellular phone, headset, etc). PAN can then be connected to a larger network. Used in this way, PANs can be seen as an extension of the Internet. Another application environment may include ship-to-ship ad hoc mobile communication (Zhou & Harada, 2012). Moreover, MANET is envisaged to be used to extend cellular network coverage in limited areas in future networks (Asadi, Wang, & Mancuso, 2014; Hoebeke et. al., 2004).

2.2.2. Types of MANET Routing Protocol

It is important to note that MANET can be classified into different types based on different criteria such as devices used or routing protocols employed. It may be divided into homogeneous and heterogeneous – on the bases of the device characteristics (e.g. transmission ranges) of the nodes involved (Bisengar et al., 2012). Homogeneous ad hoc networks nodes have the same transmission range and device characteristics whereas heterogeneous ad hoc network nodes possess different transmission ranges

and/or other characteristics (i.e. networks consist of different wireless mobile devices such as laptops, cell phones, PDAs, and other smart devices).

Routing protocols for MANET mostly assume that wireless links are bidirectional and nodes have identical transmission ranges. However, it is true that increase in heterogeneity of MANETs is substantial as the number of wireless products and applications (Phones, Computers, Smart TVs, UAVs, RFIDs and Wireless Sensor Nodes and Internet of Things) that use the services of MANETs protocols have become more common and prevalent. Asymmetric due to the variation in transmission ranges and devices capabilities of the nodes is now real. This means, in MANET, nodes' characteristics in the network can change at any time; the processing power, transmission radius, battery life and other necessary constraints of each node can be different and must be factored.

Based on the structure of the network, Ahmed & Khalifa (2017) classified ad hoc routing protocols into flat, hierarchical or location-based. Flat ad hoc routing nodes have same remit and functionality. A node can forward packets to any other neighbour without considering to the topology. There is nothing like head and subordinates. Unlike flat, hierarchical network routing has nodes playing different logical roles placed at different levels. For instance, many routing protocols segment the networks into different clusters. Each cluster has a designated head which aggregates and transmits the packet in-between the clusters. A location-based routing protocol on the other hand, explores the position of a node relative to the other nodes in the network and makes routing decision based on that.

MANET routing protocols is also divided into two broad categories general: *proactive* or *reactive* based on *when* and *how* the route is determined (Ahmed & Khalifa, 2017; Akansha & Sharma, 2016; Padmini Misra, 2016). Proactive routing

protocols decide the routes even before they are required and refresh them when the topology changes. In contrast, reactive routing protocols make a call to a route discovery process when there is a request for route. The routes are created only when they are needed and they are forgotten once source is done with data transfer.

It is important to acknowledge that these classifications are not mutually exclusive as a protocol could be placed under several categories. A routing algorithm can be placed under more than one groups according to different methods of classification. They are based on different viewpoints. However, classifying the protocols helps to understand the distinct features and the intrinsic relationship with others.

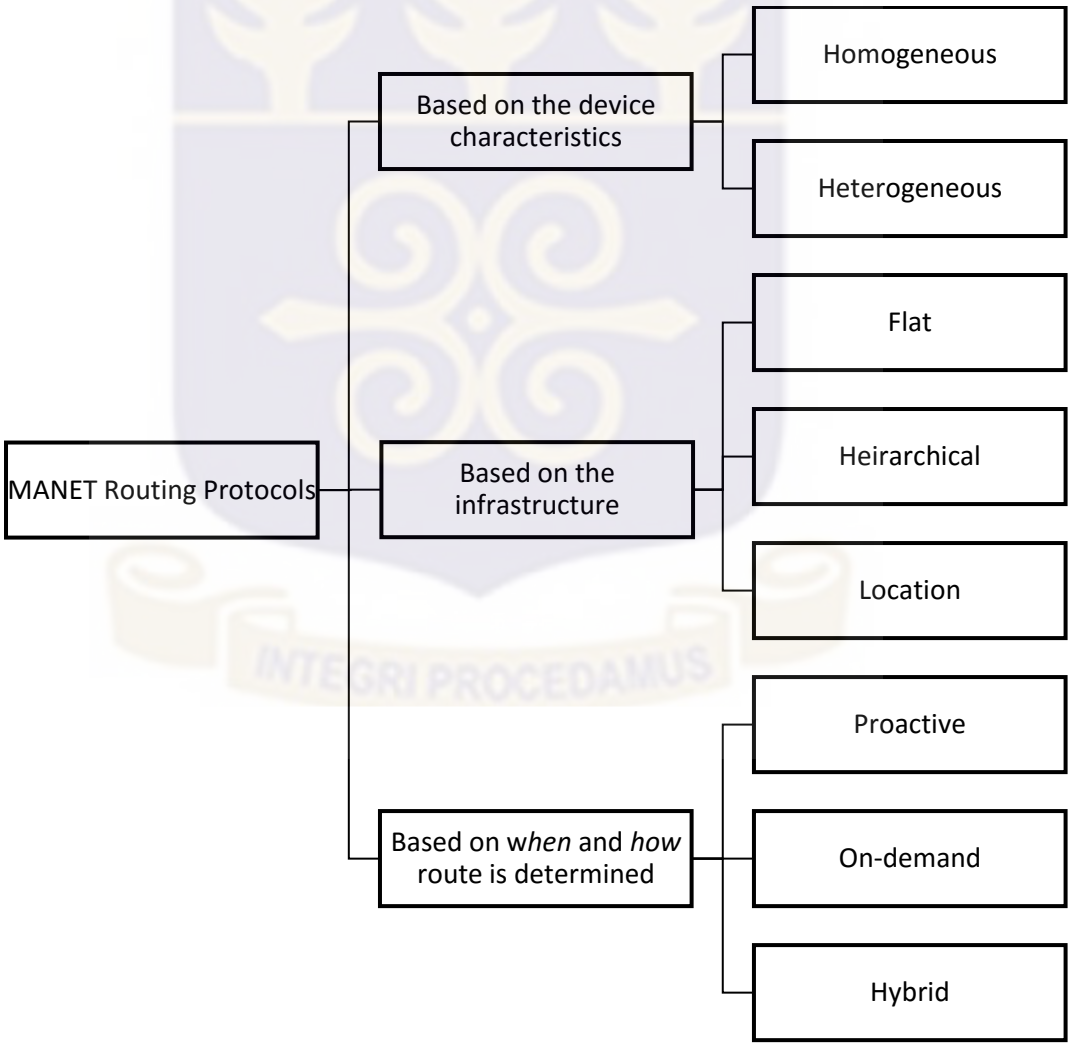

Fig 2.1: Classification of MANET routing protocols

Proactive routing protocols

Proactive routing protocols are sometimes referred to as table-driven routing protocols. Here, routes are pre-determined before a route is requested and are updated whenever the network topology changes. Each node maintains the state information of the network by maintaining table(s) that contains routing information to every other node in the network. For consistent and up-to-date route information, nodes update their routing table on changes of node to node links and propagate update messages throughout the whole network. Update of this routing table information is done either on event bases (usually triggered by topology changes) or on periodic bases. Thus each node periodically discovers and maintains routes to every possible destination in the network.

Since nodes have an up-to-date routing table available, route can be found whenever a node has data to send. Therefore, there is no further computation or additional latency incurred for the delivery of the data. However, one of the major challenges of ad-hoc network which is the constant topology change means for proactive routing protocols to be effective, nodes routing table update must also be constantly updated. But what happens when this network spans into several nodes or when there are frequent topological changes because of high mobility? This will be accompanied by several network control message exchanges. In such scenario, proactive routing protocols may not be appropriate. When the number of network nodes become large or when the network topology changes rapidly due to mobility, maintaining routes between the nodes will need large and frequent exchange of routing control packets. The amount of update traffic can be quite large (Ahmed & Khalifa, 2017). For high mobility nodes, the impact of the route maintenance overhead could be so huge such that no bandwidth is left for the data packets transmission.

Reactive routing protocols

Reactive protocols are also popularly known to be on-demand protocols. Unlike proactive where each node knows the route to all other nodes whether or not it has data to send, route creation is only initiated by a source at a time that it has something to send to a destination node. The route is created as at when a source wants to communicate to the destination. Reactive routing protocol nodes appears to be lazy. It does not know the route to any nodes, as it does not act until it has something to send and also it forgets the routes learned once it is done with the data transmission. Hence there is a need to request for route to a destination when a source node has a data to send. Nodes discover routes to a destination only on demand. Until destination of data packets is identified, the data packet source node cannot send data packet. A route to a destination remains valid so long as the route is accessible and remains so till it is not needed anymore. Popular among this class of this routing protocol are the AODV (Perkins & Royer, 1999) and DSR (Johnson & Maltz, 1999).

A source node invokes the route discovery process to discover the path to the destination anytime it wishes to send data to it. Usually route establishment process of these protocols involves three phases, namely; *route discovery*, *route maintenance* and *route recovery* of which Sharmaa et al. (2016) believe that they have separate functionalities. Whereas a route is discovered by discovery process, its longevity and stability is maintained by appropriate maintenance process. Usually, the discovery process sends out discovery messages via a broadcast scheme called flooding to discover route between source and destination nodes.

The challenge of reactive routing protocol is the fact that they incur an additional latency for discovering route to a given destination compared to their proactive counterpart (Xu, Wu, Sadjadpour, & Garcia-Luna-Aceves, 2010). In some of the

protocols, it could take a much significant time before a route is found and this may cause increase in latency for data delivery. However, the major challenge of unnecessary route updates messages to keep the routing tables up-to-date in proactive routing protocols that sometimes may be enormous traffic can be minimised by the on-demand discovery nature of the reactive counterpart. For instance, Perkins & Royer (2003) writes that the AODV algorithm though has longer latency for route establishment yet since the protocol needs no global periodic routing advertisements, the overall bandwidth demand for nodes is substantially less than in DSDV protocol which was the proactive counterpart they compared to. They do necessitate route update advertisements. It can be scale to a large population of mobile nodes as well. Another major challenge with reactive protocol is the broadcast approach used for the route discovery operations. Due to the excessive retransmissions of the route search packets, there is a severe influence on the performance of such network (Abdulai, 2009). Moreover, some measures could also be taken to alleviate these problem of longer latency and excessive retransmissions for route establishment.

Hybrid

Other protocols also exist where it is designed to employ both features of a proactive and reactive protocols. An example of such protocol, Zone Routing Protocol (Haas & Pearlman, 1998), mix both ideas of proactive and reactive protocols. For instance, in the example mentioned, nodes are placed into zones, zones can overlap though, and a zone route are managed via the proactive routing protocol, establishing and maintaining routes. For locations outside a zone, reactive protocol is in charge of building and maintaining routes.

2.2.3. Destination Sequence Distance Vector

Destination Sequence Distance Vector (DSDV) (Perkins & Bhagwat, 1994) routing protocol is a version of the Bellman Ford Distance Vector protocol for MANETs. It is a proactive protocol. Every node keeps a routing table that stores the next-hop and the number of hops to reach all destinations. Thus in each node's routing table are the lists of all accessible destinations, next hop node to take and the number of hops to make before getting to that destination. Each node periodically broadcasts its route updates and cost to outgoing links to its neighbours. Neighbours that receives from this update information recalculate their routing table using shortest path algorithm to estimate the shortest distance to each other nodes in the network. Therefore, data packets are communicated between nodes in the network via this routing table information without waiting for a route search process. The routes are known to all destinations even before a data send request.

Route entry of the route table has a sequence number that is produced by the destination node. DSDV uses the sequence numbering of the destination to guarantees loop-freedom and up-to-date routes. The DSDV guarantees loop-freedom unlike the traditional distance vector protocols. It uses sequence numbering tag to identify a fresh route from a stale route. A higher route signifies a fresh route and hence more favourable. If two routes have same sequence number, then the one with less hops are preferred. One of the main challenges of DSDV is the overhead of the periodic updates broadcasted. When a node finds a broken route it sets the hop number to an infinity and increase the sequence number to an odd number.

2.2.4. Dynamic Source Routing

Dynamic Source Routing (DSR) (Johnson, Hu, & Maltz, 2007) is an example of reactive protocols popular in MANET. Nodes of DSR protocol discovers route to any destination through the multi-hop nodes without prior knowledge of the network topology. Routes are returned through route discovery and are maintained through the route maintenance phase. That means DSR has no periodic route update messages throughout the network. Route discovery phase is initiated with a source node getting a send data request. The source node of data sends trigger, broadcasts a route request (RREQ) packet. Every node that receives this request search through their cache for the requested destination. When this route node is found in cache, it adds its own address to the header's hop sequence of the packet and simply repeats the broadcast further. This is repeated until packet reaches the requested destination node or another node with the cache entry of the request route which unicasts a route reply (RREP) packet to the source. The path is routed by using the reverse hop records. Route symmetry is not required so if the node finds other record to reach destination, it can be used instead. Hence this protocol supports unidirectional links.

DSR is a source routing protocol. Each packet of this protocol carries the complete ordered list of nodes in its header. The packets are passed on from one node of the list to another till it arrives at the destined node. Thus the packets carry the complete path that it should take. DSR has no periodic updates so it reduces the overhead for large routing updates. Nodes can store learned routes in their caches. The down side however is the fact that each packet has to carry the complete path information.

2.3. Ad hoc On-Demand Distance Vector

As mentioned above, Ad hoc On-Demand Distance Vector (AODV) is a reactive routing protocol and very popular among MANET routing protocols (Perkins & Royer, 1999). It is one of the widely discussed ad hoc routing protocols in academics and also a standard implemented in the industry. According to its developer, it is intended to be used by mobile nodes in an ad hoc network where nodes trust each other, either by utilization of preconfigured keys, or because it is realized that there are no malignants intruder nodes. Its essential advantages are that it provides adaptation to different link conditions including low processing and memory overhead. It selects unicast routes to destinations.

The following section discusses AODV as described in RFC3561 (Perkins & Royer, 2003) and how it takes advantage of some of the known mechanisms of both DSDV and DSR routing mechanisms. AODV combines the ideas of DSDV and DSR. It inherits the basic on-demand approach of discovery and maintenance of route from DSR, and uses the hop-by-hop routing, sequence numbering, and the periodic beacons of DSDV. Unlike proactive DSDV, only the nodes that are taking part of a selected path do participate in route exchanges. It guarantees a loop free through the use of sequential numbering of route and increment for every new route identified for the same destination. A node is to select a route with the highest sequence number all the time. This solves the conventional distance vector protocol's problem of "counting to infinity" (Perkins & Royer, 1999).

AODV routing protocol disseminates control messages to discover and maintain routes in the various phases; *route discovery* and *route maintenance.* It is important to mention that AODV uses flooding mechanism to disseminating the control messages in its route discovery phase. At a time that a source needs to send data to a destination,

it calls forth the route discovery mechanism. It finds the path to the destination and calls for the route maintenance should an existing path to a destination go bad whilst the source node has not finished sending its data.

2.3.1. AODV Control Messages

Original AODV issues three packets (collectively called control messages/packets) to manage the route establishment processes. They are Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs). They are the main protocols messages designed to create and manage the route information of the nodes for data packets to safely find their destination.

AODV control packets are normal IP packets. These packets have IP header and are transmitted via UDP. For example, a requesting node uses its IP address as the originator IP address for the request packet and broadcast it using the IP broadcast address 255.255.255.255. For details of the entries of the packets headers refer (Perkins & Royer, 2003).

In addition to the three aforementioned control packets, AODV has *hello message* which is like a beacon message that is used to determine a node's link to its neighbor. Each node broadcasts these beacons periodically (within a time period called HELLO_INTERVAL) to its 1-hop neighbours. Any node that receives this packet is considered as a neighbour node to the sender and this information comes very useful as it helps to keep routing table up-to-date as it can be inferred that if a node does not receive any of these beacon then there is a link broken.

AODV Route Table

Nodes of AODV network have a routing table where it keeps paths information. But unlike proactive, the nodes do not keep paths' information to any destination except to that node from which it takes part of that source-destination route chain.

A routing table contains the following fields (Perkins & Royer, 2003):

- Destination address (the IP address of the destination node)
- Destination sequence number
- Valid destination sequence number flag
- Hop count (count of hops to make before reaching destination)
- Next Hop
- List of precursors
- Lifetime (expiration or deletion time of the route) and
- And other state and routing flags (e.g., valid, invalid, repairable, being repaired)

AODV routing table are updated upon receipt of a control packet (and perhaps, data packet). On receipt of data packet, the only updates a node can make in the routing table is to set the expiration time (route lifetime) to current time plus the expiration duration. If no route is found (or is expired) on receipt of data packet, node triggers route error for a new discovery, it cannot make new entry or update any other field of the table. However, when a node receives an AODV control packet from a neighbour, it either makes a new entry or updates fields of an entry for a given destination by comparing the packets information to the entries of its route table. Thus, upon receipt of control packet, a node can make a new entry to the table or modify already existing

entry's information. The following are the various actions taken by a node on the route table upon receipt of control packet;

i. Receipt of RREQ:
   a) For the first time – the node makes new entry into the table
   b) If the destination already exists – but table contain less Sequence Number compared to that of the received packet – then the record is updated.

ii. Receipt of RREP: the node sets the reverse route to destination (it updates precursors). It does not make a new entry.

iii. Receipt of RERR: node updates the route entry by marking this route as invalid.

iv. Receipt of Hello message: node updates the lifetime field.

In summary, RREQ and RREP packets are for route discovery process whereas RERR and HELLO message are employed in route maintenance phases. For any further information refer to (Perkins & Royer, 2003). The next section discusses how these control messages are used to discover routes to destinations and maintain them.

2.3.2. Route Discovery process

As a reactive protocol, AODV source node must discover route before it can send data. When a source node gets a data sending request it checks its routing table whether it has a valid route to the said destination. If it does, then it sends the packet via that route. If there is no valid route then, it initiates the route discovery process. The node requesting for the route is herein AODV known as originating node. The originating node broadcast RREQ packet, i.e. a node that needs a route to a destination disseminates a RREQ message to all neighbour within its transmission range using the

IP broadcast address 255.255.255.255. All neighbours that receive this type of control packet process it as follows. When the receiving node is;

i. NOT the destination and knows no valid route entry to that destination: it records the packet information and rebroadcasts the RREQ packet.
ii. NOT the destination but has a valid route entry to the destination: it updates this entry of the route table and sends a unicast RREP message to the originating node or
iii. the destination node it seeks for; it updates its route table information and replies by issuing RREP to the originating node;

The basic idea of this process is to communicate with the other node for this path if it has such routing information in its table, else they also communicate to their neighbours. The process is repeated until it finally gets to the destination node itself or a node which has a valid routing entry for this destination. To offer a freshly created route, it is prudent to distinguish it from an earlier route for same destination. To achieve this, an active route is marked valid or invalid. Data packet is transmitted via only valid routes however control packets may be transmitted via an invalid route. The validity of a route is determined by comparing the destination sequence number contained in the route table of a node with the destination sequence number in the route control packet. A route table entry is invalid when the received packet has a greater destination sequence number compared to what is contained in the route table entry. The sequence numbering also helps to avoid network looping.

Each destination node owns and maintains its own destination sequence number to offer a freshly created route. The only condition that another node may change the destination sequence number for a given route is when it is reporting a lost or expired

link of the downstream (next hop) of the route towards that destination. It sets the destination sequence number to an infinity and then send RERR message towards upstream node (previous hop/precursor) which bubbles to source. Sequence number is incremented by

i. source node when a source node initiates a new route request (RREQ).
ii. destination node when it sends route reply (RREP) packet to a requesting node.
iii. Any other node when it detects a link failure and wants to forward RERR packet.

Once RREP packet is received by the originating node, it can begin sending data packets using this returned route. From now, the route maintenance phase is activated.

2.3.3. Route Maintenance

Once the route discovery process has established route successfully, data transmission can begin via this route. Due to the unpredictable topology change nature of MANET, a route maintenance phase is required to maintain this source-destination session for two reasons: (i) to find out when the route goes bad and (ii) how to handle it when this happens. A route maintenance protocol gives a feedback about the links of the route and if necessary enable them to be changed should any disruption occur due to mobility (or other reason) of one or more nodes along the route. According to the RFC3561, a route is maintained as long as that route is active.

The route maintenance of original AODV is simple. Maintenance is performed by detecting a link break before the complete failure of route. While data is being transmitted across, an active route may contain a broken link. When the data in transit along the route reaches the upstream node of the broken link, it cannot be forwarded via this link to the next hop. Original AODV has it that, if a link fades while route is

active (or if it receives a forward request of data packet destined to a node that it does not have an active route), the node that detected the loss of the link immediately drops the data packets and increment its destination sequence number in the route table to infinity to invalidate the route and updates the lifetime field to the DELETE_PERIOD. The node then broadcasts a route error (RERR) message (it may be unicast when it has only one link to this destination) which must contain the newly incremented sequence number. Any node that receives this packet updates this route entry accordingly and forwards the packet to the next upstream node. The RERR message is propagated from this node through the upstream nodes to the source node to inform these nodes that the destination is now unreachable. Repeatedly the RERR packet gets to the source node. After receiving the RERR packet at the source, it can reinitiate another route discovery if it still desires the route.

A node that rebroadcasts RREQ control packets for the same destination does so with a new sequence number higher than that of the previously lost route and tries to find a new route for its destination. This dropping of data packets and regeneration of control packets is not only going to cause loss of packets and overhead, but also delays the process of data sending. A more efficient way of repair is required. Below are the suggested improvements by the original developers.

Earlier detection of lost link

A continuous link validation is so important in ad hoc network so as to identify links that go bad early. AODV uses hello messages to offer an early link break detection. For early detection of link break (even before data arrives), Perkins and Royer suggest that AODV benefit from a local exchange of packet to check its link connectivity information periodically by broadcasting a local hello message to keep up-to-date status of the link. Only nodes that are along the active route use this hello

mechanism. Neighbour who does not receive any of these beacon packets for more than a defined time interval (ALLOWED_HELLO_LOSS * HELLO_INTERVAL), over the link is assumed to be currently lost. HELLO_INTERVAL and ALLOWED_HELLO_LOSS are set to 1000 milliseconds and 2 respectively were suggested in RFC3561. A node can therefore determine its link connectivity by listening for packets from its set of neighbours (PREV and NEXT nodes). Whenever a node receives a Hello message from a neighbour, then a link exists. Active route that uses this link is consider valid (or can even create one where such link does not exist, should it become necessary). The link lifetime for the route can also be updated if necessary. By default, AODV increases the lifetime field in the route table by increasing to a new value of ALLOWED_HELLO_LOSS * HELLO_INTERVAL. The paper also recommends that node that receives hello message with latest destination sequence number can safely update its table entry to the latest value received.

Local route repair

The original designers of AODV also come up with a local route repair mechanism to mitigate this problem of delay and packet drop when a link is lost. In this route maintenance scheme, when a link breaks in an active route, the upstream node of the break point that has data packet to deliver may choose to repair the link locally. It is an upstream node(s) that can repair when a link fails. In this mechanism of local repair, the repairing node becomes the RREQ originating node. It increases the sequence number for the said destination and then broadcasts RREQ to find this lost destination again. The repair attempt is therefore invisible to the other downstream nodes and the data source node, and so they will continue to transmit data packets. Therefore, as the

repair node awaits the discovery process to respond with RREP, it buffers the incoming data during this period of repair.

The local repair always has TTL which is always lesser than the global search TTL. However, AODV local repair attempt is only done when the break point is closer to destination than to the source node else this repair mechanism is not available and that RERR message is sent to the source node to rediscover the route.

A local link repair is likely to increase the number of delivered data packets at the destination when the link is lost since data packets will not be dropped but rather redirected through alternative route to the destination. Traditionally, it is just dropped and send RERR message which travels from where the link was lost to the source node for reroute. Delay is also minimised as destination is expected to be closer to the repairing node than the source node. RREP is expected to be received faster than if the RREQ was from the source node. However, the side effects may include the risk of repairing a route that is no longer needed and sometimes it also results in increased path length to the destination. Besides, cost of failed local repair could be so dear; time wasted time (delay), the entire route timing out and large amount of buffered data are dropped.

2.4. Route Maintenance schemes

A route maintenance phase determines the route long life and the stability. It is so important to the network as it has a significant impact on the route lifetime and the network performance. Bisengar et al. (2012) saw that data packet delivered decreases as the number of link changes between nodes experiences increases. The findings this paper suggests that a long live route improves the data packet delivery ratio and the

other metrics as more data transit will be successful. According to Kumar, Kumar, Pradhan, & Yadav, (2011) route maintenance scheme is for two main reasons; (i) to achieve network stability and (ii) to reduce the cost of excessive overhead incurred in discovering new route. Just as Sharmaa et al. (2016) also argued that route discovery and route maintenance have separate functionality. They believe that route's stability and longevity is sustained through appropriate maintenance scheme. The authors also think that global repair, local repair, backup routing, error notification and link connectivity techniques are all part of the route maintenance process.

Several route maintenance schemes have been proposed to improve the lifetime of a source-destination route. In such improvements, the route maintenance scheme assists routes to adapt to frequent link failure as a result of node's mobility, power drain, etc. Their improvement can be seen in the network performance in terms of the throughput, packet delivery ratio, latency and other network performance metrics which can be linked to the frequency of link breaks (Bisengar et al., 2012) and the number of route control packets and delay incurred to find or repair the route.

Classifications of Route maintenance schemes

A survey of various AODV route maintenance schemes, that shows the similarities and distinctions between the various schemes categorised them into *Local Repair* and *Global Repair*. Local repair is when the intermediate node of that broken link choose to repair the link without the knowledge of source and the other upstream nodes [e.g. the local route repair mechanism mentioned above, AODV-BR (Lee & Gerla, 2000), Router handoff (Abhilash et al., 2002), AODV-ABR (Lai et al., 2007)] and it is a global repair when source node will have to abandon the currently existing route that has a link break and change to a new alternative route once a link break occurs at any part of the route [e.g. the default AODV route maintenance where the source node is

the only node that can initiate route search when link fails at any point, preemptive routing (Goff, Abu-Ghazaleh, Dhananjay, & Kahvecioglu, 2001)]. Global repair usually abandons existing route and flood the network with request packet. As stated above, repairing a link locally will generally decrease the dropped packets and may be faster, yet if it fails, it will result in a drop of large amount of data and precious time is also wasted.

Route maintenance schemes may also differ in when they react. In some schemes the route becomes unreachable completely before measures are taken to repair the situation while others employ mechanism to detect link failure likelihood and tries to resolve it even before this happens. Nodes may also increase the redundancy of routes in the network by keeping alternative/backup route to a given destination that can be used on route repair. Redundancy in some route maintenance variants has been shown to speed up route search for fast route recovery. This group of the literature that uses backup route (e.g. AODV-BR, AODV-MRM) selects route from a pool of routes (often called alternative route table, ART) to the same destination to forward packets as compared to others that repair the route by sending out a route search packet either from upstream node (local repair) or from source node on repair. Thus, in backup routing, data packets are routed via alternative path of neighbour nodes on route break to continue the data transfer without making a new route search. One of the challenge of such schemes is how to identify members and maintain up-to-date ART.

2.5. Related works

Some considerable works dedicated to route maintenance processes in MANETs are discussed. The discussion here covers schemes that mostly maintain a local neighbourhood information. Most of these solutions suffer from a number of disadvantages. Below is a review of some of the existing solutions with a brief description of their drawbacks.

AODV-Backup Routing (AODV-BR) (Lee & Gerla, 2000), used alternate routes built by the overhear of a RREP packet. Each node upon overhears of a RREP packet registers into its alternate route table that address of the node it heard this RREP packets from as its next hop to the destination. Nodes that are recording these overheard packets are not necessarily taking part of the active route chain until the route decides to include them when it executes a route repair process. They are only doing this to create a redundant route to the destination with the hope that if something goes wrong it can be used to transfer data. Again AODV-BR used the overhear of data packets that are transmitted by the next node recorded to keep the alternate route table up-to-date and times them out when data packets transmission cannot be heard. Thus, no additional messages are required for this purpose.

When a link breaks at any node along a source-destination route in AODV-BR, it will perform "a one-hop data broadcast". Neighbour nodes that have entry for the route to the destination forwards those packets to the destination. The node that detected link break and broadcasted its data packet will then send RERR packet to the source node for a route rediscovery as original AODV does. The paper also looks at the local repair version that tries to repair the link locally without informing the source node but this is like the default AODV local repair (AODV-LR). The backup routes offer more stable connections than pure AODV as data packets are not just dropped because there

was a link lost. Of course, it improves performance metric such as throughput because there is no need of data drop and retransmit from the source, yet it doesn't solve the problem of frequent route breaks nor avoids dissemination of RREQ packets to search for premature route.

Moreover, in a dense environment, the "one-hop data broadcast" can result in useless heavy traffic when several nodes exist with backup of such route as each of these nodes unnecessarily delivers its duplicated data packets to the destination through their alternate routes creating a bottleneck, congestion, for the receiver. This redundancy packets result in a heavy load especially in heavy traffic condition.

In pre-emptive routing in ad hoc networks (Goff et al., 2001), routes proactively switch paths when the quality of the active route becomes suspect. It defines a threshold on the received signal strength that will give sufficient time for nodes to get new paths before the failing one gets disconnected. This threshold was estimated based on the time needed to complete a path query and the motion of the node. A warning packet is generated to alert the source node when the received packet has signal strength that is below the threshold. The source then initiates path discovery early, to avoid disconnection altogether. A path whose links have signal strength above the threshold is sought immediately and data packets are redirected via this new route even before the suspected one fails. The early search of route in pre-emptive route maintenance is by the traditional RREQ dissemination from the source node. The only difference between the original AODV and the preemptive AODV is the proactive detection link break. Preemptive AODV demonstrated improvements over the original AODV as the number of broken paths was significantly reduced, as well as latency and jitter of the packets were better. However, there is an increase of route overhead because of the frequent warning packets and path rediscoveries. If a node of suspect

link, which perhaps change course to move in favour of the link, it may trigger warning since a path became suspect. Therefore, chances are that a number of path rediscoveries may be initiated that may not be necessary yet the scheme floods the network with new route search packets. Such a phenomenon may induce the broadcast storm problem for a high mobility network.

Abhilash et al., (2002) also presented router handoff route maintenance scheme, yet another pre-emptive route repair in AODV. It improves the throughput as a result of smaller overhead and delay. By the help of a Neighbour Table (backup), other nodes of stronger links in the vicinity of a weak link along active route, are selected to merge a broken link between nodes. The mechanism selects a node in the area of a potential link break that can bypass the weak link. By putting a threshold on the received power of a node, a node can handoff to another node when the receiving power is less than the threshold.

Unlike AODV-BR that uses overhear RREP packets, router handoff uses hello message of AODV to discover neighbours and maintain them in the neighbour table. This neighbour table also contains information about the links status of each neighbour. On link break alert, the upstream node of the breaking link broadcasts a Handoff REQuest (HREQ), one-hop packet that contains the information of the link nodes (the next and previous hop nodes) that use the link. When a neighbour node receives the HREQ and can route packets between the two nodes based on its neighbour table sends a Handoff REPly (HREP) and make this entry into its route table. The node upon receipt of HREP updates its routing table to place the sender of the HREP as its next hop. Node forwards its data to the new hop thereby avoiding the broken link. HREQ and HREP are just one-hop packets.

The difference between router handoff and preemptive routing is that the latter triggers warning for source to do undergo route search procedure again but router handoff will just do one-hop local route search. Source floods the entire network with RREQ after anticipating route break whereas router handoff tries to locally repair fading link by finding an alternate node from neighbours to handover existing routing information to bridge the disconnected link. This way of route repair will always result in increase of route length. Moreover, repairing node may not have a fore knowledge of whether such neighbour exists that can repair. Besides there is the possibility of selecting a staled neighbour link to repair this route which will eventually fail. This will be unnecessary delay which will cost in large packet drop as well as all the downstream route timing out.

Adaptive Backup Routing, AODV-ABR (Lai et al., 2007) is an improvement of AODV-BR. It adopts all features except the broadcast of data packets. It uses one-hop neighbour handshake process to choose just one node to repair the broken route without broadcasting the data to all the immediate neighbours. On a break link, one-hop neighbour handshake process of AODV-ABR defines two new one-hop control messages: BRRQ (Backup Route Request) and BRRP (Backup Route Reply) containing the route information (an ID for this message, Destination IP, hop count, etc). BRRQ message is broadcast to the one-hop neighbours of the node when link breaks while sending data. BRRP is unicasted to the originator by all neighbours that received the BRRQ and have alternative route entry in its alternative route table. Node makes decision of which neighbour is selected to repair the link. With the help of hop count field, repairing node selects a shorter alternate route. AODV-ABR could repair a broken link anywhere along the primary route provided an alternate route exists except that the search range is limited to just one hop. To increase the search range,

AODV-ABR combined the AODV–LR to increase the probability of finding an alternate route to the destination. Again, ABR's repairing node may not have a fore knowledge of whether such neighbour exists and will delay unnecessarily when such neighbour node that repair this does not exist.

Du, Zhu, & Zhang (2010) modified an AODV routing algorithm on its local route maintenance (LRM) for a Tactical MANET. AODV-LRM speed up the route rediscovery by identifying an idle node. For fast return of route during route repair phase, the paper suggests the destination node does a reverse route search for idle nodes to produce many local route redundancies in the network. In this paper a threshold is set for a node's idle time. For a node that detects its idle time is greater than the threshold, it is considered to be channel-idled and the node is therefore an idle node. An idled node tries to do a backward search by broadcasting a Reverse RREQ packet (R-RREQ), a new packet of similar format as RREQ to search for the source nodes whose destination is the idle node which is doing the search. Thus; the authors proposed that a destination node begins to search for source node whenever the channel idle time is exceeded (a reverse route search method). Should source need this destination, it will search for this route according to default AODV. Route search packets from both source (RREQ packet) and destination (R-RREQ packets) when they coincide at any intermediate node, it replies both (destination and source nodes) with a RREP packet, consequently shortening the route search and control packet's traversal time. The problem with this way of route maintenance is that scheme is going to overflood the network with route request packets coming from both source and destination.

In Sharmaa et al., (2016), the authors suggested how a broken link could benefit from selecting a more informed neighbour node. In their scheme a node that overhears

more nodes was suggested to achieve the goal of local route repair and used such for in the AODV modified route maintenance scheme (AODV-MRM). In the study, each node keeps neighbour table (NT) and alternate route table (ART) using the periodical hello messages. Through the periodic updates, nodes identify their neighbours and update their neighbour table with overhearing factor (OF), a new field added to the Hello packet header. An overhearing factor of a node, according to this study, is the number of neighbour nodes a node can receive their hello packets. For example, if a neighbour receives hello packet from only one member then it has OF value 1 and OF is 2 it is receiving the hello packets from the two nodes. Thus, the OF value of a node is incremented by one any time a node receives a hello packet from a new neighbour.

On link failure, the upstream node at the break point finds a suitable neighbour from its neighbour list by selecting one with highest OF and as well has a height value greater than the height of the node itself. Thus, where two or more eligible neighbours have the same OF, they introduce yet another parameter, *height*, whose *value* is used for proper selection by selecting the one with a higher height. The height value helps to separate the forward and backward nodes and it is necessary to prevent the risk of loop on local repair. Starting from the source with a value of 1 as its height, puts this value in the data packet before sending. On receipt, the next node is to set a height to itself as the received height value plus 1. It also forwards data packets with its height value calculated and the next receiver does same. According to this study, a neighbour node which is not on the *S-D route* overhearing data packet sets its height as the average height values overheard. For example, in Fig 2.2 below the upstream node I_4 lost connection with I_5; since I_4 is the upstream node it takes the responsibility of repairing the link. It has two neighbours X and Y each with OF value of 2. Selecting the highest height valued node, the node X gets selected over node Y.

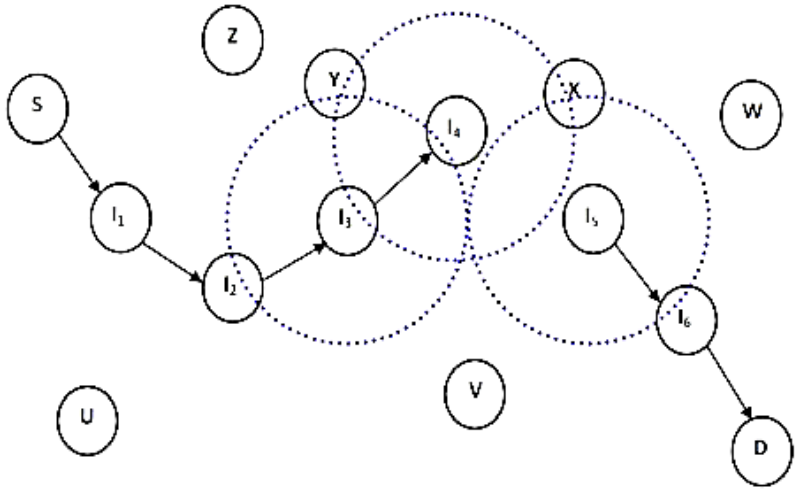

Fig 2.2: Route repair mechanism of AODV-MRM (Sharmaa et al., 2016, p. 8.).

Imagine a situation where X does not exist at all, Y is going to be used. However, Y cannot merge the link I_4-I_5 but rather I_3-I_4. It is the responsibility of upstream node(s) to repair link (Perkins & Royer, 2003). I_4 does not belong to the upstream node(s) of the link I_3-I_4 in question but I_4-I_5. Then I_4 must not keep any information of Y as its alternative route for this S-D route since it cannot be responsible of repairing I_3-I_4 link when it goes bad. Hence, Y should not appear in the ART of I_4 but rather in I_3. Thus, the ART includes an inefficient neighbour that will fail it is selected for repairs.

2.6. Chapter Summary

The chapter has discussed the ad hoc network and it challenges in finding and maintaining routes. Considering the nature of MANET, to realise the full potentials, then the network must adapt to the dynamic topological nature of the ad hoc environment. Research has improved the performance of on-demand protocol like AODV and in particular in the area of its route maintenance phase, however there is still room for improvement. Nodes mobility cause continuous topological changes and

hence route maintenance must be adaptable to these changes. Local backup routing can be useful in this area, yet for same reason of mobility, backup route maintenance can fail and its effect may be costly.

As stated in Sharma et al., (2016) the main cause of unsuccessful route repair in backup routing is the inadequate neighbour information. So most of these local repair processes end up choosing an inefficient neighbour to locally repair a failed link. The cost of local repairs failure therefore becomes unacceptable; huge amount of undelivered data packets in the buffer and unnecessary timing out of the downstream nodes as well as delay of route rediscovery. It is therefore proposed that an efficient mechanism whereby carefully selected neighbours are used to form the alternative route table from which a node is chosen to repair a failed link to better the route maintenance.

In line with Bisengar et al., (2012) observation that for two networks of similar characteristics, the one whose nodes connectivity last longer will often have a better data packet delivery ratio, the paper seeks to prolong the lifetime of the route by ensuring that each individual link can adjust themselves dynamically. Nodes of MANET have motion that cannot be controlled by the routing protocol. It is not possible in MANET to either stop all nodes from moving or to make motion of a particular pattern relative to each other to favour the network. How can we therefore create a long-live route?

It is, however, possible to create redundant paths which can be used when the previous fails and this is seen in all the backup routing protocols. The primary difficulty in creating redundant route is how to choose a node that can repair a lost link. In this thesis we present a dynamic way of creating an alternative route table containing nodes that can bridge a lost link because of the node's mobility. This will

make route to adapt to the motion of nodes relative to one another which forces the node interconnections to change. In the next chapter, an alternative way of the route maintenance mechanism which will adapt to the dynamic topological nature of the ad hoc environment is discussed.

CHAPTER THREE

METHODOLOGY

3.1. Research Method

There are several approaches to carryout scientific research and investigations. Some of these approaches includes experimentation, simulation, observation etc. This thesis was conducted using simulation approach. Evaluation of the proposed route maintenance scheme requires a large set of mobile nodes and must be conducted under different network conditions. Therefore, the choice of computer simulation approach for this study is not arbitrary. Network simulators have become a valuable tool used in research to develop, test, and diagnose network protocols. Simulations have a lot of economic benefits and flexibilities. Usually, in testing network protocol, a large set of actual hardware and infrastructure is required and may also require different conditions which may not present in real circumstance.

Therefore, using simulation method, a large set of network node and the right conditions could be created to test the various scenarios. Moreover, each scenario could be repeated so average values could be used for the analysis to minimize spontaneous errors. Through the simulation, the proposed method was tested for network characteristics and data was collected for analysis and discussion. Section 3.3 gives details of the simulation setup.

3.2. Proposed Solution

In this section we discuss a local backup route repair mechanism to improve the performance of MANET routing. In this improvement mechanism, we introduce a local route repair algorithm which will prevent frequent route break and global rediscovery operations in on-demand routing protocol. This is to avoid the default of

total route invalidation and flooding of the route request packets of the AODV protocol when a link failure occurs within the route.

3.2.1. Proposed Route Maintenance Scheme

The dynamic nature of ad-hoc network requires a route maintenance that also emulate this dynamic nature and typically adapt to the changes that occur in the topology. A loss of link connectivity results in route failure. This section presents a mechanism to keep an active source-destination session when a link break is detected. This thesis designs a route maintenance mechanism which (i) will *merge the route when a break occurs* at any node-to-node link within an active source-destination route (S-D route) and will also (ii) *delete node link from the route when link redundancy occurs* in the active S-D route. The proposed scheme can be used proactively or reactively to reconstruct a route. The thesis discusses the reactive use of this local route maintenance scheme when link break occurs using the default hello link connectivity detection.

Link Merge Operation

Fig 3.1 illustrates the transmission ranges of three neighbour nodes, n_i, n_j and n_k indicated by the colours black, green and red respectively. The blue line l indicates the trajectory of a node n_k with points indicated along the path. There exists an active link $n_i - n_k$ which is part of an active source-destination route (S – D route). From the figure, it is clear that both n_i and n_k have n_j as their one-hop neighbour. n_j can overhear both n_i and n_k. Herein this thesis refers to n_j as *JointNode* to the link $n_i - n_k$. By definition, a *JointNode* is any node that belongs to the one-hop neighbour of two or more members of an active S-D route nodes. To this communicating link $n_i -$

n_k, neighbours of such category are considered as members of this link's *JointNode set*.

In Fig 3.1, n_j is one-hop neighbour to both n_i and n_k who are (perhaps intermediate) nodes forming active link $n_i - n_k$ as part of S – D route. Let n_k trajectory through its neighbours n_i and n_j coverage to be as indicated by the line l. As n_k moves, at A to B this node n_j is not *JointNode* to this link and hence does not qualify to be a member of this link's *JointNode set* yet. However, between C through D to E where the active link $n_i - n_k$ fades, n_j remains as a *JointNode*. Due to the mobility of nodes, these phenomena (node becoming members of *JointNode* set and expiring to be a member, hence link between them fading) always happens in AODV.

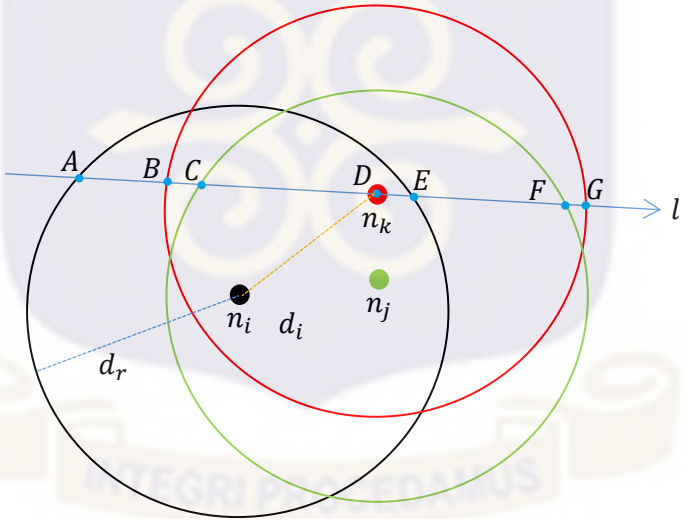

Fig 3.1: A *JointNode* of an active node-to-node link

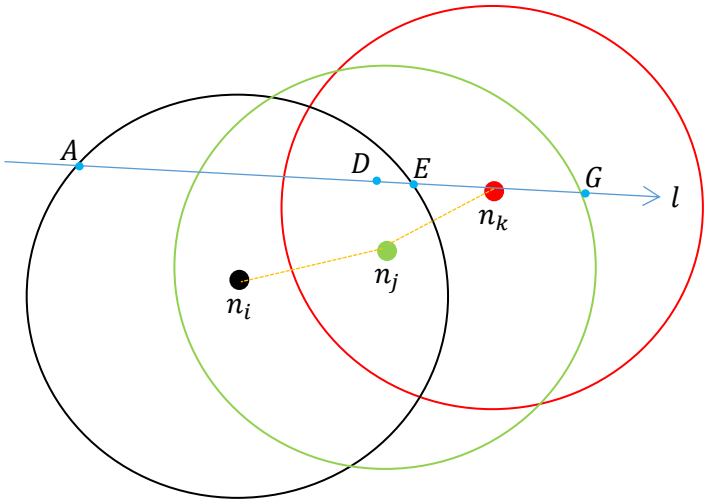

Fig 3.2: A *JointNode* merges a fade link to prevent break in an active link

JointNode n_j can potentially link to both active nodes, hence, could potentially be used to bridge the receding/receded link (as shown in Fig 3.2) which may eventually render the entire route useless. In this study we propose that these *JointNodes* be identified and kept up-to-date for local route maintenance purposes should a node-to-node link goes bad in a valid S-D route.

Link Redundancy Deletion Operation

In most literatures cited, local routes repair often leads to increase in routing length. However, no direct measure is taken to reduce this route length, except by making best choice route at a new route discovery phase. This thesis presents an innovative route length shortening mechanism which dynamically shorten the route length without having to wait for the next route rediscovery process which will only occur when a link break repair process fails for an active route.

Due to the mobility of the nodes, some nodes in the S-D route that were very far away which could not hear each other directly except via intermediate nodes during discovery phase may become directly reachable. Route maintenance should be able to

detect these links and take advantage by taking out of the S-D route chain those nodes that become redundant in order to reduce the hop counts. Herein, any active session node that can receive packets from any node either than its precursors (its next and previous nodes) for a given S-D route is defined as having a redundant node(s). For instance, an active route having the links $n_i - n_j$ and $n_j - n_k$, if node n_i can reach node n_k directly without n_j as shown in Fig 3.3, then n_j is considered redundant to this route. The S-D route chain can be shortened by cutting off this redundant part (i.e. from $n_i - n_j - n_k$ to $n_i - n_k$). Therefore, such nodes are identified and deleted from the route and this is referred to as *redundancy deletion* in this thesis. It is important to note that redundant node n_j can span into several nodes. This is demonstrated in later sections.

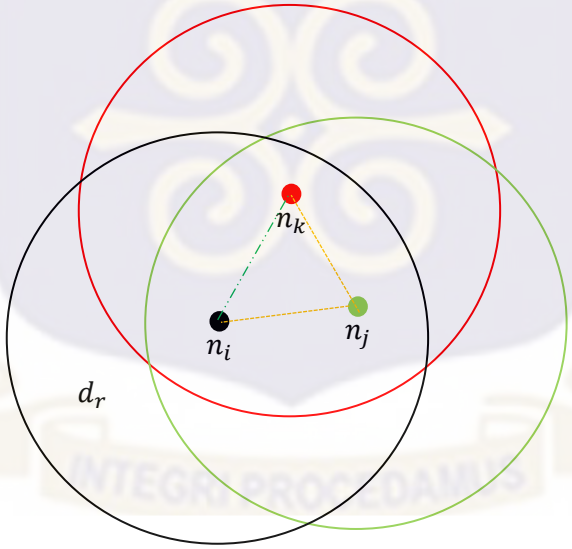

Fig 3.3: Nodes can reach each other without intermediates in link redundancy

The two mechanisms described above; *link merge* and *link redundancy deletion* are the proposed *route repair operations*. Both route repair operations require the identification of the S-D session and *JointNodes* to individual links of a route. The next section discusses how the above mentioned repair operations can be achieved.

Identification of *JointNodes* of a link

To enable us to identify nodes taking part of a given active S-D route (source, intermediate and destination nodes), each S-D route session is considered to have a unique ID known to each participating node. Each node taking part of a given route also has an ID and is also considered continuously broadcasting a one-hop neighbour packet (this study uses the AODV Hello message) which contains the S-D route ID, sender's ID and the sender's height (the node's position on count of the S-D route nodes). Thus, each active node knows its hop count from source node which is 0 and increases by a margin of 1 per hop to the destination node that has the highest height value that is equal to the route length (the value of the hop count field). Each active session node broadcast this packet to its one-hop neighbours.

Whenever a node receives this one-hop broadcast packets from the active route node that are in its neighbourhood, it treats them as follows;

i. Any node that is not part of the S-D route but can receive from just only one node, after a wait time, should discard this packet.

ii. However, if a node which is not taking part of this S-D route receives this type of packet from more than one neighbour, then this node is a potential member of the *JointNodes* linking the nodes that sent these received packets. The receiver processes the packets and unicasts this information to the upstream node among the two nodes that sent their 1-hop packet, identifying itself as a member of the *JointNode* set for the said link. Note that a link here may not be direct, i.e. the nodes may not be adjacent to each other, perhaps of some hops away. Receiver updates its *JointNode* set with this information. The receiver updates its *JointNode* set as follows;

    a. Makes an entry, if such entry does not exist already or

b. Updates its lifetime value by increasing it to a current time plus an allowed period (and other necessary accompanies; signal strength, energy, traffic congestion, buffer status, link residual lifetime, etc. should this adaptation information be necessary). This other adaptation is not considered here in this study.

iii. Moreover, when a receiver who is part of this S-D route receives this hello packet, it should also process it as follows (beside default use of hello packet to monitor link status). If a node receives from other nodes either than its precursors (the next and previous hop nodes), then there exist a potential link redundancy. Thus; any active node member that identifies itself as a *JointNode* to any node either than its next and previous node signifies the presence of redundancy in that route. However, care must be taken not to abandon a good link for a weak one which will fail soon after establishing. It is also good to acknowledge here that source and destination nodes have only next and previous hop nodes respectively, but not both.

The redundancy mentioned at this stage (same can be said of the *JointNode* membership suggestion) is only suspicion until nodes process it and confirm it as care must be taken to avoid a too weak or bad link being established and also to avoid a ping-pong between the two operations discussed (link merge and redundancy deletion). Hence, this is implemented by incorporating a mechanism to check the advantage of the route to be established over the already existing one before accepting this new link. Several checks can be incorporated including signal strength, energy status of node involved, traffic congestion, buffer status, link residual lifetime, etc. This thesis focuses on how to implement the merge and delete operations and hence do not place premium on the various check mechanisms. It accepts link formation owing to the advantage of a gain in the hop count.

The receiver of an active node's broadcast, who identifies itself as a *JointNode* (including the ones that identifies redundancy exist in the route) must calculate its gain in the hop count and check the signal strength to both links if it is not below a defined threshold. The processed information is unicast to the upstream node (the node with the least height) identifying itself as member of its *JointNode* set. Based on these values, where several options exist, the decision is taken whether to establish the link with this node or to ignore it.

Hop-Gain (H)

The section discusses how the route length is affected when a route repair operation occurs between any two nodes within the S-D route, a priority cost-benefit processing. With the knowledge of route hop counting field, *Height*, in the hello packet, an idea of what gain (in-terms of hops) that can be made after each operation. Fig 3.4 - Fig 3.7 illustrates the route repair operations as discussed above. From source node to the destination node, each mobile node knows its height in the route, Fig 3.4. As mentioned earlier, in-between the two nodes involved in a repair operation can span several hops which are cut-off when an operation succeeds, refer to Fig 3.5 and Fig 3.6. It is also important to note that in an *adjacency link merge operation*, Fig 3.7, no node is cut off the S-D route but rather it is increased in length by 1 with a new node added to the route.

In general, whenever a route repair operation is executed between any two nodes of heights i and k of active $S - D$ route of total route length m, the new route length will be;

Route length after link redundancy deletion: $m - (k - i) + 1$

Route length after link merge: $m - (k - i) + 2$

Where $k - i$ is the height *difference* between the two nodes involved in the repair operation such that, $0 \leq i < k \leq m$ and $1 \leq m \leq \infty$; ∞ is the total route length. Note that in AODV a route is only valid for a finite number of hops. By default, the maximum route length that can be considered valid is equal to the network diameter which has a default value of 35, any hop beyond this value is considered infinity (Perkins & Royer, 2003)

$$Total\ hop\ count = m$$

Fig 3.4: Height numbering of an active *S – D route* of $\boldsymbol{m}$ hops

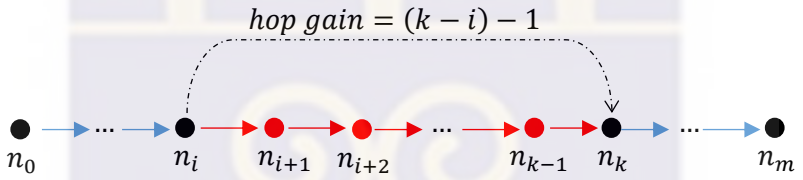

Fig 3.5: Link redundancy deletion operation along *S – D* route

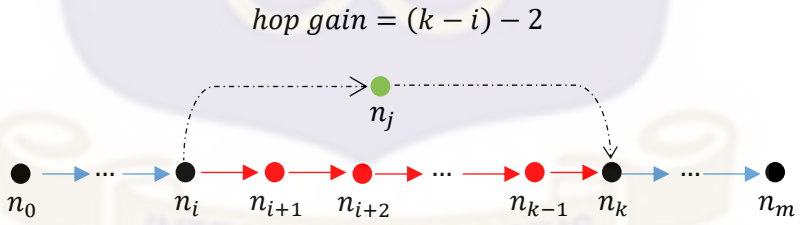

Fig 3.6: Link merge operation along *S – D* route

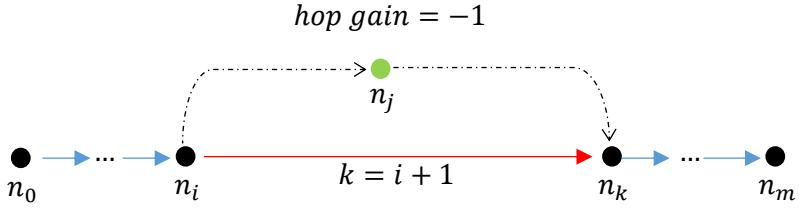

Fig 3.7: Adjacency route merge operation along *S – D route*

From the height field information in the received hello packet, a node computes the Hop Gain, H, as follows;

$$H = m_{before} - m_{after}$$

$$= m - [m - (k - i) + 2] = |\boldsymbol{k} - \boldsymbol{i}| - \mathbf{2};\ \text{for link merge operation}$$

$$= m - [m - (k - i) + 1] \quad = |\boldsymbol{k} - \boldsymbol{i}| - \mathbf{1};\ \text{for redundancy deletion operation}$$

Where m_{before} is total hop count or route length before the repair operation.

m_{after} is route length after the operation.

For all operations; $H \geq -1$ since from the above $0 \leq i < k$. Redundancy deletion operation will always result in a positive gain as $m_{before} > m_{after}$ and link merge operation will in its worse situation occur between adjacent nodes where $k = i + 1$, therefore, $H = -1$ as shown in Fig 3.7. A case which must be resorted to only for an imperative reason (a special case intended to repair a failing link – it will cause a severe loss of performance as a result of the loss of connection if the operation is not committed). Moreover, $H = 0$ when just a node is deleted and replaced with a node, which is not necessary unless the node goes bad or perhaps we intend to make other gains like energy, traffic balancing, etc. In summary, the proposed route repair operations discussed is analysed as follows:

i. A link merge using a *JointNode* will always cost the network an additional hop if only if the operation occurs between adjacent nodes. (i.e. if $k = i + 1$; $H = -1$).

ii. There is no gain or loss in hop if the operation is to "cut-off a node and reconnect with a *JointNode*". Thus a link merge operation between nodes that are two hops away from each other (i.e. if $k = i + 2$; $H = 0$).

iii. Any other operation makes a gain in hop (i.e. $H \geq 1$).

Therefore, the route repair operations are proposed to be implemented as follows. All other things being equal and with the exception of the adjacency link merge, case 1 above, all the route repair operations can be resorted to as an *alternative* even when there is no loss of connection but for other reasons of optimisation. Such mechanism can be used to optimise the route by incorporating other network sensitive parameters like link status, traffic volume, bandwidth, energy, distance, buffer status, link residual lifetime/duration, and etc. to offer the maximum performance of an ad hoc network. Adjacency link merge is however recommended for an *imperative* case because of the cost of which may be incurred. Thus, nodes perform this operation but for technical analysis that it is good to do so, e.g. when route is (perceived) disjoined. Thus, in this route maintenance scheme, nodes can dynamically configure themselves to form best routes even if the returned routes from the discovery process (or has altered due to the continuous topology changes) is not the best one.

The down side of this proposed maintenance scheme, besides the negative effect of an unsuccessful local route repair, will be that the neighbours of the active route nodes is actively involved in listening to the medium to find out whether it can act as a *JointNode* all the time. They also actively transmit packets to their link repair nodes to keep their *JointNode* set up-to-date. It is important to acknowledge that the expected *JointNode* triggers to a link repair node (upstream node of a link) could become large if the link is located in a very dense area. In spite of these mentioned shortcomings of the proposed method, it is still prudent for the repair node to have information at hand to dynamically adapt to the changes in the route maintenance phase.

Height Recounting for Downstream Nodes

For the route repair operations, height value h, of each downstream node from the node that executed the operation are affected by the operation. Thus towards

destination node, nodes that are above node n_i that have height value h, such that $h > i$. However, upstream nodes' height numbering is not affected by the operations (i.e. below node n_i towards the source node such that $h < i$). For instance, whereas after adjacency link merge operation, each of the downstream node's height is affected by 1 more hop since $n_i - n_{k=i+1}$ is now $n_i - n_{j=i+1} - n_{k=i+2}$, a node redundancy deletion rather shortens the height by 1 as difference in the height of the downstream nodes. Route recount is required for only the downstream. If the downstream nodes are informed the kind of operation, they can sync their height values.

In this thesis, synchronisation of the height value of downstream nodes after either of the repair operations discussed above is achieved by using the height field value of the hello packet. Each node, when link is not broken, must receive hello packet from both next and previous nodes. This work propose that nodes use hello messages to synchronize their height value. Whenever a node receives from its previous hop (from its upstream node) it must synchronize with the height value in the message. The receiver of a hello packet from a previous hop node with its height value not being same as the receiver's height minus 1 confirms a change in hop counting. Receiver node therefore changes to the new by adding 1 to the received height (i.e. nodes set their height value according to the height value of its upstream node). Node updates its height field in the routing table accordingly and transmits a hello packet with this new value. Any node that notices a change in its height value must not wait for the next hello interval but send out a new hello packet immediately. This process bubbles up till it gets to the destination node.

In this way, there is no need to define a new packet for hop recount or modifying data packets to have height field and overhear it or hop recounting across the route from hop-to-hop to recalibrate the route each time route repair action occurs. Sharmaa

et al., 2016 and Lai et al., 2007 synced the downstream nodes' height by overhear of data packet which was modified to have a height field. This study could adopt this mechanism but this study aims to separate data transfer from routing mechanism so that AODV's sole aim as a routing protocol independent on the type of data traffic and/or transport layer protocol being used to send the data over the network.

Flowchart of the proposed route maintenance operation

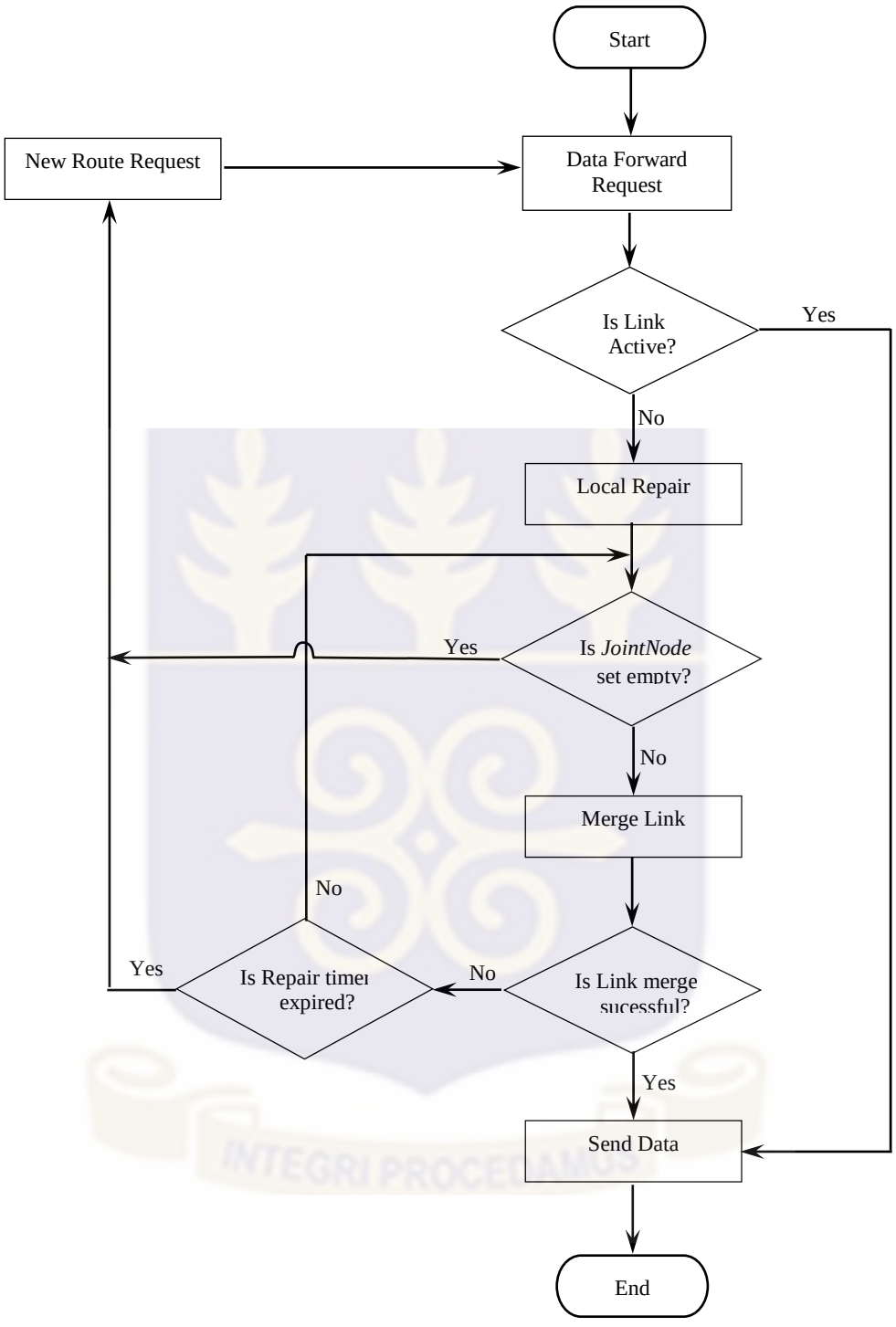

Fig 3.8: Flowchart of the proposed route maintenance scheme

3.2.2. Proposed Modification to the existing AODV Module

The proposed route repair mechanism mentioned here can be applied to the on-demand MANET routing protocols. In order to test the route adaptations, the modified AODV was simulated with the features discussed in order to take full advantage of the route adaptation mechanisms mentioned. The AODV module of the NS 2.35 (Issariyakul & Hossain, 2012; Fall & Varadhan, 2011) was modified and recompiled alongside the original version. The modifications are discussed as follows.

Details of the Hello packet

A hello packet which is a RREP with TTL = 1. To achieve the proposed model, the hello packet is modified to include the height field, all other fields remain as stated in the (Perkins & Royer, 2003).

Destination IP Address:	The node's IP address.
Destination Number:	The node's latest sequence number.
Hop Count:	0
Lifetime:	ALLOWED_HELLO_LOSS * HELLO_INTERVAL
Height:	The height value of the sender. Newly added field.

JointNode update packet

There is the need for a new update packet which is triggered by the neighbour only when it identifies itself as a *JointNode*. It is a unicast packet. The thesis also used the RREP packet with other information just like the hello packet except that it has a hop-gain field instead of height field. The hop-gain field value is calculated as the difference of the two height values of the route nodes that sent their hello messages with same destination IP address.

Details of the JointNode set

A *JointNode* set here represents the alternative route table. This is where an upstream node keeps the information of *JointNode* notifications received from the *JointNode* nominating packet. The table has these fields:

Destination IP Address:	The node's IP address of destination.
Destination Sequence Number:	The node's latest sequence number.
Next hop ID:	IP address of the *JointNode*
Hop-Gain:	Hop-Gain

Link merge packet

It is one-hop unicast route request packet sent to a selected *JointNode* set member on link merge operation. It is to check the validity of a *JointNode* and also to prepare it to route packet to the next hop. A receiver immediately replies with an ACK to the sender and copy the requested route from the ART into its route table.

JointNode back-off packet

In order to avoid unnecessary *JointNode* update packets, to the upstream node, *JointNode* back-off packet to avoid any further trigger. For simplicity sake, it also has every feature of the hello except its lifetime field represents a back-off period. The height field carries the maximum hop-gain of the *JointNodes* in the sender's *JointNode* set. This packet which is triggered only when a node identifies its *JointNode* set has exceeded the allowed number of nodes. The maximum number of nodes allowed in the set and back-off period values were set to 3 and 3000ms respectively. The choice of these values were arbitrary. A further study may be done to ascertain which value will be best.

3.2.3. Proposed Algorithms

This section discusses the route maintenance algorithms. Algorithm I is executed by a neighbour that identifies itself as a *JointNode*. Algorithm II is executed by an active route node whenever it receives a *JointNode* update packet. Algorithm IV implements the link merge operation and the algorithm V implements the link deletion operation when link redundancy occurs. Thus, whereas the algorithms I & II assists in building the neighbour knowledge necessary for the route repair operations, the IV and V implements the operation itself. Algorithm III is only called by the algorithm IV for a *JointNode* selection.

Algorithm I: *JointNode* Notification

Executed when a node receives hello packet for a destination it does not know:

1. Node checks its neighbour table (NT).
2. If an entry exists for this node, it updates the lifetime field else makes a new entry.
3. Receiver checks if other active entry exists with the same destination address in the NT;
4. Then receiver is a *JointNode* that joins the nodes that send the packets.
5. Calculate hop-gain and make entry into node's alternative route table.
6. Node sends *JointNode* update packet to the upstream node (the node with the least height value).

End of *JointNode* Notification

Algorithm II: *JointNode* Admission

This algorithm is executed by a receiver of JointNode update packet.

1. If sender is **not** already in the *JointNode* set,
2. If (maximum number of *JointNodes* is exceeded)
3. If (hop-gain of the received packet $>$ max hop-gain in the *JointNode* set)
4. Then replace the least hop-gain entry with this sender.
5. Else (Drop packet and send back-off message)
6. Else (add sender to the *JointNode* set).
7. Else (updates the lifetime field and return).

End of *JointNode* Admission

Algorithm III: *JointNode* Selection

This is executed by an upstream node that wants to repair a broken link

1. If the *JointNode* set is not empty
2. Return the node with the highest hop-gain, H_{max}
3. Else return (Null).

End of *JointNode* Selection

Algorithm IV: Link Merge Operation

This is executed by an upstream node that detects failure of its next hop link (or deliberately want to for other reason) while forwarding data

1. Start LOCAL_REPAIR_TIMER
2. If (***Algorithm III*** is ***not*** Null)

3. Unicast a link merge packet to the return of ***Algorithm III*** AND wait for a period of 2×(NODE_TRAVERSAL_TIME) + PROCESSING_TIME
4. If [no reply within (2×NODE_TRAVERSAL_TIME +PROCESSING_ TIME) AND LOCAL_REPAIR_TIMER is not expired]:
5. Delete node from *JointNode* set GO TO 2.
6. If (LOCAL_REPAIR_TIMER is expired): sends RERR message and return.
7. node replaces its NEXT link with this JointNode.
8. Sends new hello packet with its new height value for height sync.
9. Restore DATA packet transmission.
10. Else (sends RERR message and return).

End of Link Merge Operation

Algorithm V: Link Redundancy Deletion

When a node receives a new hello packet whose destination address exists in the route table:

1. If sender is neither the precursor (or PREVIOUS hop node) nor NEXT hop node
2. Then
3. If (sender's height, $h_s >$ Receiver's height, h_r) then;
4. Replace next hop with the sender's node
5. Else (Replace previous hop with the sender's node)
6. Send hello message with the new height value for height synchronisation
7. Continue to handle packet according to the default AODV hello mechanism

End of Link Redundancy Deletion

3.2.4. Example of the Proposed Route Maintenance

To demonstrate how the above mentioned route maintenance works, we illustrate a simplified scene of route changes as shown in the **Error! Reference source not found.**. In this example the node 0 is considered as a source who is continuously sending packets to node 4 with nodes 1, 2 and 3 as intermediate nodes. Assume each node to be stationary. Thus the S-D route looks like this; $\{0-1-2-3-4\}$.

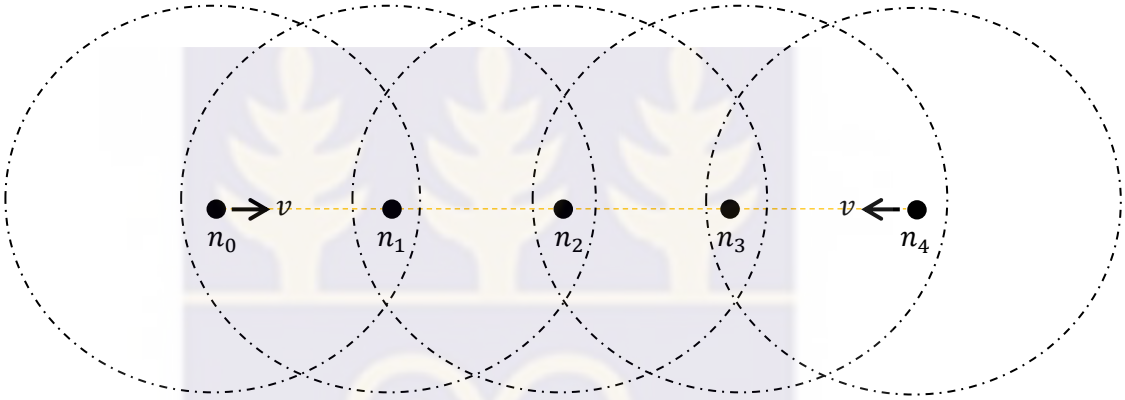

Fig 3.9: Demonstration of link merge and redundancy deletion repair operations.

If the source node, node 0, begins to move to the position of node 4 along the plane of the nodes' arrangement, then AODV typically will change the initial route to $\{0-3-4\}$ as it journeys to its destination. This is because all links remain active until the moving node 0 passes the entire coverage diameter of its next hop node 1 and loses the link with the node 1. At the same time, it enters into the coverage range of node 2 and 3 as shown it the figure. A new route search is committed which returns the least hop route $\{0-3-4\}$ over $\{0-2-3-4\}$.

However, for the proposed route maintenance scheme, node 0 will overhear node 2 which is neither the next nor previous node, and algorithm V is triggered, the *S-D route* become is $\{0-2-3-4\}$. Similar thing happens when node 0 comes into the

coverage of the node 3 directly. The proposed route maintenance detects the link redundancies and respond by deleting them off the *S-D route*. The route continued to modify itself dynamically without broadcasting a request packets from {0–1–2–3–4} to become {0–2–3–4}, {0–3–4} and {0–4} while it does not necessarily have to stop forwarding their packets. Similarly, a one plane voyage of the destination node 4 will see the *S-D route* changed from {0–4} to {0–3–4}, {0–2–3–4}, and finally {0–1–2–3–4} as time ticks. Of course, by virtue of the link merge operation. At first node 3 overhears both node 0 and 4 so triggers the algorithm 1 and hence node 0 has node 3 in its *JointNodes* set. On link failure, uses this active *JointNode* to merge the route back forming the {0–3–4}. The destination node 4 continues and hence node 2 comes in and the process continues.

3.3. Simulation Environment

This section describes the implementation of the proposed maintenance protocol in a simulation environment. The choice of simulation to test the protocol is not arbitrary as network simulator has become a valuable tool in research to develop, test, and diagnose network protocols. Simulation comes in with economic benefits and flexibility as developing and testing of network protocols can require a large set of actual hardware and infrastructure and may also require various conditions in which it can easily be tested by simulation. This study, for the very reasons stated above, can be tested and results can easily be analysed than experimental results because they are repeatable and data can easily be gathered for analyses in simulation.

The NS-2 simulator was used Fall & Varadhan (2011). NS-2 has been widely used in network protocol testing and development both in the academia and research. Through its packet tracing, the simulator generates trace files which collect simulation

data for results compilation. Performance of the proposed AODV route maintenance scheme in the various scenarios were evaluated. They were processed for the necessary information from the simulations and the performance metrics were then computed under those considerations.

Table 3.1 summarises the simulation details. Two parameters (the number of nodes and the mobility) were varied. Random waypoint (RWP) mobility model was used for representing nodes' mobility. For each scene, the simulation was carried out for five times and average of observed values were taken to reduce estimation errors.

Table 3.1: Simulation parameters

Simulation parameter	**Value**
Simulator	$NS\ 2\ (v.35)$
Transmitter range	$250\ meters$
Bandwidth	$2\ Mbps$
Interface queue length	50
Traffic type	CBR
Packet Rate	$4\ packets/s$
Packet size	$512\ bytes$
Number of traffic flows	5
Simulation time	$900s$
Topology size	$1000m\ \times\ 1000m$
Max-speed	$1, (5), 10, \ldots, 35\ ms^{-1}$
Number of nodes	$25, (50), 75, \ldots, 200$
Number of trials	5
Mobility model	RWP

3.3.1. Simulation Scenarios

The following scenarios were simulated:

1. Traditional AODV with the local Route maintenance (AODV) and
2. AODV with the proposed link merge and link redundancy deletion operation route maintenance scheme (AODV-LMD)

In each case, the parameters varied were

1. Mobility speed
2. Node density

Each case was ran 5 different times and the result averaged. In each case, the performance of the proposed system was analysed using the following performance metrics:

1. *Normalised routing overhead (NRO):*

   This metric compares the total number of control packets transmitted during the simulation time compared to the total data traffic generated. In this case, each packet sent over a hop is counted one and hence for a multiple hops transmission, each hop transit is considered as one count. It is calculated as the total number of routing control packets sent by all nodes divided the number of data packets successfully received at destination. Control packets consist of RREQ, RREP, RERR and the Hello packets in AODV protocol. However, in the case of the proposed protocol it includes the additional defined packets to update the *JointNode* information.

$$NRO = \frac{\sum Routing\ control\ packets}{\sum Data\ packets\ delivered}$$

A lower NRO is preferred to a network since a high NRO will mean much of the bandwidth is consumed in the name of maintaining the network instead of using to transmit data packets.

2. *Packet delivery ratio:*

   Packet delivery ratio (PDR) is defined as the total amount of data received at all the destinations divided by the total amount of data transmitted during the simulation. A high packet delivery ratio is desired in a network.

$$PDR = \frac{\sum Data\ packets\ received\ by\ the\ destination}{Data\ packets\ send\ by\ the\ source}$$

3. *Throughput*

   Throughput is the average time rate of data packets transferred successfully by a sender to the receiver. It is calculated as the total number of data packets received at all destinations over the total time.

$$Throughput = \sum \frac{Amount\ of\ packets\ received\ at\ the\ destination}{Total\ time}$$

4. *End-to-end delay (Average delay)*

   End-to-end delay is the average time difference between the time a data packet is sent by the source node and the time it is successfully received by the destination node. In a network, a lesser end-to-end delay is preferred.

$$End - to - end\ delay = \frac{1}{n}\sum_{n=i}^{N}(TR_n - TS_n)$$

Where TS_n is the packet sent time,

TR_n is the packet received time and

N is the total number of data packets received.

3.4. Chapter Summary

In this chapter, the scientific research approached adopted for this study and the proposed solution to the problem of frequent route break and search were been discussed. The study used simulation method in order to meet the demand of a large set of network node and the right conditions to test the different network scenarios.

The chapter also proposed two local repair operation, link merge and link redundancy deletion operations, to maintain the route returned by the route request phase. The link merge operation uses the identified *JointNodes* to merge a node-to-node link that fades within an active route. This process saves the route from failing just because one of its links in the route's set of links failed. The second operation, link redundancy deletion, shortens the route length dynamically without route search phase. The two operations help to minimise route breaks in an active route for on-demand routing protocol. The chapter also analysed how the route length may be affected by the operation with the hop-gain. This is a priority cost benefit analyses where nodes could determine the effect of the operation on the route length before committing it hence can be used to optimise the said route even when there is no risk of route break.

Finally, the chapter concluded with the simulation and how the data collected from the simulation was analysed. The network scenarios were varied in two dimensions – nodes' mobility and density.

CHAPTER FOUR

SIMULATION RESULTS AND FINDINGS

4.1. Introduction

The chapter presents and discusses the results of the simulation. It evaluates the performance of the proposed AODV modification compared to AODV with its local repair mechanism as base routing protocol. The main focus is to demonstrate how mobility and density of the nodes impact on the network metrics of AODV routing protocol and how effective the proposed modification mediates. The network metrics considered here includes how the available bandwidth is utilised to transmit data packets, how it avoids the unnecessary packet drops and delays when repairing a lost link and amount of data transferred in a unit time.

4.2. Simulation Results and Analysis

The evaluation of the proposed route maintenance operation was conducted under two different network conditions – *node density* and *node mobility*. The detailed simulation parameters were mentioned in table 3.1. Five simulations were run for the same parameters and average values were taken for each scenario to reduce estimation error. The simulation results were obtained for AODV and modified AODV.

4.2.1. Impact of Node Density

The node density was varied, 25, 50, 75, 100, 125, 150, 175 and 200 nodes at a fixed velocity of 10 ms^{-1}. The impact of the node density on the two protocols are shown in Fig 4.1- Fig 4.4

Normalised Routing Overhead

In the Fig 4.1, normalised routing overhead (NRO) is plotted against the node density for both the original and the proposed AODV. NRO is observed to increase with node density for both the original and modified AODV mechanism. This is expected because more nodes mean unnecessarily repeat of broadcast of the control packets by each node, hence more control packets are committed during the route search phase.

The graph also shows a clear performance advantage in the proposed method. The ratio of the generated control packets to data packets delivered to the destination were more in the original method than in the case of the proposed method as the node density increases. Routing overhead must have increased in the AODV than the modified-AODV because of the extra reroute control packets that AODV will commit to re-establish the route. On the other hand, the link merge mechanism must have committed less control packet in its operation. The result of re-establishment which generates packets that span across the entire network in AODV caused the considerable increase in the route overhead packets compared to the node-to-node packet exchanged to repair a failed link in the modified-AODV.

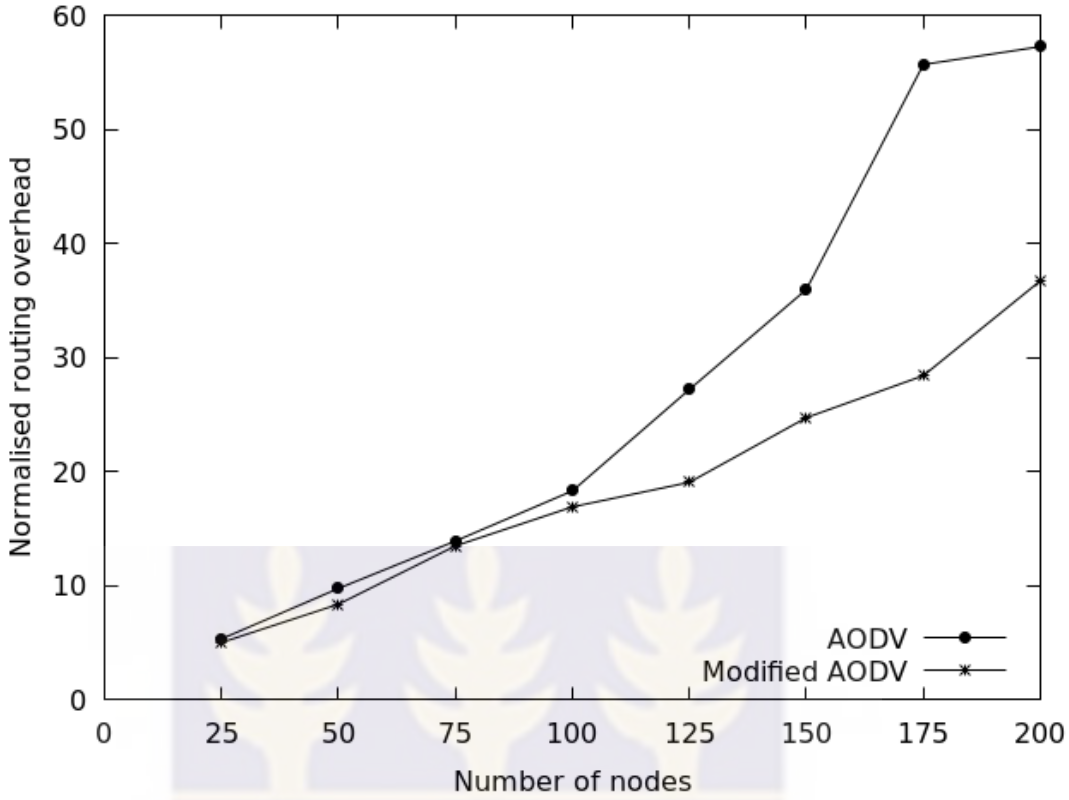

Fig 4.1: Normalised routing overhead vs. nodes' density

Packet delivery ratio

The Fig 4.2 shows the packet delivery ratio (PDR) for varied node density. It can be observed in the figure that PDR increased nominally to a point and then decreased with varying node density. The changes were marginal for the modified AODV but significantly dropped at high density.

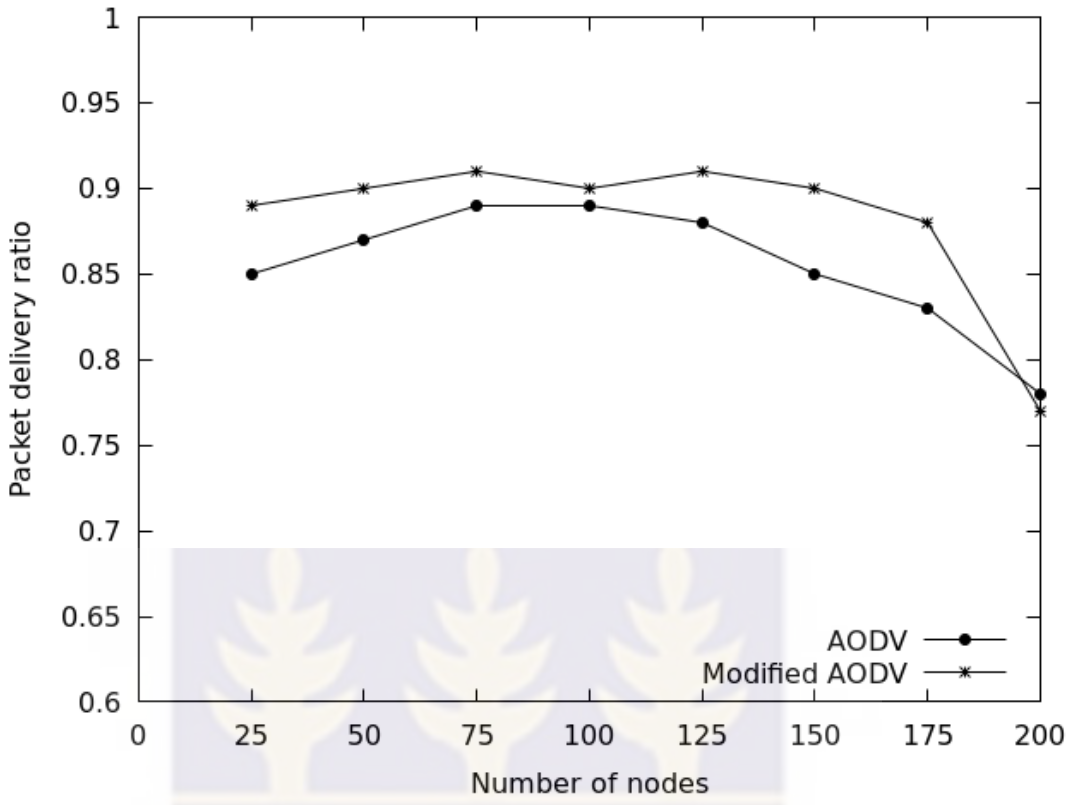

Fig 4.2: Packet delivery ratio vs. node density

Though marginal, the graph (Fig 4.2) shows that the proposed method has more packets successfully delivered when the nodes were less than 175 yet they reduced significantly when the number of nodes increased further. AODV original marginally outperformed the modified version at the highest node density. This might be the cause of increase in channel contention at the S-D route node(s) neighbourhood as a result of the several nodes identifying themselves as *JointNode* and triggering the *JointNode Nomination algorithm* which sends *JointNode* notification packet. Though the S-D route node sends a back-off signal (in the *JointNode Admission algorithm*) to node new *JointNodes* to back-off for 3 seconds after its *JointNode* set exceeds the maximum limit, the several neighbours present in a link's vicinity might overwhelm the upstream nodes with their notification packets causing a layer 2 collision, a bottleneck for packet

reception. The number of dropped packets at the upstream node therefore increases. Besides, for same reason packets might be buffered at the interface queue, which can overrun as they await processing. This dropped packets translate to the low PDR.

Throughput

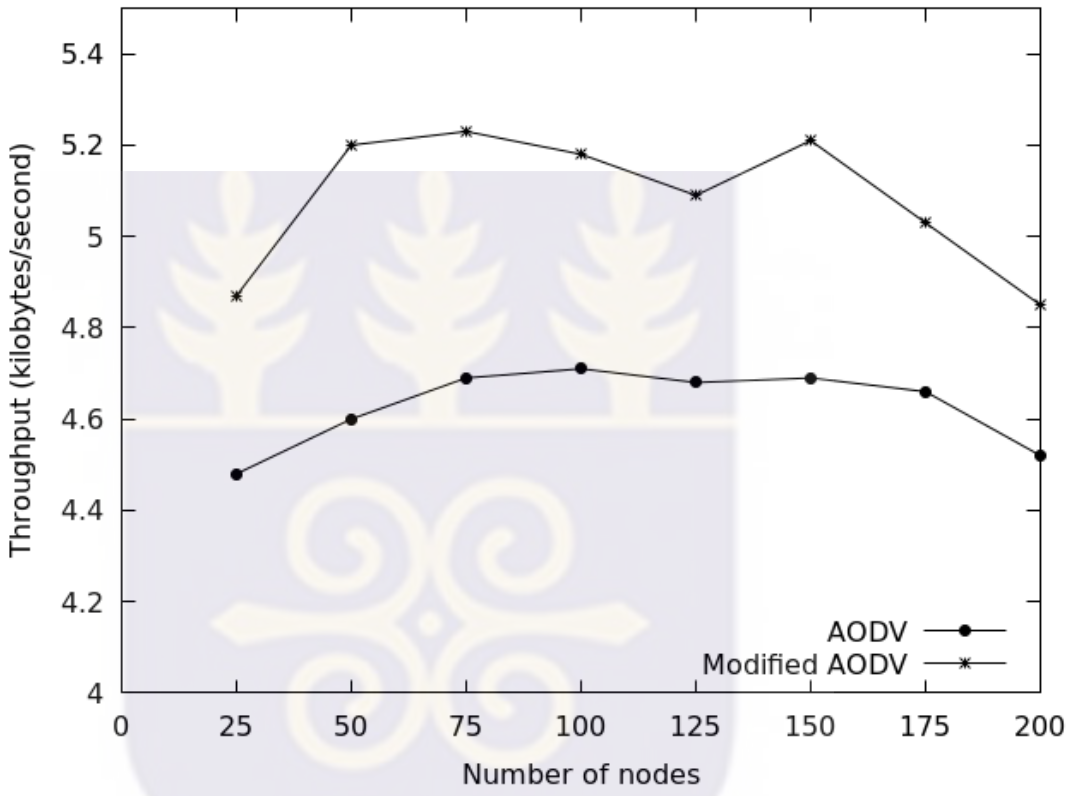

Fig 4.3: Throughput vs. node density

The Fig 4.3 is a plot of the throughput against the network density. The graph shows a marginal increase in the throughput with the node density to a point and begins to decrease. The modified AODV is observed to have a higher throughput in density variations than the normal AODV. The proposed method uses a neighbour with the highest potential to continue forwarding the packets. The figure also indicates a significant drop in the throughput of modified-AODV when the nodes exceeded 150. This could be the cause of the several nodes that triggers *JointNode* notification

updates to the upstream nodes of the individual links on the S-D route. Yet, the modified version still performed much better than the original AODV in the dense network and this is due to the significant reduction of the frequency of route search and dropped packets.

End-to-end delay

Fig 4.4 measures the average end-to-end delay of data packets received at the destinations. It is observed that end-to-end delay increases when node density increases. The results also suggest that modified-AODV beats AODV substantially when node density increases. Thus, modified-AODV has shown that packets from source to the destination are less delayed, though under highly dense situation it saw a significant rise in delay.

Due to high channel contention caused by excessive retransmissions of route request packets, and specifically in modified-AODV, where more neighbours exist as *JointNodes,* they will all trigger the *JointNode* nomination packets to the upstream node. The packet wait time at the interface queue are prolonged as some packets are received into the buffer to await for processing time. Probability of packet collisions causing layer 2 retransmission and contention delay is also high. Therefore the significant reduction of routing overhead by modified-AODV must have been translated into better end-to-end delay in the dense networks as shown in the figure.

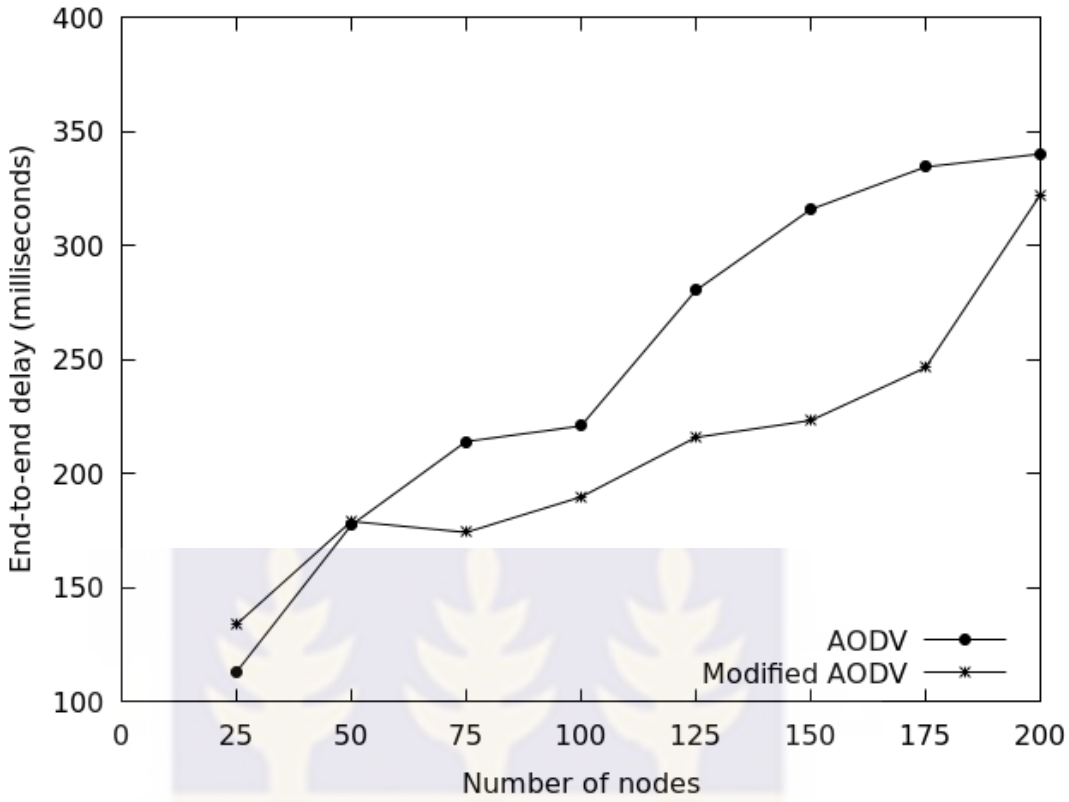

Fig 4.4: End-to-end delay vs. node density

4.2.2. Impact of Node Mobility

This section presents the effect of node mobility on the two protocols. Nodes' maximum velocities were stepped 1, 5, 10, 15, 20, 25, 30 and 35 ms^{-1} for a fixed number of nodes (50 nodes). The results are shown in Fig 4.5 - Fig 4.8 for varying node mobility.

Normalised Routing Overhead

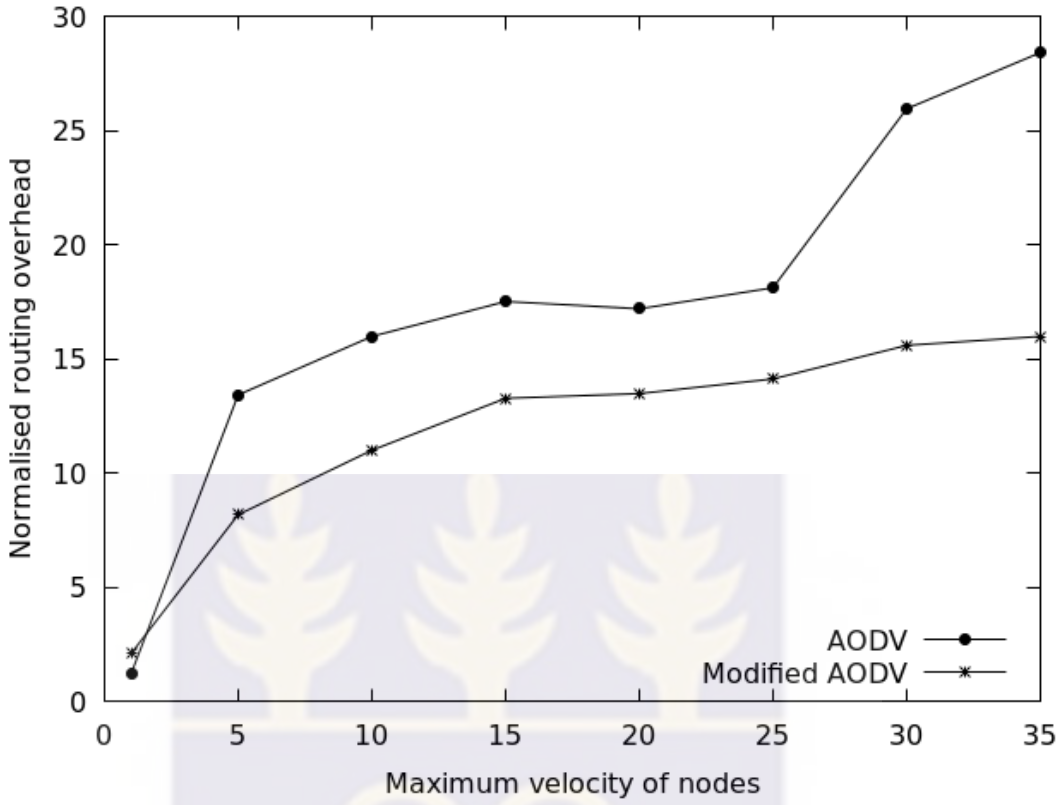

Fig 4.5: Normalised routing overhead vs. nodes' mobility

The Fig 4.5 shows a plot of NRO with mobility. NRO is observed to increase with mobility increase. It is observed that the existing method has nominally less NRO than that of the proposed method for much less mobility (when the maximum velocity was $1ms^{-1}$) however the proposed method showed a significantly less overhead to data ratio as mobility increased.

For a much less mobility, node-to-node link change occurs rarely. The frequency of network topology change is low and few route repairs are observed if not none at all. However, the proposed link repair still incurred the additional overhead to identify *JointNodes*. As mobility becomes higher, it causes a significant number of links to change. That increases the RREQ packets generated and disseminated by AODV to re-establish paths whereas the modified AODV will establish the paths without

committing a global route RREQ packets. These activities potentially caused the increase in the overall routing overhead but it is much better for the proposed scheme.

Packet delivery ratio

Fig 4.6 shows the comparison of the packet delivery ratio of original AODV and the modified-AODV. The results show that PDR decreased significantly with increasing velocity. This is because faster mobility of nodes causes more node-to-node link failures. For the comparison, result depicts the modified-AODV has more packets delivery success and seemed to normalised as the velocity increase whereas the AODV line continues to fall with increasing velocity. The difference has resulted from the way the two protocols maintain the S-D route. This is because broken links are repaired using *JointNode* in modified-AODV so nodes don't necessarily have to drop the packet because a link failed, hereafter it restores data delivery even when break is far from destination. Therefore, more data sent from the source node is successfully delivered to the destination node. AODV would merely drop packets if the break point occurred at a point far away from the destination. Further, a route repair failure results in packet drops and thus low packet delivery ratio. Since the proposed repair algorithm uses a carefully selected neighbour node to repair the route it is likely to succeed.

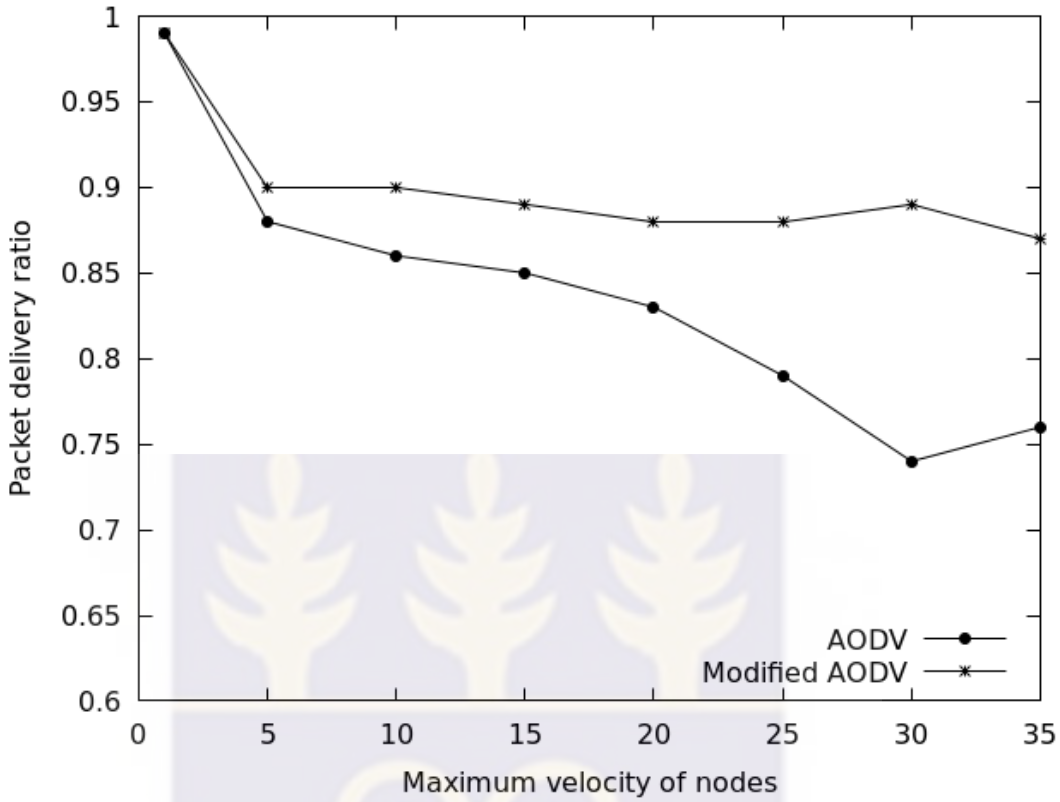

Fig 4.6: Packet delivery ratio vs. node mobility

Throughput

The Fig 4.7 shows the total data packet that was successfully delivered to the individual destinations over the total time. The AODV with the merge and redundancy deletion route maintenance mechanism is observed to have higher throughput in the mobility variation. This is because mobility increase causes frequent and unpredictable link change, whereas the proposed method uses the best possible neighbour to continue forwarding data packets, the original AODV may rather just drop the packets and rediscover the route. The superiority over the original AODV becomes noticeable at all velocities except $1ms^{-1}$ which will behave like a static topology.

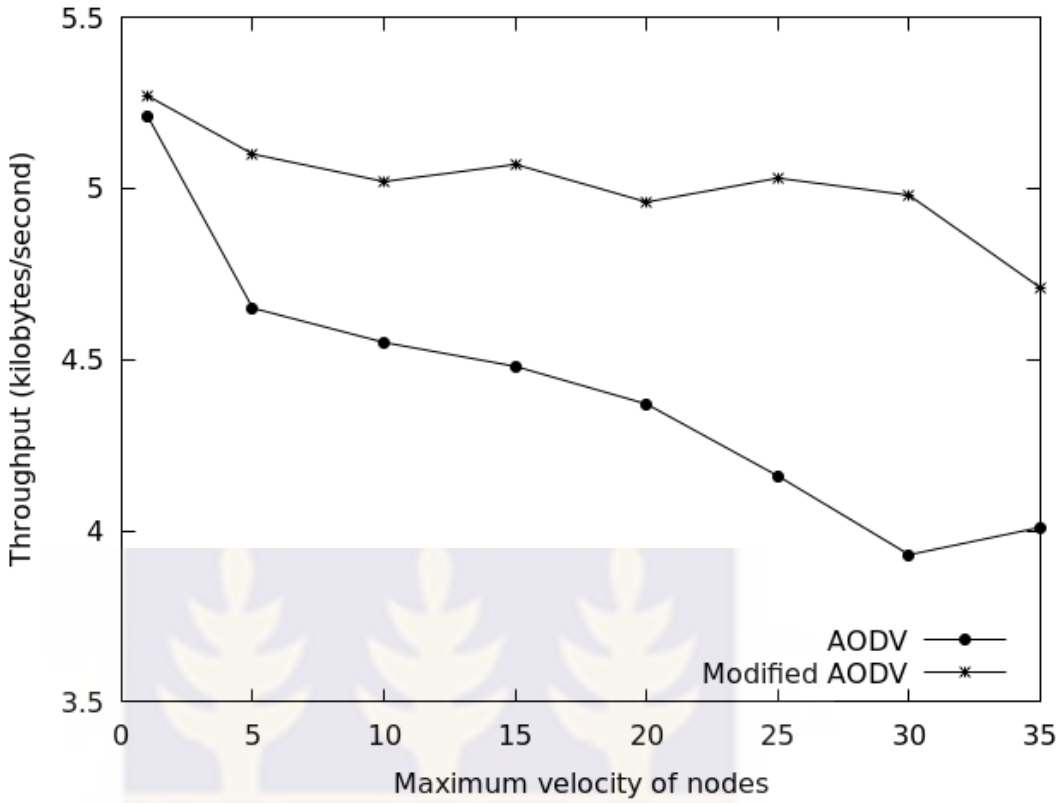

Fig 4.7: Throughput vs. node mobility

End-to-end delay

In Fig 4.8 the average delay of the data packets from source to the destination is shown. The average end-to-end delay increased when nodes' maximum velocity increases for both protocols but a comparison of the two showed the modified-AODV has a substantially lesser delay than its counterpart. The two almost performed same until the mobility was relatively high, however the performance difference became more profound at high speeds. Thus, modified-AODV has shown that packets from source to the destination are less delayed under high mobility.

This is observed because more link change occur within routes for fast moving nodes and that the protocol needs to restore. Fast moving nodes quickly run out of coverage of each other and that the protocol needs to restore when they fade. Modified-AODV merges the links using the *JointNodes* it has kept in its *JointNode* set and upon

success, transmission is restored with less delay as compared to original AODV that invokes the reroute search.

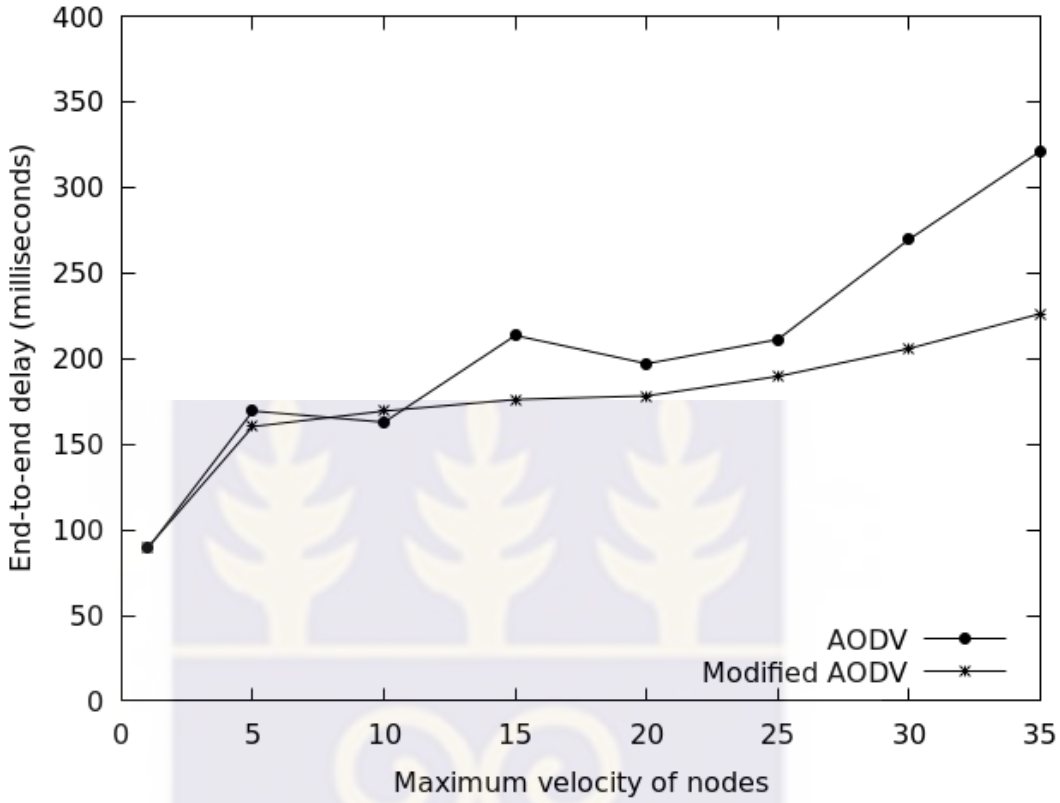

Fig 4.8: End-to-end delay vs. node mobility

4.3. Chapter Summary

This chapter has extensively evaluated the two protocols, original AODV (with its local route repairs) and the proposed AODV (with the proposed link merge and link redundancy deletion route maintenance scheme). The chapter has presented the simulation results. It was proposed that AODV can know the neighbour which can repair a broken link when a link failure occurs and also when a link redundancy occurs. The performance of the proposed AODV modification has been evaluated and compared with the original AODV and the simulation results suggest the proposed route maintenance algorithm performs well. The differences in the network which is

local repair processes has shown a significant decrease in the normalised routing overhead packets and delay. The results show a significant increase in the throughput and a packet delivered in the network in the proposed scheme. It improves S-D route's lifetime and the networks capacity to support QoE.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1. Introduction

The chapter presents a brief summary on the study and the findings, and finally suggest directions for future work.

5.2. Summary of the study

Mobile ad hoc networks which started in the military field applications have become very popular area of interest to modern researchers because of the many applications in the civil society where a quick deployment is important over reliability. Yet there are a lot of challenges present in this area. The traditional wired network solutions cannot fit well with MANET and requires specialised solutions to its own problems. Due to the limited constraints which includes limited wireless radio transmission range, no fixed motion and boundary, limited bandwidth, and others of the individual nodes involved in MANET, much research effort has been dedicated to finding solutions to these challenging issues over the recent past.

A number of routing protocols have been proposed for MANET and can be categorized into proactive and on-demand routing protocols. The proactive conventions periodically find routes and maintain them, thus there is a route to all destinations making it unscalable when the network size becomes large. In such situations, there is an excessive route overhead associated with the periodic broadcast of the route updates between the nodes of the network. Alternatively, on-demand protocols have only the source invoking the route discovery to find the route to the destination. It does that only when it has a data send request. They are much more

scalable. Nonetheless, they also incur a punitive cost of excessive retransmission of the route search packets.

AODV which is an example of on-demand routing protocol has two phases; route discovery and route maintenance. In an attempt to reduce the excessive retransmission of the control packets, several study efforts modify the route discovery phase which blindly floods the network with route request packets. Yet other works attempt to prevent several route searches by ensuring long-live routes through a well-planned route maintenance phase. This in effect reduces the frequency of the route search. This paper is an example of the later which attempts to prevent frequent route breaks.

It was therefore proposed in this paper that a returned route be guarded against frequent link breaks through route maintenance mechanism to minimise global route search. Specifically, the hop-to-hop linked nodes were made to identify nodes that they share in their 1-hop neighbour (herein referred to as *JointNode*) and keep them as backup to the link nodes and when node notices a link failure, they can select from that pool (*JointNodes set*) to locally repair the link without rediscovery of this route entirely. In addition to this, the paper suggests a mechanism to eliminate redundant node(s) in a given active route whenever an upstream node notice that it can reach out to a downstream node directly without the need of its intermediate node(s). To achieve these, the periodic broadcast of "hello" packets between neighbouring nodes, an already existing mechanism in the AODV, was used.

To demonstrate the performance of the proposed route maintenance scheme, the AODV module of the NS-2 simulator was used. The measured network performance metrics of the modified version of the AODV were compared to the original AODV with its local route repair scheme activated.

5.3. Summary of Findings

The main objective of this study has been to design and analyse a route maintenance scheme that minimize route breaks so as to prolong the lifetime of an active route for an on-demand protocol of MANET. This was to significantly improve the network performance metrics (normalised routing overhead, packet delivery ratio, average end-to-end delay and throughput as defined in chapter 3). The analysis of the proposed route maintenance scheme compared to the local repair of the default AODV for a range of system parameters (specifically node density and mobility) were carried in the NS-2 simulator. In all cases of varying mobility considered, AODV with the modified maintenance scheme exhibited a good performance gain than the default scheme, especially, when the mobility was high.

For increasing node density, almost all cases considered, modified-AODV exhibited higher performance gain in terms of the measured metrics compared to the traditional AODV with its local route repair scheme. However, in a heavily dense network, low PDR were recorded for the proposed scheme. The number of nodes identifying themselves to be a shared node for a link became many. All these nodes will attempt to signal the upstream node of a given link and this may overwhelm the said node with their *JointNode Nomination* packets. This resulted in a marginally high degree of packet drop which consequently resulted in the low PDR recorded for the proposed scheme.

5.4. Suggestions for Further Work

The study proposes some areas for further research which include the following:

- Further work is needed to evaluate the cause of the increase in the drop packets and mitigates it. The packet delivery ratio was observed to have increased with the node density, however it tends to drop as the number of nodes continue to

increase to a highly dense environment when evaluating the protocols. We implemented a *JointNode* back-off mechanism to silence nodes from overwhelming the upstream nodes with their *JointNode* trigger message for a period of 3 seconds. This time value was arbitrary, it is suggested that further study be done to evaluate both the mechanism to minimise the neighbour nodes overwhelming the upstream node with these *JointNode* trigger messages.

- This proposed mechanism also suggested a way to optimise the route dynamically by incorporating other network sensitive parameters such as energy, link status, congestion, distance, etc. in order to reconfigure already established route to a local best route dynamically without necessarily resorting to the route rediscovery process. Future study should consolidate these quality measures into this route maintenance scheme to develop a more comprehensive model to feature load balancing and acceptable thresholds for path optimisation.
- It is also suggested the suitability of the proposed method be further studied by testing it with other network scenarios like varying packet rates.
- The study implemented the said scheme via computer simulation only, it is therefore suggested that proposed method be implemented in a real system.

REFERENCES

Abdulai, J., Ould-Khaoua, M., Mackenzie, L., & Mohammed, A. (2008). Neighbour coverage: a dynamic probabilistic route discovery for mobile ad hoc networks. *International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS 2008*, (pp. 165-172). Edinburgh, Scotland.

Abdulai, J. (2009). Probabilistic Route Discovery for Wireless Mobile Ad Hoc Networks (MANETs). *Ph.D Thesis submitted to University of Glasgow*. The Faculty of Information and Mathematical Sciences.

Abhilash, P., Perur, S., & Iyer, S. (2002). Router Handoff: An Approach for Preemptive Route Repair in Mobile Ad Hoc Networks. In S. S., P. V. K., & S. U. (Eds.), *High Performance Computing - HiPC* (pp. 347–357). Springer, Berlin, Heidelberg. doi:https//doi.org/10.1007/3-540-36265-7_33

Agarwal, M. M., Govil, M. C., & Sinha, M. (2016). DPAODV–A Dynamic Probabilistic-based Energy Efficient Routing Protocol for MANETs. *International Journal of Applied Engineering Research, 11*(6), 4024-4030.

Ahmed, D. E., & Khalifa, O. A. (2017). A Comprehensive Classification of MANETs Routing Protocols. *International Journal of Computer Applications Technology and Research, 6*(3), 141-158.

Akansha, C., & Sharma, V. (2016). Review of Performance Analysis of Different Routing Protocols in MANETs. *International Conference on Computing, Communication and Automation (ICCCA2016)*, 541-545.

Alslaim, N. M., Alaqel, A. H., & Zaghloul, S. S. (2014). A Comparative Study of MANET Routing Protocols. *IEEE*, 178-182.

Amaldi, E., Capone, A., Malucelli, F., & Signori, F. (2002). UMTS radio planning: optimizing base station configuration. *IEEE Veh. Conf.*, *2*, pp. 768-772.

Asadi, A., Wang, Q., & Mancuso, V. (2014). A Survey on Device-to-Device Communication in Cellular Networks. *16*(4), 1801 - 1819. doi:10.1109/COMST.2014.2319555

Ball, M. G., Qela, B., & Wesolkowski, S. (2011). A Review of the Use of Computational Intelligence in the Design of Military Surveillance Networks. *Springer-Verlag Berlin Heidelberg*.

Bekmezci, I., Ozgur, K. S., & Samil, T. (2013). Flying Ad-Hoc Networks (FANETs): A survey. *Ad Hoc Netw.*, 1254 -1270. doi: http://dx.doi.org/ 10.1016/j.adhoc.2012.12.004

Bisengar, A., Ouadoudi, Z., Nourddine, E., Mohamed, R., & Mohamed, O. (2012). Adaptive Velocity and Distance Based Routing Protocol for MANET. *Journal of Theoretical and Applied Information Technology*, *45*(2), 397-405.

Cerri, D., & Ghioni, A. (2008). Securing AODV: The A-SAODV Secure Routing Prototype. *IEEE Communication Magazine*.

Du, Q., Zhu, j., & Zhang, E. (2010). A Novel AODV Routing Algorithm Based on Local Route Maintenance in Tactical MANETs. *Information Science and Engineering (ICISE), 2010 2nd International Conference.* IEEE.

Fall, K., & Varadhan, K. (2011). The ns Manual. *The VINT Project.* Retrieved from http://www.isi.edu/nsnam/ns/ns-documentation.

Goff, T., Abu-Ghazaleh, N. B., Dhananjay, P. S., & Kahvecioglu, R. (2001). Preemptive Routing in Ad Hoc Networks. *ACM SIGMOBILE,* 43-52.

Haas, J. Z., & Pearlman, R. M. (1998). The performance of query control schemes for the zone routing protocol. *ACM SIGCOMM, 1998 conference on Applications, technologies, architectures, and protocols for computer communication*, (pp. 167-177). Vancouver, British Columbia, Canada. doi:10.1145/285237.285279

Hoebeke, J., Moerman, I., Dhoedt, B., & Demeester, P. (2004). An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *Journal of of Communications Networks*, 60-66. Retrieved from cwi.unik.no/images/Manet_Overview.pdf

Issariyakul, T., & Hossain, E. (2012). *Introduction to network simulator NS2* (2nd ed.). New York: Springer. doi:DOI: 10.1007/978-1-46141406-3

Jain, J., & Bandhopadhyay, R. G. (December 2011). On Demand Local Link Repair Algorithm for AODV Protocol. *International Journal of Computer Applications, Volume 35*(5), 20-25.

Johnson, D., & Maltz, D. (1999). *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks.* IETF Draft.

Johnson, D., Hu, Y., & Maltz, D. (2007). *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4.* IETF Mobile Ad Hoc Networking

Working Group INTERNET DRAFT. Retrieved from http://www.ietf.org/rfc/rfc4728.txt

Kamalakkannan, P. S. (2012). Enhancing Route Maintenance in RSEA-AODV for Mobile Ad Hoc Networks . *IEEE* , 464-469.

Kant, L., Young, K., Younis, O., Shallcross, D., Sinkar, K., Mcauley, A., Manousakis, K., Chang, K., & Graff, C. (2008). Network science based approaches to design and analyze MANETs for military applications. *IEEE Communications. IEEE Communications Magazine* *46*(11) 55 - 61 **DOI:** 10.1109/MCOM.2008.4689245

Kawish, B. S., Aslam, B., & Khan, S. A. (2008). Reduction of Overheads with Dynamic Caching in Fixed AODV based MANETs. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, 2*(4), 688-693.

Khana, A. u., Bilalb, S. M., & Othmana, M. (2013). A Performance Comparison of Network Simulators for Wireless Networks. Retrieved from https://arxiv.org/abs/1307.4129

Kim, J.-s., Zhang, Q., & Agrawal, D. P. (2004). Probabilistic broadcasting based on coverage area and neighbor confirmation in the mobile ad hoc network. *Globecom 2004 Workshops.* IEEE Communications Society.

Kumar, R., Kumar, S., Pradhan, S. P., & Yadav, V. (2011). Modified route-maintenance in AODV Routing protocol using static nodes in realistic mobility model. *International Journal on Computer Science and Engineering (IJCSE), 3*(4), 1554-1562.

Lai, W. K., Lin, S.-Y. H., & Yuh-Chung. (2007). Adaptive backup routing for ad-hoc networks. *Computer Communications, 30* , 453–464.

Latiff, L. A., Ali, A., Chia-Ching, O., & Fisal, N. (2005). Location-based Geocasting and Forwarding (LGF) Routing Protocol in Mobile Ad Hoc Network. *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ELearning on Telecommunications Workshop*. Lisbon, Portugal: IEEE. doi:10.1109/AICT.2005.55

Lee, S.-J., & Gerla, M. (2000). AODV-BR: Backup Routing in Ad hoc Networks. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 1311-1316.

Li, Z., & Yang, X. (2016). AReliability-oriented Web Service Discovery Scheme with Cross-layer Design in MANET. *2016 IEEE International Conference on Web Services* (pp. 404-411). San Francisco, CA, USA: IEEE. doi:10.1109/ICWS.2016.59

Padmini Misra. (2016, 4 17). *Routing Protocols for Ad Hoc Mobile Wireless Networks*. Retrieved from http://www.cse.wustl.edu/~jain/cis788-99/ftp/adhoc_routing/

Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. *Proceedings of ACM SIGCOMM'94*, (pp. 234-244).

Perkins, C. E., & Royer, E. M. (1999). Ad hoc On Demand Distance Vector Routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, (pp. 90-100). New Orleans, LA.

Perkins, C., & Royer, B. E. (2003). *RFC3561: Ad hoc On-Demand Distance Vector (AODV) Routing.* IETF.

Qingsong, D., & Eryang, Z. j. (2010). A Novel AODV Routing Algorithm Based on Local Route Maintenance in Tactical MANETs. *IEEE*.

Raich, A. R., & Vidhate, A. (2013). LOCATION AWARE MODIFIED AODV TO SELECT BEST PATH. *International Journal of Research in Advent Technology (IJRAT), 1*(1).

Sharmaa, D. K., Patraa, A. N., & Kuma, C. (2016). An improvement in performance of mobile ad hoc networks using modified route maintenance,. *Computers and Electrical Engineering*.

Sooriyaarachchi, S. J., Fernando, W. A., & Gamage, C. D. (2016). Anaylisis of packet flooding in dense MANETs using a probabilisitc model. *International Conference on Advanced Communication Technology (ICACT), 2016 18th* (pp. 39-45). IEEE. doi:10.1109/ICACT.2016.742366

Truong, H.-L., & Dustdar, S. (2012). Services-Oriented Architecture for Mobile Services. In K. Anup, & B. Xie, *Handbook of Mobile Systems Applications and Services.* CRC Press.

Wu, Z. (2011). A Novel Scheme of Dual Backup Routing in Ad Hoc Networks. *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*

Intelligent Systems and Applications. Wuhan, China: IEEE. doi:10.1109/ISA.2011.5873457

Xu, H., Wu, X., Sadjadpour, H. R., & Garcia-Luna-Aceves, J. (2010). A Unified Analysis of Routing Protocols in MANETs. *IEEE TRANSACTIONS ON COMMUNICATIONS, 58*(3).

Zhou, M.-T., & Harada, H. (2012). Cognitive maritime wireless mesh/ad hoc networks. *Journal of Network and Computer Applications, 35*(2), 518-526. Retrieved from https://doi.org/10.1016/j.jnca.2010.12.018

Zhuang, L., Liu, Y., & Liu, K. (2009). Hybrid gateway discovery mechanism in mobile ad hoc for internet connectivity. *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. 30*, pp. 143-147. ACM.

APPENDIX A

1. Source Code for sending JointNode Notification

```
void
AODV::sendNom() {
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_reply *rh = HDR_AODV_REPLY(p);
int Height_Difference = abs(getHeight(id,nb1) –
                        getHeight(id,nb2)); //absolute value

#ifdef DEBUG
fprintf(stderr, "sending nomination packets from %d at
%.2f\n", index, Scheduler::instance().clock());
#endif // DEBUG

 rh->rp_type = AODVTYPE_NOM;
 // rh->rp_flags = 0x00;
 rh->rp_hop_gain = (Height_Difference-2);   //H = (k – i) – 2
 rh->rp_dst = index;
 rh->rp_dst_seqno = seqno;
 rh->rp_lifetime = (1 + ALLOWED_NOM_LOSS) * NOM_INTERVAL;

 // ch->uid() = 0;
 ch->ptype() = PT_AODV;
 ch->size() = IP_HDR_LEN + rh->size();
 ch->iface() = -2;
 ch->error() = 0;
 ch->addr_type() = NS_AF_NONE;
 ch->prev_hop_ = index;          // AODV hack

 ih->saddr() = index;
 ih->daddr() = IP_BROADCAST;
 ih->sport() = RT_PORT;
 ih->dport() = RT_PORT;
 ih->ttl_ = 1;

 Scheduler::instance().schedule(target_, p, 0.0);
}
```

2. Source Code for receiving JointNode Notification

```
void
AODV::recvNom(Packet *p) {
//struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
AODV_Neighbor *nb;
```

```
nb = nb_lookup(rp->rp_dst);
 if(nb == 0) {
   nb_insert(rp->rp_dst);
 }
 else {
   nb->nb_expire = CURRENT_TIME +
                   (1.5 * ALLOWED_NOM_LOSS * NOM_INTERVAL);
 }

 Packet::free(p);
}
```

3. A method for calculating the height of the node on the route

```
int
getHeight(nsaddr_t id, AODV_Neighbor nb){
  int counter =0;
  AODV_Precursor *pc = rt_pclist.lh_first;

  for(; pc; pc = pc->pc_link.le_next) {
    counter++;
     if(pc->pc_addr == id)
    return counter;
  }
   return NULL;
}
```

4. Modified version of the recvHello method of the AODV

```
void
AODV::recvHello(Packet *p) {
//struct hdr_ip *ih = HDR_IP(p);
bool isRouteVariable = false;
struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
aodv_rt_entry *rt = rtable.rt_lookup(ipdst);
AODV_Neighbor *nb;

 nb = nb_lookup(rp->rp_dst);
 // This should look for the next and previous node
 if(nb != null){
      int nb_height = getHeight(rp->rp_dst,nb->id);
      //nb_height = rp->rp_hop_count;
      int node_height = getHeight(rp->rp_dst,this->id)
      int hop_gain =  std::abs(node_height- nb_height)-1;
```

```
   if(nb_height > node_height+1)
   {
      //Downstream neighbour, hence my hop-count unaffected
     this->next_hop_ = rp->rp_src;
      rt_update(rt,rp_dst_seqno,this->rp_hop_count, nb->id,
      rp->src,(CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT));
   }else if(nb_height< node_height-1 ){
      //Upstream neighbour, hence my hop-count is affected
     this->prev_hop_  = rp->rp_src;
     rt_update(rt,rp_dst_seqno,this->rp_hop_count+hop_gain,
      nb->id,rp->src,(CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT));
   }
 }

 if(nb == 0) {
     isRouteVariable =true;
   nb_insert(rp->rp_dst);
 }
 else {
   nb->nb_expire = CURRENT_TIME + (1.5 * ALLOWED_HELLO_LOSS *
   HELLO_INTERVAL);
 }
 if(isRouteVariable ==true)
     aodv_nomination(*p);
 Packet::free(p);
}
```

5. JointNode Set Cache Entry

```
class JointNodeSet {
        friend class AODV;
        friend class aodv_rt_entry;
   public:
        JointNodeSet(u_int32_t a) { jt_addr = a; }
   protected:
        LIST_ENTRY(JointNodeSet) nb_link;
        nsaddr_t   jt_addr;
        double    jt_expire;      // ALLOWED_HELLO_LOSS *
                                     HELLO_INTERVAL
};
LIST_HEAD(aodv_ncache, JointNodeSet);
```

6. **Source Code for JointNode Selection**

```
// A method that returns a node from the JointNode Set
nsaddr_t
getJointNode(){
  nsaddr_t myHighest_HopGain_node;
  int Highest_HopGain = -2;

  //Loop through the JointNode Set
  rt = JointNodeSet.head()
  for(nsaddr_t r : rt)
  {
      //select from JointNodeSet where hop-gain is greatest.
      //return node_id;
      if (Highest_HopGain < r.getHopGain()) {
         Highest_HopGain = r.getHopGain;
         myHighest_HopGain_node = r. jt_addr;
      }
  }
  return myHighest_HopGain_node;
}
```

7. **Source Code for merging the Link using a JointNode**

```
void
aodv_link_merge(aodv_rt_entry *rt, Packet *p)
{
     #ifdef DEBUG
       fprintf(stderr,"%s: Dst - %d\n", __FUNCTION__, rt->rt_dst);
     #endif

     // Buffer the data packet
     rqueue.enque(p);

     // mark the route as under repair
     rt->rt_flags = RTF_IN_REPAIR;

     // set up a timer interrupt
     Scheduler::instance().schedule(&lrtimer, sendMergePacket,
     rt->rt_req_timeout);
    lrtimer = Scheduler::instance().clock();
```

```
    do
    {
        if(getJointNode() != null){
            jtnode = getJointNode();
            sendMergePacket(
                    nsaddr_t ipdst, // Route destination IP adr
                    jtnode,         // IP packet destination
                    dst_seqno,
                    lifetime,
                    timestamp,
                    hop_count       // Hop Count
                );
            Packet::free(p);
            //set up a timer interrupt
            wait_time = (2*NODE_TRAVERSAL_TIME+0.01)
            Scheduler::instance().schedule(wait_time,
            sendMergePacket, rt->rt_req_timeout);
            jt_repair_expire = Scheduler::instance().schedule();
        }
    }while(!lrtimer && jt_repair_expire);
    if(!lrtimer && recvAck()){
        // Update the rt entry
        rt_update(rt,rp_dst_seqno,this->rp_hop_count,nb->
        jtnode,(CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT));
    }else{
        drop(p, DROP_RTR_MAC_CALLBACK);
        // Do the same thing for other packets in the
        // interface queue using the broken link
        while((p = ifqueue->filter(broken_nbr))) {
             drop(p, DROP_RTR_MAC_CALLBACK);
        }
        nb_delete(broken_nbr);
    }
}
```

8. Default values of the initial the route repair

```
#define NOM_INTERVAL              1               //sec
#define MaxNomInterval            (1.25 * NOM_INTERVAL)
#define MinNomInterval            (0.75 * NOM_INTERVAL)
#define ALLOWED_NOM_LOSS          3               // packets
#define MAX_ALLOWED_ JOINTNODES 3
#define LOCAL_REPAIR_WAIT_TIME    0.30            //NS2 = 0.15s
#define NODE_TRAVERSAL_TIME       0.03            //NS2 AODV
```

APPENDIX B

Table B1: Performance comparison of AODV and Modified AODV for varying node density

Node Density	NRO		PDR		Throughput		E2E Delay	
	AODV	Modified AODV	AODV	Modified AODV	AODV	Modified AODV	AODV	Modified AODV
25	4.48	4.87	0.85	0.89	5.35	5.03	113.1	134.0
50	4.60	5.20	0.87	0.90	9.77	8.39	177.5	179.1
75	4.69	5.23	0.89	0.91	13.93	13.5	214.1	174.3
100	4.71	5.18	0.89	0.90	18.32	16.92	220.9	189.8
125	4.68	5.09	0.88	0.91	27.21	19.07	280.3	225.9
150	4.69	5.21	0.85	0.90	35.97	24.74	315.8	240.3
175	4.66	5.03	0.83	0.88	55.68	28.42	334.4	266.5
200	4.52	4.85	0.78	0.77	57.25	36.73	340.1	322.2

Table B2: Performance comparison of AODV and Modified AODV for varying maximum velocity

Max Velocity	NRO		PDR		Throughput		E2E Delay	
	AODV	Modified AODV	AODV	Modified AODV	AODV	Modified AODV	AODV	Modified AODV
1	5.21	5.27	0.99	0.99	1.24	2.12	89.1	89.7
5	4.65	5.10	0.88	0.90	13.43	8.19	169.4	160.3
10	4.55	5.02	0.86	0.90	15.99	11.01	162.8	169.4
15	4.48	5.07	0.85	0.89	17.52	13.28	213.5	176.1
20	4.37	4.96	0.83	0.88	17.20	13.49	197.0	178.2
25	4.16	5.03	0.79	0.88	18.12	14.13	211.2	189.6
30	3.93	4.98	0.74	0.89	25.95	15.60	269.7	205.9
35	4.01	4.71	0.76	0.87	28.41	15.99	321.1	226.2