UNIVERSITY OF THE WESTERN CAPE

Situation-aware routing for wireless mesh networks with mobile nodes

by

Hlabishi Isaac Kobo

A thesis submitted in fulfilment of the
degree of Master of Science in the
Faculty of Science
Department of Computer Science

Supervisor: Dr WD Tucker

March 2012

Keywords

C.2.1 [**Network Architecture and Design**]: Network Communications/Network Topology/Wireless Communication; C.2.2 [**Network Protocols**]: Routing Protocols/Wireless Mesh Network; C.2.3 [**Network Operations**]: Network monitoring/Public networks; D.2.8 [**Metrics**]: Performance Measures

Abstract

This thesis demonstrates that a situation-aware algorithm improves quality of service on small mesh networks running BATMAN-adv with some mobile nodes. BATMAN-adv is a proactive mesh routing protocol that counts beacons as a link quality metric. BATMAN-adv was modified to give more recently received beacons more weight, thereby calculating a more precise indication of the current state of a link that BATMAN-adv can use to forward packets. BATMAN-adv 'original' was compared with a situation-aware version in two laboratory test beds with the same voice traffic profile on actual hardware with a realistic voice traffic profile; with controlled transmission rates and buffer sizes to simulate congestion. The second test bed included mesh potatoes, PCs and laptops as mobile nodes. BATMAN-adv achieved better jitter and packet loss than the situation-aware version in the initial, smaller test bed, and average throughput for both versions was almost identical. However, in the second slightly larger test bed, with additional mobile nodes, the situation-aware algorithm performed better than the original BATMAN-adv algorithm for all quality of service metrics, including throughput. Thus the thesis concludes that a situation-aware protocol offers a promising solution to address issues pertaining to mobility, congestion and scalability for voice traffic in mesh networks with mobile nodes.

Declaration of Authorship

I declare that *Situation-aware routing for wireless mesh networks with mobile nodes* is my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

Hlabishi Isaac Kobo

March 2012

Signed:

Acknowledgements

Firstly I would like to thank God Almighty for protection and a healthy life. This has been an emotional and intellectual journey that would not have been possible without the assistance and support of a number of important people and structures. I give special thanks to the Telkom/Cisco/THRIP Centre of Excellence (CoE) that provided the necessary equipment especially Telkom for financially sponsoring my studies

This thesis would not have being possible without the BATMAN development team for introducing their insight in the BATMAN code. Equally, I will be forever grateful to the people from Village Telco group for always being there when we needed help consequently becoming a special resource for my research.

There are special mentors that I must acknowledge due to their importance in my work who urged me on by way of untiring support and seemingly unlimited belief in me. Those are Dr William Dr Tucker, who besides his supervisory role, provided me with valuable parental guidance. My special gratitude is due Michael J. Norman and Prof. I. Venter who provided me with additional support and advice throughout the course of this study. Equally, special thanks go to Mr D. Leenderts, Ms F. Jacobs and Ms R. Abbott for the technical and administrative support. How could I forget Ms Z. Ramaila for the role she played providing emotional support since my first year and finally editing my thesis? I would not have been here without her help.

There are many that from behind the scenes have encouraged and supported my work, and I wish to thank them. Those are my colleagues in the Bridging Application and Networks Group (BANG), R. Hoorn, M. Irabaa, Z. Chitedze and D. Zulu. Thanks to Xiaoming 'Andy' Liu for brainstorming the ideas and Z. Roro for helping with the experimental set-ups. A big thank you goes to Pastor Pascal for his guidance and words of wisdom, may the good Lord bless him.

I am dedicating this thesis to my parents, Mr and Mrs Kobo, for their undying love and support throughout my studies.

May God bless you all, THANK YOU.

Contents

List of Figures

List of Tables

GloMoSim	**G**lobal **M**obile information system **Sim**ulator
GPS	**G**lobal **P**ositioning **S**ystem
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing
GSR	**G**lobal **S**tate **R**outing
GZRP	**G**enetic algorithms **ZRP**
HARP	**H**ybrid **A**d Hoc **R**outing **P**rotocol
HP	**H**ewlett-**P**ackard
HSR	**H**ierachical **S**tate **R**outing
IARP	**I**ntrazone **R**outing **P**rotocol
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
ICT	**I**nformation and **C**ommunication **T**echnology
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IEEE 802.11	Set of WLAN standards
IERP	**I**nterzone **R**outing **P**rotocol
IP	**I**nternet **P**rotocol
IPERF	**I**ntelligent **PERF**ormance prediction
IRU	**I**nterference-aware **R**esource **U**sage
ISP	**I**nternet **S**ervice **P**rovider
LAN	**L**ocal **A**rea **N**etwork
LMR	**L**ightweight **M**obile **R**outing
LRR	**L**ink **S**tate **R**outing
LS	**L**ink **S**tate
LSP	**L**ink-**S**tate **P**ackets
MAC	**M**edia **A**ccess **C**ontrol
MANET	**M**obile **A**d hoc **NET**work
MIC	**M**etric of **I**nterference and **C**hannel-switching
MP	**M**esh **P**otato
MPR	**M**ultiPoint **R**elays
MRL	**M**essage **R**etransmission **L**ist
NDP	**N**eighbour **D**iscovery **P**rotocol
NGO	**N**on-**G**overmental **O**rganization
NS2	**N**etwork **S**imulator **2**
OGM	**O**riginator **M**essage

OLSR	**O**ptimized **L**ink **S**tate **R**outing
OPNET	**O**ptimized **N**etwork **E**valuation **T**ool
OSI	**O**pen **S**ystems **I**nterconnection model
PC	**P**ersonal **C**omputer
QoS	**Q**uality **o**f **S**ervice
QRY	**Q**uery
RAM	**R**andom **A**ccess **M**emory
RPY	**R**eply
RQ	**R**equest
RTT	**R**ound **T**rip **T**ime
SATNAC	**S**outhern **A**frica **T**elecommunication **N**etworks and **A**pplications **C**onference
SEAD	**S**ecure **E**fficient vector routing for mobile **AD** hoc networks protocol
SECN	**S**mall **E**nterprise **C**ampus **N**etwork
SETT	**S**um of all the **ETT**s
SHARP	**S**harp **H**ybrid **A**daptive **R**outing **P**rotocol
SMA	**S**imple **M**oving **A**verages
SrcRR	A high throughput routing protocol for 802.11 mesh networks
SRTT	**S**moothed **R**ound **T**rip **T**ime
SSR	**S**ignal **S**tability **R**outing
TCP	**T**ransmission **C**ontrol **P**rotocol
TORA	**T**emporary **O**rdered **R**outing **A**lgorithm
TTL	**T**ime **T**o **L**ive
TQ	**T**ransmit **Q**uality
UDP	**U**ser **D**atagram **P**rotocol
UWC	**U**niversity of the **W**estern **C**ape
WCETT	**W**eighted **C**umulative **E**xpected **T**ransmission **T**ime
Wi-Fi	**W**ireless **F**idelity
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WMA	**W**eighted **M**oving **A**verage
WMN	**W**ireless **M**esh **N**etwork
WRP	**W**ireless **R**outing **P**rotocol
ZHLS	**Z**one-based **H**ierarchical **L**ink **S**tate protocol
ZRP	**Z**one **R**outing **P**rotocol

Chapter 1

Introduction

This thesis demonstrates that a situation-aware algorithm based on the current state of dynamic mesh networks improves the quality of service on dynamic networks with mobile nodes running BATMAN-adv. The thesis presents a situation-aware routing metric calculation that prioritizes the most recent link quality data to inform routing decisions on static wireless mesh networks (WMNs) with mobile nodes. This type of network is referred to as a dynamic mesh network. We believe these types of mesh networks will become prevalent as mesh network protocols improve and mobile devices become more powerful and able to run such protocols. As the network grows or shrinks, mobile nodes move around and congestion takes place, and the link quality is affected. Thus the routing protocol should be able to adapt and react based on such changes. The goal is to improve the effective usage of link quality metrics using situation-aware routing in BATMAN-adv in order to optimize quality of service (QoS) and throughput on such network.

1.1 Background

The former United Nations secretary-general Kofi Annan observed, *"Wireless technologies have a key role to play everywhere, but specially in developing countries and countries with economies in transition. With considerable speed and without enormous investments, Wi-Fi can facilitate access to knowledge and information, for example by making use of unlicensed radio spectrum to deliver cheap and fast Internet access. Indeed, it is precisely in places where no infrastructure exists that Wi-Fi can be particularly effective, helping countries to leapfrog generations of telecommunications technology and infrastructure and empower their people."* [34].

WMNs gained much popularity as well as research attention recently due to their inexpensive deployments. A WMN is a distributed network which can self-discover and self-heal [1]. There is published evidence that WMN provides ideal infrastructure to provide affordable wireless Internet access to the less privileged in developing countries. Information access remains a major concern in our society especially in rural areas. Even though people in such areas still rely on newspapers, television news and radios for information, issues such as unaffordability, illiteracy and unemployment become partial impediments for accessibility. Another additional challenge is that these media sources seem to not always deliver news content preferred by a considerable percentage of the intended end-users. Information access, however, should be made possible to everyone on demand. Access to information in rural areas is often linked to means and mode. Wireless networks together with wireless technologies seem to provide a better alternative for the developmental acceleration in the developing world particularly rural areas.

A WMN is packet-switched and has a static wireless backbone [65]. WMNs tend to have little or no mobility at all. Mobile ad-hoc network (MANET) is a form of WMN which is very dynamic. This research study introduces a hybrid between WMN and MANET by adding mobile nodes to the WMN. This type of network is herein referred to as dynamic WMN with mobile nodes. Mobile nodes enhance the expansion and scalability of the network. However, since mobile phones have great mobility and instability, the routing protocol needs to be optimized to adapt to changes in the network. Thus, we implement a situation-aware algorithm in the Better approach to mobile ad-hoc network advanced (BATMAN-adv) protocol.

BATMAN is a proactive mesh network routing protocol (explained in detail in Section 2.2.1). BATMAN's control messages, called originator messages (OGMs), are relatively small packets of 52 bytes. BATMAN's nodes do not maintain the routing information of the entire network [1]. Rather, each node only maintains information about the best next-hop towards a destination [1, 53]. This reduces the signal overhead and avoids unnecessary knowledge about the whole network. The objective of this protocol is to enhance the probability of delivering a packet. The protocol maintains information about the existence of a node and does not check the quality of a link [30]. All BATMAN nodes periodically send/broadcast OGMs. Each OGM contains the original sender's address, address of the node rebroadcasting the OGM, the Time To Live (TTL) and a sequence number. The sequence number is incremented for each OGM, i.e. the first OGM gets 1 and so on. Thus, BATMAN also keeps track of the 'freshness' of an OGM. Any sequence number received with a value lower than the previous one gets dropped [53]. The TTL is used to limit the number of hops on which the packets must pass through before it expires (gets dropped). Upon receiving the OGM, each node then rebroadcasts it to its neighbours. However, each node only rebroadcasts OGMs coming through the current best next-hop. The number and the reliability of the OGMs determine the route discovery as well as neighbour selection.

BATMAN has two protocol implementations, BATMANd and BATMAN-adv. BATMANd is an earlier version and runs on OSI/TCP layer 3 (Network layer). This version can also run on Android mobile phones referred to as BATphone. This version suffers from asymmetric links. Its biggest advantage is that it can scale up to more than 200 nodes [63]. BATMAN-adv on the other hand is an improvement from BATMANd and is based on the OSI/TCP layer two (MAC layer) [63]. This version however, can only scale up to 20 nodes and is therefore only suitable for Small Enterprise Campus Networks (SECN) and is often referred to as the SECN version [67].The major drawback of BATMAN-adv is that it cannot run on mobile phones (yet) because it is difficult to support on multiple mobile phone kernels.

1.2 Motivation

We chose BATMAN because it has in general a high stability level and high packet delivery ratio [1]. The BATMAN routing protocol metric is simple to compute making it fast to process.The basic BATMAN is also the primary routing protocol for mesh potato devices (see Section 2.2.2).

Our research group, Bridging Applications and Networks Group (BANG) has been working together with two non-governmental organizations (NGO), Transcape (`http://www.transcape.org`) and Federation of Rural Coastal Communities (FRCC) (`http://orgs.tigweb.org/federation-of-rural-coastal-communities`), both based in the rural Eastern Cape of South Africa. The main purpose of this cooperation is to bridge the digital divide in these rural areas. The concerned villages surround the Tshani and Mankosi communities and includes: Canzibe, Lwandile, Mamolweni and Hluleka. Each NGO is concerned with its immediate community villages, thus we work with Transcape for Tshani, Mankosi and Canzibe and FRCC for Lwandile, Mamolweni and Hluleka. Figure 1.1 represents the village clusters connected in between by long-range 5 GHz links. There is a hospital situated at Canzibe and a clinic situated in Lwandile. These two are also connected by long range 2.4 GHz links.

We chose to specifically modify BATMAN-adv because it is an improved version of BATMAN and also suits the geographical structure of the two sides in particular. As shown in Figure 1.1, a few rondavels can be grouped together forming a cluster which are close to each other. The idea is to use SECN within a cluster to enhance the performance and quality of service. For example, the headquarters for Transcape is based at Tshani

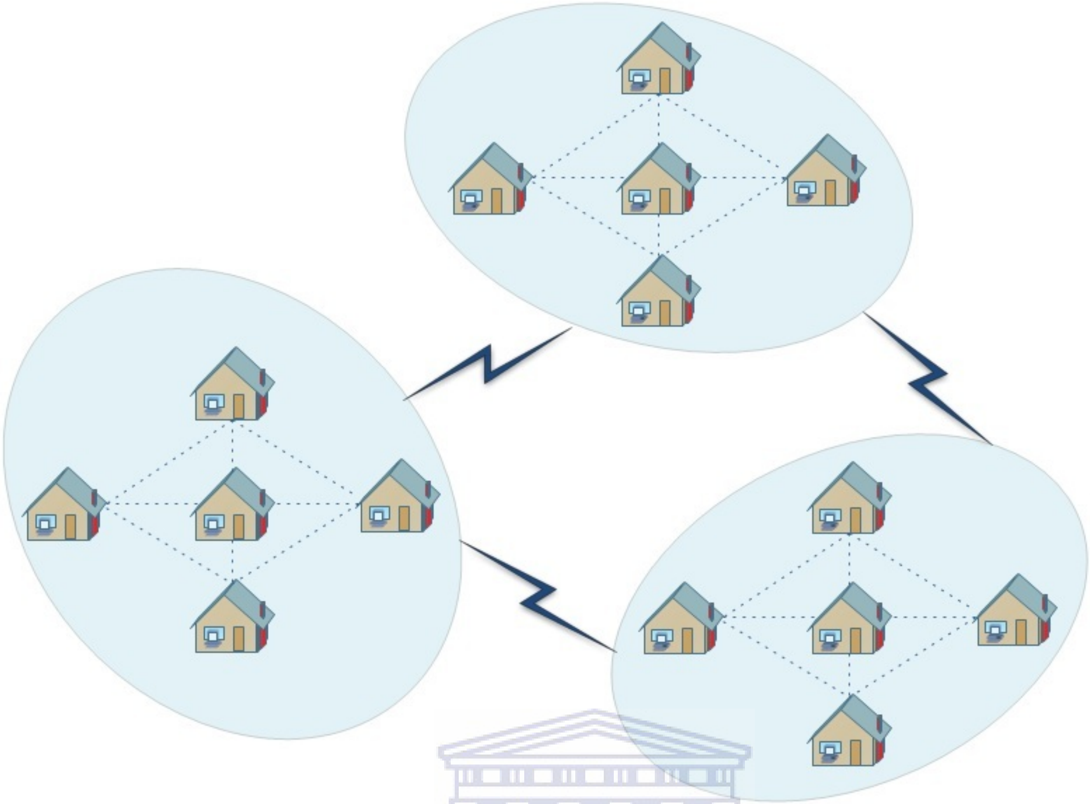


Figure 1.1: The villages near Mthatha in Eastern Cape, South Africa
A dynamic mesh network, applicable to a rural NGO headquarters with a co-located backpackers, with static mesh routers inside rondavels where NGO workers live and work, and mobile mesh nodes on their laptops. Note that link quality in such networks changes as NGO workers move around to work with one another, and can improve using the situation-aware algorithm described in this thesis.

and NGO workers as well as the backpackers that stay there have laptops. The idea is to leverage the laptops to extend and strengthen the SECN network. The clusters are then connected together using supernodes which support long distance point-to-point.

1.3 Research question and approach

The main objective of the research is to evaluate the effectiveness of the situation-aware technique on BATMAN; based on congestion, scalability and mobility. Our research question is formulated as : **Can situation-aware routing improve the BATMAN protocol to realize better quality of service in a mesh network with some mobile nodes?** To answer the research question , we designed a situation-aware routing algorithm that factors in the current network situation. We evaluated the algorithm on two network test beds: the small scale of four nodes and larger scale of twelve nodes in order to measure the impact of scalability. The small scale test bed consisted of three personal computers (PC) and one laptop while the larger scale consisted of two PCs, two laptops and eight mesh potato devices. The algorithm was tested upon jitter, packet loss and throughput; latency increased in the larger scale test bed. We used IPERF to generate the packets and monitor performance. In achieving mobility, the laptop devices were moved around during the transmission of packets. We regulated the packet transmission rate and transmission buffer size to achieve congestion levels.

1.4 Thesis outline

Chapter 2 presents related work. It first looks at the challenges of dynamic mesh topologies. It then discusses related wireless mesh routing protocols focussing on three classifications, reactive, proactive and reactive protocols. Link quality metrics and moving averages complete the chapter.

Chapter 3 presents research methods, experimental design and implementation. Limitations of the previous work are identified to form the objectives of this study. A research question is identified as well as methods to answer it. Chapter 3 also presents the design and implementation of the situation-aware algorithm for BATMAN.

Chapter 4 presents the results obtained from the two experimental test beds. We compare the findings 'between' the two test beds and 'within' the two protocols.

Chapter 5 concludes the thesis and discusses limitations of the study, recommendations and future work. It first discusses the conclusive arguments from the results then identifies the limitations of the experimental design. It also outlines recommendations based on what we learned from this study and suggests ideas for work that we could not do because of time constraints as future work.

The **Appendices** present both published and unpublished papers on this study. The project proposal was summarised by a work-in-progress short paper for Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2010 in a poster (see Appendix A). Initial results were presented in a full SATNAC paper in 2011 published in the proceedings (see Appendix B). The final results are neatly summarized in a draft paper (see Appendix C). It should also be noted that the papers have co-authors that collaborated in writing the papers, but the work presented in this thesis is solely the work of the thesis author.

Chapter 2

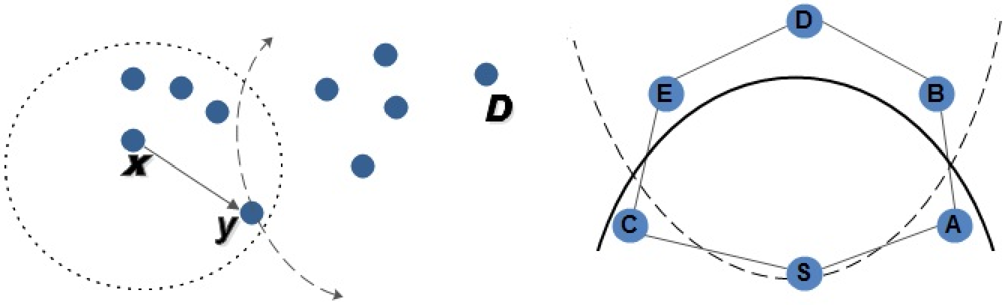
Literature review

This chapter outlines an analysis of work related to dynamic mesh routing protocols and their metrics. The challenges of these dynamic networks are discussed together with different approaches to deal with them. Section 2.1 gives a brief description of dynamic topologies and their challenges, and offers two examples of algorithmic solutions. Section 2.2 reviews wireless mesh routing protocols: reactive protocols, proactive protocols and hybrid protocols. Section 2.3 looks at link quality metrics and Section 2.4 is about different weighting average solutions and their role in routing.

2.1 Dynamic topologies and challenges

Dynamic wireless mesh network topologies are characterised by an ongoing movement of nodal positions. Rapid changes in such networks incur instabilities such as link failure when links move in and out of range. Routing in these kinds of topologies becomes a very big challenge as a result [61]. The flexibility of network size due to the ad-hoc addition of mobile nodes poses another challenge of scalability, especially with increasing mobility [33].

Amid these challenges encountered in dynamic topologies, Tsirigos and Haas proposed multipath routing to address them [61]. This method splits the routing information into small portions forwarded alongside all available paths towards a destination using diversity coding. Data load is distributed over multiple paths to minimize packet drop rate, achieve load balancing and improve end-to-end delay [61]. Another method proposed by Karp and Kung called Greedy Perimeter Stateless Routing for wireless networks (GPSR) utilises the nodes' positions towards destination nodes to deal with routing challenges in dynamic topologies [33]. This method takes advantage of the node's geographical locations to achieve scalability and mobility control by the use of greedy forwarding and perimeter forwarding. In greedy forwarding, packets are routed to a node geographically closest to the destination node. Greedy forwarding is illustrated in Figure 2.1(a). In cases where greedy forwarding fails like in Figure 2.1(b), GPSR uses perimeter forwarding [33].



(a) Greedy forwarding (b) Perimeter forwarding

Figure 2.1: Greedy versus Perimeter forwarding [33].
The two algorithms used in GPSR protocol to address challenges encountered in dynamic topologies. In 2.1(a), node x forwards any packet destined to D to x as it is the closest as compared to all other neighbours in the wireless range of x denoted by the dotted circle. In 2.1(b), source node S cannot use greedy forwarding to A or C to utilize paths *(A-B-D)* and *(C-E-D)* respectively because S is a local maximum in its proximity to D. For this reason the right-hand rule would be applied and route *(A-B-D)* chosen as a result.

2.2 Wireless mesh routing protocols

Routing is a process of delivering data packets from a source (sender) node to destination (receiver) node on the network. Routing protocols on the other hand deal with the maintenance, creation, establishment and discovery of such routes [53]. Routing protocols are classified into three protocol categories: reactive, which is on demand, proactive, which is table driven, and hybrid, in which the former two are merged. Section 2.2.1 looks at reactive protocols. Section 2.2.2 and 2.2.3 looks at proactive and reactive protocols respectively.

2.2.1 Reactive protocols

The reactive protocols, also referred to as on-demand protocols, create a route from source to destination only when needed, i.e. when there is data to be sent. This approach uses network flooding to find the routes [53, 68]. Reactive protocols are suited for mobile ad-hoc networks where there are frequent topology changes due to the mobility of routers [1, 53]. According to Abolhasan *et al.* flood-based route discovery provides high network connectivity and low message overheads [1]. The method does not waste bandwidth by propagating control packets where they are unnecessary [1]. This scheme leads to higher latency on the network because of route discovery. Pinto argues that reactive protocols are suitable for a less dense network with static traffic patterns whilst proactive for dense networks with bursty traffic patterns [53]. The on-demand route search is proportional to the size and the network traffic type of the network and hence the delay is less in small networks with consistent traffic patterns.

Ad-hoc On-demand Distance Vector (AODV) is one of the popular reactive protocols and thus creates routes on demand. AODV uses single path routing and is based on a hop-by-hop routing metric [38]. Single path routing is whereby a node can only have one path towards a destination [38]. AODV's routing table only stores information about the best next-hop towards a destination [53]. Sequence numbers are used to ensure loop-free routes as well as the freshness of the routing information [38, 53, 56]. The AODV protocol uses unicast, broadcast and multi-tasks for communication on the network. Unlike other reactive protocols which use flooding for route discovery; AODV use broadcast to flood route requests to the intermediate node and the destination nodes replies with a unicast route reply [8]. There are multicast groups where a multi-task of sequence number takes place [38]. From AODV, an Ad-hoc On-demand Multipath Distance Vector (AOMDV) routing protocol was born. Unlike AODV, AOMDV is based on multipath routing. This protocol was developed to alleviate link failures and link breaking suffered in AODV [38, 56]. This model creates a backup plan for unforeseen circumstances in a highly dynamic network.

The use of periodic beacons/control messages is very popular in reactive protocols, however others are source based and hence do not make use of this method e.g. DSR and srcRR. The Dynamic Source Routing (DSR) as defined by Johnson *et al.*, is a simple and efficient routing protocol designed specifically for multihop wireless ad-hoc networks for mobile nodes [31]. The DSR protocol needs no administrator or any existing network infrastructure as it can self-configure and self-organise [18, 31, 32, 35]. DSR has two fundamental mechanisms in Route Discovery and Route Maintenance. These two components allow the nodes to self-discover and self-maintain routes to dynamic destinations in the ad-hoc network [35]. Route discovery mechanism enables a node which wishes to send a packet to a particular destination to acquire a source route to that destination that is used only when a route to the destination is unknown. This is done through broadcast transmission of ROUTE REQUEST messages towards all nodes in the vicinity of the initiator node. Each ROUTE REQUEST message has the initiator and a target for identification purpose. Figure 2.2 shows an example of a route discovery process whereby the initiator node A broadcasts route request packets to its neighbours and the propagation goes on until a route is established.

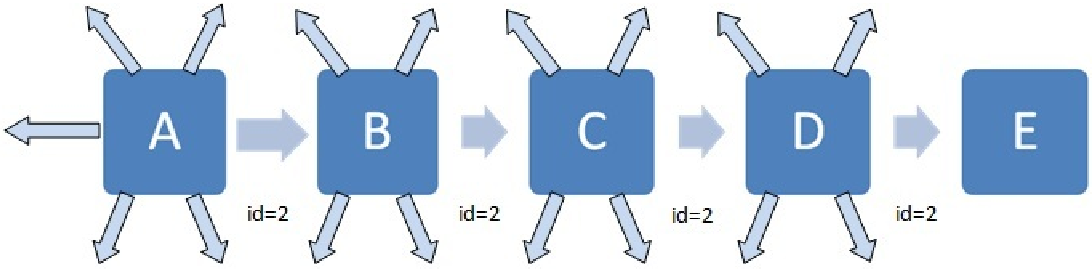


Figure 2.2: Route discovery in DSR.
Source node A searches for a route to destination node E [31], all intermediate nodes broadcast the request further to their respective neighbours. The *id* is a unique identifier of the request and hence does not change.

The recently discovered routes are stored in the node's cache so that the node can first refer to it before invoking the route discovery mechanism when it wishes to send a packet [32]. Each packet sent contains the source address and the destination address. Route maintenance mechanism enables each node to detect a source route leading to a destination that no longer works due to topology change or any external factor [31]. DSR need support from the (MAC) layer to identify link failures [18]. Both route discovery and route maintenance operate on demand [31]. According to Khatri *et al.*, DSR conserves power as the nodes can enter a sleep mode when they are idle and also save a lot of bandwidth [35]. Johnson *et al.* maintain that DSR scales down the number of routing overheads caused by the protocol to zero [31]. However the main weakness of DSR protocol is the scalability issue thus suitable for small networks [49].

SrcRR is another reactive protocol based on DSR with source initiated data traffic [4]; it inherits the operation of link caches from DSR. The major difference between srcRR and DSR is that srcRR uses the ETX metric for its routing decisions [4] (refer to Section 2.3 for full description of ETX). srcRR operates at a lower layer and operates independently of IP [4]. The srcRR use IP headers and uses 32-bit addressing scheme [4]. The nodes maintain a mapping from srcRR 32-bit address to 48-bit 802.11 MAC addresses.

Associativity Based Routing (ABR) is a source-initiated reactive protocol which was developed in 1996 at Cambridge University by C.K Toh [60]. It is a bandwidth efficient distributed routing protocol used in ad-hoc networks [59]. ABR uses both point-to-point and broadcast routing [59]. ABR uses periodic beacons to let the neighbours know about its existence [59], thus it uses a similar approach with AODV. ABR's routing decisions are based on the link stability [12] referred to by many as a property of associability. The periodic beacons contain associativity ticks where all the alternating nodes between source and destination ticks upon receipt. The neighbours count the number of beacon ticks received and evaluate the stability of the connection between themselves and their sources [12]. Thus the destination node selects the most stable route [12]. If there is more than one route with the same weight of associativity stability, the shortest path (minimum hops) gets selected [59] while the other routes get discarded.

While ABR uses link-stability-based routing, another beacon-based protocol, Signal Stability Routing(SSR) uses signal strength and node location stability for routing decisions. The link stability is directly proportional to signal strength. The protocol evaluates the signal strength of the beacons. The signal strength is then stored in a signal stability table [47]. SSR differs from ABR in the sense that only route requests received from a stable link get forwarded by the receiving node.

Other on-demand routing protocols are based on link reversal routing (LRR). Link reversal routing were developed to suit ad-hoc networks due to their ease adaptability and scalability with more emphasis on fast changing topology networks [62]. These protocols reduce overheads on the network through their on demand architecture and by confining all the topological changes to affect local nodes only [47, 62]. The link reversal algorithm is based on the graph theory principle called a Directed Acyclic Graph (DAG). In routing terminology DAGs are destination oriented [47, 62]. DAG is a directed graph with no directed cycles, thus a graph with directed arcs [47]. A graph is acyclic when it has no loops. There are three LRR-based routing algorithms in existence; Gafni-Bertsekas (GB) algorithm, Lightweight mobile routing (LMR) and Temporary ordered routing algorithm (TORA). GB is the oldest LRR algorithm which dates back to 1981 [47], it was designed for packet-radio networks. The idea is to keep a directed route from all the nodes to the destination node. Thus no node which only consists of incoming links exists in the DAG except the destination itself and the checking is done proactively [62]. GB uses two fundamental methods to handle nodes without outgoing links: full reversal and partial reversal.

The full reversal method reverses the direction of all the links [62],thus leaving the node with only outgoing links. However the partial reversal method, not all the directions of the links are reversed. Only the edges which have not been reversed recently get reversed. Each node keeps track of all the links that have been reversed in the last iteration and only reverses those that have not been reversed [47, 62]. GB algorithms are deadlock and loop free and can maintain multiple routes through the use of the DAG. According to Park this algorithm becomes unstable in the portions of the network which become disconnected from the destination [49]. This leads to nodes transmitting control and message packets until the network is reconnected hence not converging which results in lot of bandwidth consumption [49].

Due to the instability endured in GB algorithms, the Lightweight Mobile Routing (LMR) algorithm was created to address issues of non-convergence for partitioned networks found in GB [47, 62]. Unlike GB LMR is on demand. According to Vainio [62], LMR is loop and deadlock free like GB however more stable. Like many other reactive protocols, LMR has two fundamental functionalities of Route Establishment and Route Maintenance [47]. LMR consists of three types of packets which are used predominantly, Query (QRY), Reply (RPY) and Failure Query (FQ). During route establishment, a node issues a QRY packet i.e. when a route is needed. The packet is flooded through the network over the undirected links. If any of the neighbours knows the route to the destination, a (RPY) packet is send back to the source node and thus a route from source to destination established. Route maintenance is invoked when a route to a destination is needed [47], only when the last route to the destination ceases to exist. Network traffic at the routes gets monitored and after a certain period of network inactivity, the route is considered inactive. If the recently lost link was the last route to the destination, the concerned (source) node sends a failure query (FQ) packet. This informs the destination so that other nodes which used that route before can stop doing so. LMR also suffers like other LRR algorithms in case of partitioned networks. Through false reply propagation, temporary invalid routes are created [49].

The LRR algorithm, Temporary-Ordered Routing Algorithm (TORA) is based on both the GB algorithm and the LMR algorithm [49]. TORA is an adaptive, scalable and efficient distributed routing protocol/algorithm for mobile ad-hoc networks [18, 32]. It is an on-demand source-initiated routing protocol [32]. The main feature in TORA is the localization of control messages around the point of topological change [18, 32]. TORA also consist of three basic functionalities in Route Creation, Route Maintenance and Route Erasure. These functions uses the three control packets respectively, Query (QRY), Update (UPD) and Clear (CLR). TORA uses a height parameter to determine link direction between nodes. The biggest disadvantage of TORA is that like the LMR protocol, temporary invalid routes are inevitable.

2.2.2 Proactive protocols

In proactive protocols, also known as table driven, each node in the network maintains a table containing routing information of the entire network. Each node then periodically broadcasts control packets (hello packets) to the whole network to let them know about its existence. The routing tables are periodically updated to maintain the adequacy of the routing information and thus keeping the network up to date in-line with the topological changes. The biggest advantage of this scheme is the minimization of route discovery delay and consequently lower latency in delivering a packet. However because of the periodic updates of control messages that get propagated through the whole network, the overheads increases. Thus bandwidth consumption also rises. This protocol scheme encompasses the family of link state, distance vector as well as cluster-based algorithms. Cluster-based protocols can be link state or distance vector.

Link-state protocols are based on the principle of a distributed map [22], Thus every node establishes a network connectivity map in the form of a graph. The routing information on the map is stored in a distributed database i.e. a routing table. These protocol types consider shortest path first and use Dijkstra's algorithm. The Optimized Link State Routing (OLSR) protocol is a proactive protocol which is based on the link state algorithm. OLSR's objective is to reduce the size of the control packets as well as the overhead cost by broadcasting control packets [26, 42, 53]. This protocol is the optimization of the link state protocol for mobile ad-hoc network and uses hop-by-hop routing [26, 42]. Multipoint Relays (MPR) are the key concepts in OLSR; MPRs are the subsets of the neighbours of which a node uses to forward broadcast messages. MPRs reduces duplicate retransmission in the same region and thus minimize flooding overheads [26, 53]. Each node selects its MPR on the network. The set of MPRs is selected such that every node can reach its 2-hop neighbour. Each node in the network periodically broadcasts information about its neighbour that has selected it as an MPR. Each node calculates its routing information and updates its table after receiving MPR selectors list. The routing path between source and destination is selected based on the sequence of hops in between [42].

Many Link-State (LS) routing protocols suffer from the effects of message flooding. To alleviate this shortcoming Global State Routing (GSR) which is a MAC efficient protocol like DSDV (see below for DSDV) was introduced. The main goal of this protocol is to address the shortcomings endured in many LS (link-state) protocols such as flooding of routing messages. This controls the size and the number of the control packets in order to achieve optimized MAC throughput (GSR). GSR algorithm keeps the core of the LS schemes without the flooding mechanism. It adopts the information dissemination used in Distributed Bellman-Ford (DBF) which does not use flooding [14]. LS packets are flooded into the network only when there is a topology change. However in GSR nodes only maintain a link state table based on the up-to-date information received from neighbouring nodes which is exchanged further with its local neighbours periodically [14]. Information dissemination is invoked when a link state with a high sequence number replaces the one with a small sequence number [14], hence it is similar to DSDV in this regard. The large size of the message update often lets GSR throughput degenerates as it consumes lot of bandwidth.

To overcome this deficiency Fisheye State routing was created (FSR). FSR is also a LS routing protocol inspired by the fish-eye technique by Kleinrock and Stevens [51]. Figure 2.3 shows the scope of FSR for the centered node, the scope is defined by the number of hops needed to reach a particular node.

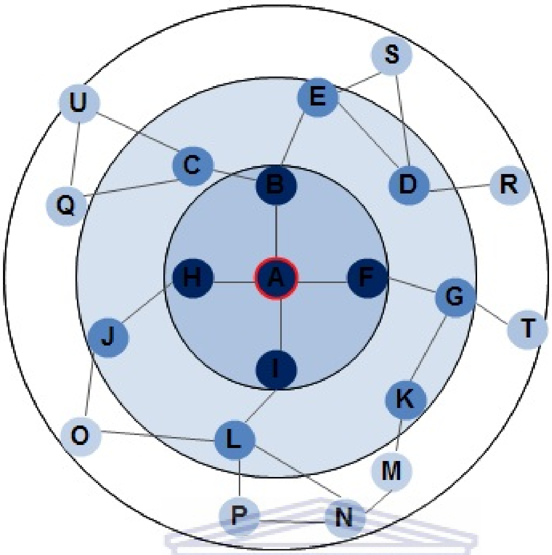


Figure 2.3: Fisheye state routing.
Scope of fisheye state routing (FSR) shown by the centered node encircled in red [51], the cycle shows one, two, and three hop neighbours respectively.

According to Pei *et al.* this technique is used to reduce the size of information required for graphical data representation [51]. Thus the eye of a fish captures high detailed pixels near a focal point and as the distance from the focal point increases the pixel details decrease [51]. This translates to maintaining accurate rate-distance and path-quality information about the immediate neighbour of a node with progressively less detail as the distance increase [51]. The message sizes are thus reduced as information gets exchanged more frequently with closer nodes than further nodes and hence such update does not contain information about all nodes [41]. FSR scales well in large networks where there is high mobility [51].

Hierarchical State Routing (HSR) is another link state proactive routing whose topology is organised hierarchically. Its characteristics are its multilevel partitioning (logical and physical) and clustering of its mobile nodes. The network is partitioned into clusters which have cluster heads [41, 50]. The cluster heads further form clusters and so on. The goal of this clustering is to utilize radio channel resources efficiently and to reduce the network layer routing overhead [51]. The clustering is both physical and logical. The physical clustering is based on the geographical relationship between the nodes while the logical clustering is based on logical and functional kinship between nodes [50]. Logical partitions play a key in mobility management of the network which keeps track of the mobile nodes while keeping control of message overheads at low [50].

The link-state protocols inform the entire network for any topological change encountered converse to distance vector protocols which only informs its neighbours about any change. Distance vector protocols often use Bellman-Ford and Ford-Fulkerson algorithms. Perkins and Bhagwat applied the classic Bellman-Ford algorithm to a much popularised Destination-Sequenced Distance Vector (DSDV) which is both a distance vector and a proactive protocol [15, 52]. The algorithm has been slightly modified to deal with the poor looping properties and the time dependent nature of the interconnection topology describing links between mobile hosts. Each mobile host maintains a routing table that enlists all available destination, the next node to reach the destination, the number of hops to those destinations as well as the sequence number assigned by the destination node. The motivational goals behind the creation of this protocol was to address the looping issues encountered in other distance vector routing protocols while on the other hand keeping the simplicity of the Bellman-Ford algorithm. The sequence number distinguishes stale routes from new ones and thus avoids the creation of routing loops which often occurs in many multipath protocols [15, 41, 52]. The nodes periodically send updates to its neighbours, which to its detriment increases routing overheads. Updates are either sent as full dump or incremental/triggered dump. Full dump sends the entire table which spans more packets while the incremental dump only sends those destinations that underwent a route change since the last full dump update [24]. Incremental updates are used in more stable network to avoid extra traffic [41].

The incremental/triggered updates are applied when an important routing change occurred. The receipt of a new metric (distance) for some destination or the receipt of a new sequence number causes the triggered update [24]. According to Hu *et al.* the latter tends to stands more and hence referred to DSDV-SQ (sequence number) [24]. When DSDV detects a broken link, a new routing update is created with an infinite metric and increases the sequence number. The advantages of using the Bellman-Ford algorithm is the simplicity and computation efficiency due to distributed characteristics [14]. However its low convergence and tendency of creating routing loops works to its detriment hence it is not suitable for highly mobile networks.

Based on DSDV-SQ version of DSDV, Secure Efficient Vector (SEAD) routing for mobile ad-hoc networks protocol was developed [24]. Unlike DSDV, SEAD does not use average weighted setting time to delay the sending of triggered updates [24]. The other notable difference is that it does not increment the sequence number for a destination whose link was detected as broken. Another DSDV based protocol, BABEL, is a proactive protocol using ETX metrics [1, 44]. BABEL uses Cisco's Enhanced interior Gateway Protocol (EIGRP) techniques to avoid loops [44]. This protocol consists of two characteristics to optimize relay performance: history-sensitive route selection to minimize route flaps and reactive update to force routing information request when a link failure to a neighbour occurs [1]. BABEL operates on both IPv4 and IPv6 addressing schemes.

The work done by Royer and Perkins was aimed at maintaining routing information on the network by using multiple tables in the formation of Wireless routing protocol (WRP) which is another proactive distance vector protocol [57]. Each node in the Network maintains four routing tables: distance table, routing table, link cost table and Message Retransmission List (MRL) table [15, 57]. The distance table contains the distances between source nodes and their destination nodes. The routing table contains information such as the best path towards a destination [41]. Link cost table contains the cost of the link to each neighbour and the number of time-outs based on error-free messages received from that neighbour [41]. The MRL contains the sequence number of the update message, retransmission counter and number of updates contained in the update message.

The HSR protocol discussed above is a cluster-based link state protocol while Cluster-head Gateway Switch Routing (CGSR) below is a cluster-based distance vector protocol which shows that the architectural design in clusters is applicable to both. CGSR is based on the DSDV routing algorithm; Nodes are grouped into clusters where cluster-heads are elected. The following Figure 2.4 illustrate routing in CGSR, node A, E and H are cluster-heads.

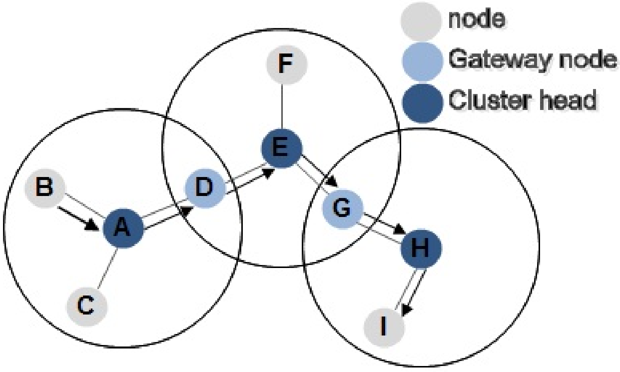


Figure 2.4: Clusterhead gateway switch routing.
An example of CGSR routing from node B to node I [57]. A packet from node B travels through A, its cluster head then to D, the gateway node until it reaches destination I in the same fashion.

All nodes in communication range of a cluster belong to that cluster. A node that is in connection with two or more clusters is called a gateway node, e.g. Node D and G in Figure 2.4. Due to frequent cluster heads changes in dynamic networks topologies, CGSR uses least cluster change algorithm to avoid performance degradation [41]. A change in a cluster-head occurs only if a topology change results in two cluster heads coming together for the formation of one cluster or one of the nodes moving out of the range of all the cluster heads [41, 47]. Two tables are used, cluster member table containing the cluster head of each destinations and DV-routing table containing the next best hop towards a destination [47]. A packet is submitted to the cluster head which further send it to the gateway node that connects to other cluster-head on route to the destination [41]. The destination cluster-head then send the packet to its prescribed destination.

A Better Approach To Mobile Ad-hoc Network (BATMAN) is a proactive mesh network routing protocol which unlike other proactive protocols defeats the drawbacks of bandwidth consumption. BATMAN's control messages, called Originator Messages (OGMs), are relatively small packets of about 52 bytes. BATMAN's nodes do not maintain the routing information of the entire network [1]. Rather, each node only maintains information about the best next-hop towards the destination [1, 45]. This reduces the signal overhead and avoids collecting unnecessary knowledge about the whole network. The objective of this protocol is to enhance the probability of delivering a packet [30].

All BATMAN nodes periodically send/broadcast control packets, or OGMs. Each OGM contains the original sender's address, address of the node rebroadcasting the OGM, TTL (time-to-live) and a sequence number. The sequence number is incremented for each OGM, i.e. the first OGM gets 1 and so on. Thus, BATMAN also keeps track of the freshness of an OGM. Any sequence number received with a value lower than the previous one gets dropped [45]. The TTL is used to limit the number of hops on which the packets must pass through before it expires (gets dropped). Upon receiving the OGM, each node then rebroadcasts it to its neighbours. However, each node only rebroadcasts OGMs coming through the current best next-hop. The number and the reliability of the OGMs determine the route discovery as well as neighbour selection.

Algorithm 1 The BATMAN Algorithm

```
Let $G = (N, E)$ be the network graph, where $N$ is a set of nodes and $E$ is a set of links between two nodes. Let $K$ be a set of one-hop neighbours for each node $i \in N$ and let $s$ be the source and $d$ be the destination.
Step 1: Consider routing message $m$ from $s$ to $d$ on network $G$. Eliminate all links $(s, i)\ \forall\ i \neq K$ to reduce the graph.
Step 2: Associate each link with weight $W_{si}$ where $W_{si}$ is the number of originator messages received from the destination through neighbour node $i$ within the current sliding window.
Step 3: Find the link with largest weight $W_{si}$ in the sub-graph and send $m$ along the link $(s, i)$.
Step 4: If $i \neq d$ repeat Steps 1 to 4 for routing message from $i$ to $d$ in the sub-graph $S$.
```

BATMAN has a bidirectional link-check parameter to address the problem of asymmetric links suffered in most wireless networks. BATMAN node uses the rebroadcast of its own OGMs as an asymmetric link check. It awaits a rebroadcast of its own OGM from the neighbours within a certain time-frame. However, it also experiences serious flaws in dealing with length of the time-frame. A short time frame makes BATMAN more strict in choosing links, thus leading to more ignored links which could be used in one direction. If the time-frame is big and BATMAN less strict, more links will be accepted and hence resulting in wrong routing decisions. BATMAN advanced, referred to as BATMAN-adv, is a Layer 2 protocol introduced to overcome this setback by using a Transmit Quality (TQ) algorithm. BATMAN-adv consists of two fundamental functions: receiving link quality (RQ) and transmit link quality. Receiving link quality deals with the probability of transmitting a packet successfully towards a node [67]. The transmitting link illustrates the probability of transmitting a packet successfully towards a neighbour [67]. TQ is the most important because RQ does not influence the routing decision. RQ is determined by the by the number of received OGMs. Echo link quality (EQ) is the number of the rebroadcasted OGMs from neighbours. TQ is calculated by dividing the EQ by the RQ i.e. [67]:

$$TQ = \frac{EQ}{RQ}. \tag{2.1}$$

Local link quality is propagated throughout the network to inform other nodes about the transmission quality. For this reason, BATMAN has an added field TQ which is 1 byte on the original OGM packet making it 53 bytes long. This field is set to maximum length of 255 for every new OGM. On receiving the packet a neighbour would calculate its local link quality into the received TQ value before rebroadcasting it. Thus every receiving node knows the transmit link quality towards the originator node. The local link quality is added to the TQ as follows [67]:

$$TQ = TQ_{incoming} \times TQ_{local}. \tag{2.2}$$

Again BATMAN-adv does not always rebroadcast a newly calculated TQ, TQ received through the best ranking neighbour is rebroadcast instead to support asymmetric link scenarios. In essence BATMAN-adv keeps track of two TQ values, local TQ as well as the global TQ. Local TQ describes the transmit quality towards every single neighbour calculated by counting the received OGMS plus TQ calculation. The global TQ on the other hand describes the link quality towards every multi hop neighbour calculated via BATMAN packets [67].

2.2.3 Hybrid protocols

Hybrid protocols exhibit the behavioural design of the two above mentioned protocols, reactive and proactive protocols. Hybrid protocols are very challenging because the switch from protocol to the other needs to be very sharp. However this is still a major concern [53] and thus hybrid protocols are still theoretical rather practical due to their complex implementation [1].

MeshDv is a hybrid protocol which uses the combination of proactive route computation for the routers and on-demand path request for clients [25, 53]. The proactive route is based on the destination-sequenced distance vector (DSDV) protocol [25]. This protocol runs on TCP/OSI layer 2 [25] and it uses IPv6 addressing scheme only [53].

Haas' work [19]; introduces us to one of the most integral algorithms in hybrid protocols, the zone based routing. Zone-Routing Protocol (ZRP), a zone-based hybrid protocol, proactively maintains routing information for the local neighbourhood hereby referred as the routing zone. It reactively acquires routes to destinations that are outside the routing zone [20]. The routing zone is a collection of nodes whose distance (in hops) is no greater than the zone source radius [20]. The zone radius is based on the number of hops not the geographical distance [36]. The nodes of a zone are divided into peripheral nodes and interior nodes [36]. Peripheral nodes are those nodes whose minimum distance is exactly equal to radius r [36] while interior nodes have a minimum distance of less than the radius. Figure 2.5 illustrates an example of the zone routing concept with radius of two nodes, the zone belongs to central node A and node E, L and I are peripheral.

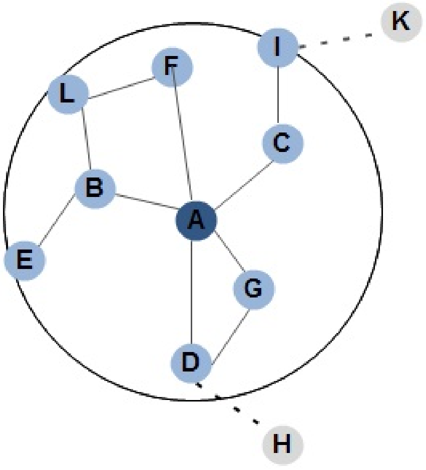


Figure 2.5: Zone routing protocol.
An example of ZRP with a radius of two hops [20]. Nodes E, L and I are peripheral nodes because they are exactly two hops away from the centre node while nodes H and K are beyond the routing zone of central node A.

A node needs to know its neighbours before a routing zone is constructed [20, 36]. A neighbour in this case is a node which is one hop away and in direct communication with the source node. The MAC provides the identification of each node participating in the network. In other instances Neighbour Discovery Protocol (NDP) is used to implement the route discovery process [20, 36] which typically use hello beacons. The neighbour discovery information is used as the basis for local proactive monitoring of routing zones through Intrazone Routing Protocol (IARP) [20, 36]. IARP is a family of link sate proactive routing protocols which maintain information of nodes only within the routing zone [36]. On the other hand, the Interzone Routing Protocol (IERP) acquires routes to destinations beyond the routing zones [20]. Thus IERP is a family of reactive protocols and hence concludes the hybrid architecture of ZRP. IERP does not use broadcasting for route requests as in other reactive protocols but rather uses a process of Boardercasting [20, 36]. Boardercasting uses a the routing information provided by IARP to direct a packet to the zone border (peripheral) [20, 36]. This packet delivery method is provided by Boardercasting Resolution Protocol BRP [36]. The components mentioned above form the core of the ZRP architecture, Figure 2.6 shows how they relate to each other.

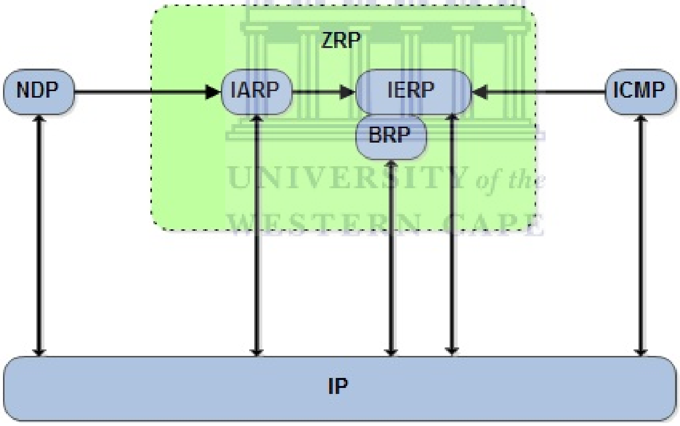


Figure 2.6: Zone routing protocol's components.
The components that complete ZRP. This illustrates the architectural design of ZRP. All components are attached to the IP layer. The proactive part is carried by link state IARP which uses NDP while IERP offers the reactive functionalities using BRP supported by ICMP.

Based on ZRP, [36] introduces an artificial intelligence approach in routing by using Genetic Algorithms (GA) to find multiple shortest paths to provide load balancing and tolerance. ZRP uses reactive routing to reach to destination beyond the routing zone in a purpose of finding the shortest path [36]. This returns one path which in time becomes congested or suffers route failure leading to an inconvenient routing dilemma. The protocol proposed by Kumar and Ramachandram [36], GZRP seeks to address this limitation by using genetic algorithms to find multiple shortest paths [20]. In case one path fails for unforeseen circumstances, other paths can be used to transmit packet to the desired destination.

Apart from genetic algorithms, different approaches have been exhibited taking advantage of the seamless prominence of zoning in hybrid architectures. The Global Positioning System (GPS) popularly applied in mappings in the modern era, has also been used in zoning protocols by Joa-Ng and Lu in the form of Zone-based Hierarchical Link State protocol (ZHLS) [28]. Unlike other zone-based protocols, ZHLS partition the network into no overlapping zones and also the zones do not have cluster heads like the other hierarchical protocols [28, 41]. It is organized hierarchically as in HSR and hence consists of two levels of topologies, node level and zone level. At the node level, the physical interconnectivity between nodes in a zone is known [41]. Zone level is how zones are connected to another. If one node in a zone is physically connected to another node in a different zone, then a virtual link exists [41]. There are two types of Link-State Packets (LSP) in ZHLS, node LSP and Zone LSP. Node LSP contains routing information of its neighbours and is propagated with the zone whilst zone LSP contains zone information which is propagated globally [41].

The core architectural design of zoning is broadly adopted in this family of protocols. Hybrid Ad-Hoc Routing Protocol (HARP) by Nikaein *et al.* inherits this from ZRP while as well as hierarchical structure from ZHLS and incorporates them together in one routing protocol [46]. HARP uses distributed dynamic routing to create routing zones which provide the proactive element in the protocol structure [46]. This protocol uses the notion of zone level stability to create and select paths [46]. Zone-level stability is an extension of node level stability exhibited with ABR (see Section 2.2.1). Unlike ZRP and ZLHS, HARP algorithm only deals with the finding and monitoring of a path between a source and a destination while DSR deals with the topology generation [46]. Thus HARP does the proactive duties while DSR (i.e. reactive protocol) offers the reactive duties completing the hybrid architecture.

The trade-off between proactive and reactive routing is a major concern in hybrid protocols but yet very fundamental. Ramasubramanian *et al.* introduces a sharp approach of striking a balance between the two [55]. Their protocol, Sharp Hybrid Adaptive Routing Protocol (SHARP) for a mobile ad-hoc network adjusts the degree of propagating proactive information as well as the reactive information [55]. This uses proactive-route dissemination and reactive-route discovery [55]. The protocol adapts between reactive and proactive routing by dynamically varying the amount of information shared proactively [55]. The protocol also utilizes the components of routing zones whereby routing inside the zone is maintained proactively while reactive routing is used to access nodes outside the proactive zones.

The zone radius thus plays a key role in partitioning the network into zones. The higher the radius, the larger the zones, lower the packet rate and variance delay which cultivates the increase in routing overhead; thus a shift to a more proactive architecture [55]. On the other hand, the low radius means reduction of routing overheads while it enhances chances of jitter and higher loss rates [55], a shift towards reactive architecture. Given this kind of trade-off, SHARP is able to even turn completely proactive by setting the zone radius to zero and reactive by setting the radii to equal the network diameter [55]. The trade-off thus poses a big challenge in many hybrid protocols. SHARP proactive routing is based on DSDV and TORA while its reactive routing is based on AODV [55].

Table 2.1 summarises the wireless mesh routing protocols discussed above. The table also highlights metric, topology and classification of each protocol. The metric refers to the routing metric or algorithm used by that protocol. The topology depicts the architectural structure whilst classification describes the type of that protocol.

Table 2.1: Summary of the mesh routing protocols.
This table shows a summarised review of the routing protocols.

Protocol	Reactive	Proactive	Hybrid	Metrics	Topology	Classification
AODV	X			Hops	Flat	Distance Vector
ABR	X			Link Stability	Flat	Source Initiated
AOMDV	X			Hops	Flat	Distance Vector
BABEL		X		ETX	Flat	Distance Vector
BATMAN		X		OGM Count	Flat	semi Link State
CGSR		X		Cluster Head	Hierarchical	Cluster/Link State
DSDV		X		Hops/Distance	Flat	Distance Vector
DSR	X			Hops	Flat	Source Initiated
FSR		X		Hops	Flat	Link State
GB		X		DAG	Flat	Link Reversal
GPSR		X		Forwarding	Hierarchical	Link State
GSR		X		Hops	Flat	Link State
GZRP			X	Hops/Zones	Hierarchical	Zone
HARP			X	Hops/Zones	Hierarchical	Zone
HSR		X		Cluster Head	Hierarchical	Cluster/Distance Vector
LMR	X			DAG	Flat	Link Reversal
MeshDV			X	Hops/Distance	Flat	Distance Vector
OLSR		X		Hops/ETX/ETT	Flat	Link State
SEAD		X		Hops	Flat	Distance Vector
SHARP			X	Hops/Zones	Hierarchical	Zone
SrcRR	X			ETX	Flat	Source Initiated
SRR	X			Link Stability	Flat	Source Initiated
TORA	X			DAG	Flat	Source Initiated
WRP		X		Hops/Distance	Flat	Distance Vector
ZHLS			X	Hops/Zones	Hierarchical	Zone
ZRP			X	Hops/Zones	Hierarchical	Zone

2.3 Link quality routing metrics

Routing protocols use metrics to select the best routing path. Several situation-aware routing metrics have been proposed, as well as applied, in many routing protocols. The

hop count routing metric is a simple computable metric that counts the number of hops between a sender and its destination. Hop count is commonly used in routing protocols such as AODV, DSR and DSDV [56, 64, 68]. Hop count is simple to compute when compared to other metrics, and this is the main reason it has been preferred by many routing protocols. However, hop count does not consider packet loss or bandwidth, and hence results in low throughput [64, 68].
Yang *et al.* proposed a situation-aware metric, Expected Transmission Count (ETX) which considers the number of MAC layer transmissions needed to successfully deliver a packet through a link [10, 68]. Thus addressing some of the drawbacks encountered in hop count metric. The ETX metric captures the effects of packet loss and path length. Each node broadcasts probe packets to its neighbours and they send a back a reply/report [10]. The metric is calculated by the number of probe packets received by its neighbour in both directions [64]. ETX is isotonic, thus ensures easy calculations of minimum weight paths [68]. The ETX metric does not consider bandwidth of the links, interference, or the link transmission variance [68].

The link quality $LQ_{i,j}$ between node i and j is the fraction of successful packets from node i received by j within a window period N [29, 68]. The neighbour link quality $NLQ_{j,i}$ is the fraction of successful packets from node j received by node i. Thus the ETX of a link $l_{i,j}$ is calculated as follows [29, 68]

$$ETX_{i,j}(n) = \frac{1}{LQ_{i,j}(n) \times NLQ_{j,i}(n)}. \tag{2.3}$$

The ETX of a route $R_{s,d}$ from source s and destination d is the sum of all ETX values of each 1 hop links. Thus formulated as follows [10]:

$$ETX_{s,d}(n) = \sum_{l_{i,j} \in R_{s,d}} ETX_{i,j}(n). \tag{2.4}$$

The ETX metric was optimised by Draves *et al.* to form a new Expected Transmission Time (ETT) metric aimed at overcoming the shortcomings of ETX [16]. Furthermore Esposito *et al.* implemented ETT for OLSR protocol and tested it on a tested, the results show that ETT performs better than ETX and has low packet loss ratio and low round trip time [17]. ETT takes bandwidth and link transmission difference into consideration for its path selection computation. The ETT of a link is measured by the expected Layer 2 durations it takes to successfully transmit a packet through that link [65, 68]. The relationship between ETT of a link l and ETX is defined by Yang *et al.* as follows [68]:

$$ETT_l = ETX_l \frac{s}{b_l}, \tag{2.5}$$

Where b_l is the transmission rate of link L while s represents the packet size. However since ETT uses a single path, channel interference (both inter-flow and intra-flow) becomes a major drawback in ETT like in ETX [68]. Intra-flow interference is interference between intermediate routers sharing the same path while inter-flow is between neighbouring routers competing for the same channel.
The intra-flow interference is however addressed in Weighted Cumulative ETT (WCETT) developed by Draves *et al.* to improve both ETX and ETT [16, 68]. This is done through the use of multi path channels. Given path p, WCETT is defined by Draves *et al.* as follows [16]:

$$WCETT(p) = (1-\beta) \sum_{link\ l \in p} ETT_l + \beta max_{1 \leq j \leq k} X_j, \tag{2.6}$$

where β is tunable parameter subject to $0 \leq \beta \leq 1$. X_j is the sum of transmission times on channel j and captures the intra-flow interference. The $max_{1 \leq j \leq 1} X_j$ is the maximum number of times of appearance of channel j along a path. The calculation of the WCETT metric can be interpreted in two ways; the estimation of the end-to-end delay of the path and the determination of the channel diversity of the path [37]. The first part of the expression is the sum of all the ETTs (SETT) for all links of the path, which corresponds to an estimation of the end-to-end delay experienced by the packet [37]. It also represents the total resources consumed by on that path [16]. Resources consumed are referred to as air time, which forms part of ETT. The second component represents the set of hops that have much impact on the path throughput i.e. path bottleneck [16]. This set of hops referred as Bottleneck Group (BGETT) by [37] is used to quantify the channel diversity. Thus the first component represents latency while the second one represents path throughput. The WCETT is the weighted average between SETT and BGETT, hence striking a balance between the two [16, 37]. However, since it is not isotonic, it has not been used by any algorithm [68]. Another drawback of WCETT it does not consider inter-flow interference and its effects.

The Metric of Interference and Channel-switching (MIC) addresses the shortcomings of WCETT by considering inter-flow interference as well as solving some of the non-isotonic effects. MIC estimate inter-flow interference by considering the number of interfering nodes in the neighbourhood and also uses virtual nodes to minimise route computation costs [11]. The calculation of MIC is based on ETT, thus the MIC metric of a path p is defined as [68]:

$$MIC(p) = \frac{1}{N \times min(ETT)} \sum_{link\ l \in p} IRU_l + \sum_{node\ i \in p} CSC_i, \tag{2.7}$$

where N is the total number of nodes and min(ETT) is the smallest ETT in the network. The other two components IRU (Interference-aware Resource Usage) and CSC (Channel Switching Cost) are defined respectively as [68]:

$$IRU_l = ETT_l \times N_l, \tag{2.8}$$

$$CSC_i = \begin{cases} w_1, & \text{if } CH(prev(i)) \neq CH(i) \\ w_2, & \text{if } CH(prev(i)) = CH(i) \end{cases}_{0 \leq w_1 < w_2}, \tag{2.9}$$

where N_l is the set of neighbours that interferes with the transmissions of link l, CH(i) is the channel assigned for the transmissions of node i and prev(i) represents the previous hop of node i along path p. The IRU_l represents the aggregated channel time of neighbouring nodes consumed by link L transmissions. It favours paths that consumes less channel times at their neighbours, thus capturing inter-flow interference [68]. The CSC component on the other hand favours paths with more channel assignment diversity and hence represents the intra-flow interference. It gives high weight to paths with consecutive links using the same channel than those that alternate their channel assignments [68]. Although MIC is not isotonic, Yang *et al.* proved that it is possible to introduce virtual nodes which decompose MIC into isotonic link weight assignments [68]. This is because the non-isotonicity of MIC is caused by different increments of path weights due to additions of links on a path. They also showed that there exists no scheme that can turn WCETT into an isotonic form because its non-isotonicity depends on $maxX_J$ component. Thus the weight increment of adding a link to a path depends on the frequency of the channel on the path. The combination of channel assignment can become infinite as the length of the path increases unlike in MIC where the possible assignments of channels for a preceding link are limited, therefore virtual nodes cannot be used.

2.4 Moving averages

The idea of using moving averages is not new. It is commonly used in economic systems for computing and plotting of stock markets. A Moving Average (MA) is an arithmetic

result calculated by averaging a number of past data points [3]. A Simple Moving Average (SMA) is calculated using the mean of a given set of values. The sum of the set is divided by the number of elements in that set. It is similar to a statistical computation of a mean yet different by the fact that only a recent N number of data values are considered in SMA. The drawback of SMA is the fact that all points in the data series are weighted the same irrespective of where it appears in the sequence [3]. To address this setback on the basis that recent data is more significant than old data and ought to have a greater influence in the final results, various types of moving averages have been invented including Weighted Moving Average (WMA), Exponential Moving Average (EMA) and Double Exponential Moving Average (DEMA). These types of MA's are thus more responsive to recent data.
Weighted Moving Average (WMA) is another type of MA that also gives more weight to recently received data values. WMA multiplies the most recent value with its sequence value and monotonically decreases with iterations [18]. For example, given a set of 10 values, WMA would multiply the value at index 10 with 10 and value at 9 with 9 and so forth.

Exponential Moving Average (EMA) is another type of MA that gives more weight to recent values in order to make it more responsive newer information [3]. In relation to routing protocols more weight is applied to more recent OGMs, for example, for precise current link quality estimation. In stock market analysis, EMA uses the formula [43]:

$$E_{t+1} = \alpha P + (1 - \alpha) E_t, \tag{2.10}$$

where the EMA at time t is E_t and the next EMA is E_{t+1} and P is the current price and $\alpha = \frac{2}{1+N}$ is the smoothing factor and N is the number of time periods. Adya *et al.* applied this in network protocols in terms of round trip-time formulated as [3]:

$$S_{t+1} = \alpha R + (1 - \alpha) S_t, \tag{2.11}$$

where S_{t+1} is the SRTT at time t+1, S_t is the SRTT at time at the previous time period t and α is a smoothing operator between 0 and 1. The above formula has been used as the basis of per-hop Round Trip Time metric (RTT). Each node broadcasts a probe packet to its neighbours every 500 milliseconds [16]. The neighbours respond with an acknowledgement probe. The sender then measures the round trip delay and calculates an exponential moving average for each neighbour. The metric covers a lot of link quality factors which [16] identifies as: first, the probe/probe back experiences a queuing delay when the node or neighbour is busy resulting in high RTT. Secondly if the nodes in the neighbourhood are busy, the probe /probe back experiences delay due to channel contention also resulting in high RTT. Thirdly a probe/probe back may have to be retransmitted several times if the link is highly loaded or is lossy. The metric however leads to lot of route instability since it load-dependent [16]. It also suffers from overheads when the measuring round trips. Draves *et al.* reduced the probe size to 137 bytes which is still big and it also does not take link data into account [16]. SMA and EMA are also often used as estimators to present performance comparison during evaluations of different routing protocols or link quality metrics [10].

2.5 Summary

In this chapter we presented work related to dynamic wireless mesh network. Dynamic topologies are very flexible in terms of scalability, mobility and connectivity. However they are characterized by link instabilities due to rapid topological changes. The mobility of nodes causes an unstable link quality which negates throughput and ultimately affects the quality of service on the network. Link quality is very important because it determines the reliability of the network links which carry information. Many different approaches have been devised to deal with the routing challenges under such conditions; intelligent algorithms, routing protocols and link quality metrics. Tsirigos and Haas proposed multipath routing where data load is divided and broadcast along the available links surrounding a source or intermediate node [61]. Routing protocols forms a core of the solutions reviewed in this chapter, this consist of reactive, proactive and hybrid categories.

Reactive protocols are on demand, thus triggered by a certain call on the network like data to be sent. Most reactive protocols incur high latency because of the periodic beacon updates. Proactive protocols, referred to as table-driven protocols, maintain routing tables with information of the entire network which is updated periodically to maintain its adequacy. It consists of link-state, distance-vector and cluster-based classes of protocols. This protocol scheme provides good reliability and low latency; however it entails high overhead resulting in excessive consumption of bandwidth and hence suffers from scalability issues. Another proactive protocol, namely BATMAN, addresses these setbacks by reducing the size of the control packets extensively. Hybrid protocols try to overcome the shortcomings in both reactive and proactive schemes by incorporating them together. This leaves the trade-off between the two as a major concern.

Routing protocols use routing metrics for their routing computational decisions. They are the driving force behind protocols. Routing metrics are often the major difference in various routing protocols. Many of them are derivations of mathematical algorithms. Moving averages are commonly used in economic systems and they are recent past data manipulation methods.

Chapter 3

Methods

This chapter discusses methods selected for this research and how they were applied and is divided into four sections. Section 3.1 looks at the research gaps and challenges identified from the literature reviewed in Chapter 2. A research question is restated Section 3.2 and Section 3.3 discusses methods answering it, this includes design and evaluation. The experimental design consisting of an experimental test bed and procedure is discussed in Section 3.4. This section also presents different perspectives of experimental designs from the reviewed literature. Section 3.4.3 discusses the implementation of the methods as well as a thorough description of the proposed algorithm together with its pseudo-code.

3.1 Limitations and challenges of the related work

In Chapter 2 we studied related literature concerning wireless mesh routing protocols, routing metrics and moving averages. The routing protocols consist of three classifications namely; reactive, proactive and hybrid protocol schemes. Arguably reactive protocols suffer from high latency on the network due to its route discovery mechanism and thus results in slow throughput on the network [53, 68]. This protocol scheme is not dynamic as it fairs poorly in dense networks with unstable network patterns [53]. Proactive protocols on the other hand endure high overheads on the network caused by periodic propagation of control messages and hence results in high bandwidth consumption. It is said that the control messages are often too big in most of these protocol schemes except BATMAN. The BATMAN protocol's control messages are relatively small, 53 bytes. Despite this, BATMAN does not consider link quality in its routing decisions which works to its detriment. The Hybrid protocol scheme, which incorporates reactive and proactive protocols, has its own limitations as well [30]. The biggest challenge for this scheme is the trade-off between both protocols because the degree of adjusting between the two needs precise consideration and hence remains a major concern [53].

The routing metrics which are the driving force behind the routing protocols, also have some challenges and limitations. The traditional Hop-count metric is not situation aware and does not consider packet loss and bandwidth. Arguably many routing metrics such as ETX, ETT, WCETT and MIC are situation aware but still have some limitations which deter the smooth routing. Even though ETX takes packet loss into account, it still does nothing about interference. ETT, like ETX uses single path which leaves the chances of route congestion wide open [65]. WCETT overcomes a lot of drawbacks encountered in ETX and ETT, most notably the intra-flow interference. Inter-flow interference is not however accounted for. This metric is however not isotonic and not preferred by any routing protocol [68]. Although MIC seems to overcome all these hurdles hypothetically, Jiang *et al.* showed that it is unrealistic because the interference range is higher than the transmission range [27]. BATMAN protocol ideally improves routing, however, its metric is based on the number of OGMs received in a current window. Given the rapid changes in dynamic topological structures, the current situation of the network state needs to be considered when routing decisions are made. Even though BATMAN-adv brought in new elements of link quality considerations; they still depend on the OGM count. Thus we are saying that the order of the OGMs in the sliding window should not have the same impact and therefore recent OGMs should assume more weight in the final routing decision than the old ones. Another limitation of BATMAN-adv is not designed to run on the mobile phone yet.

3.2 Research question

The research question was presented in Section 1.3. This question's main focus is aligned with the protocol's routing decisions. We want to model the BATMAN's OGM count in such a way that it prioritizes recently received OGMs in determining the strength of the link. Recently received OGMs provide precise indication of the network state at any particular point. Thus if less OGMs have been received recently, it is an indication that there might be some disturbance caused by certain network factors which would bear a negative impact on the overall performance. Factors such as congestion, interference and low link strength could all lead to packet loss on the network [2, 69]. On the other hand if a lot of OGMs are received then it implies that the link is lively. However, the degree of recentness needs thorough consideration. According to BATMAN this phenomenon of recentness is represented by the sliding window which in most cases is 60 seconds. A lot could still happen in a 60 seconds period and hence we are saying that the degree of recentness needs to be applied in a monotonic sense relative to succession. We can achieve this by using moving averages algorithms and bring about quality of service. A weighted moving average was thus adopted.

3.3 Research methods

This section discusses the methods used to carry out the research requirements. This section consists of two subsections namely, design methods and evaluation methods. The design methods subsection outlines the research guideline and structure. The evaluation methods subsection discusses data evaluation and analysis methods.

3.3.1 Design methods

The design of the research project follows the software engineering Spiral model methodology proposed by Boehm [7]. Pressman [54] defines it as an evolutionary software development process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model. Boehm and Hansen further describes it as a risk model generator used to guide multi-stakeholder concurrent engineering of software intensive systems [6].

The reason we adopted this model is based on the nature of the research project whose main aim is to optimize an existing project and hence spiral is ideal guideline for this study. The project builds on the existing BATMAN-adv and therefore not aimed at building a new protocol instead improving the current best one. The model involves four major tasks as shown in Figure 3.1 below. This model is structured in a circular mode where each circle (iteration) encompasses all the four steps of Analysis, Evaluation, Development and Planning.

The first phase of this model involves determining objectives, alternatives and constraints of the project. On this note different protocols were analysed through literature study and BATMAN was chosen with the objective of optimizing it. The selection of BATMAN-adv is further justified by the work of Abolhasan *et al.* where three current proactive were evaluated and their findings confirm the superiority of BATMAN-adv over the others in many aspects [1]. BATMAN layer 2 version (BATMAN-adv) was chosen instead of the layer 3 version (BATMANd). This is despite the fact that the layer 3 version can run on mobile phones and can support many nodes, It was discovered to have routing flaws caused by the symmetric links. This protocol was further analysed where gaps and challenges were identified (Section 3.1) leading us to the rational justification

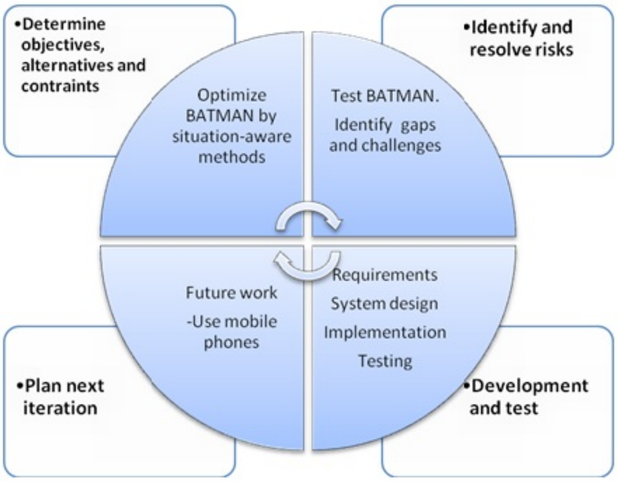


Figure 3.1: The spiral model. [7].
The research roadmap illustrate the guideline followed to carry out the research. The processes are repeated in a clockwise iterative fashion until the last prototype.

of the ultimate objectives of the study. Situation-aware methods are thus proposed as alternatives. The constraints identified involve testing; the initial objective was to test on a rural environment with mobile phone handsets in action. However due to time and resources the rural environment is emulated through test bed while the mobility is ensured by moving laptops.

The next phase was to evaluate the alternatives in relation to the constraints [7]. This was to identify any risks involved and developing suitable solutions. This phase also involved prototyping. The first prototype of this study is the BATMAN-adv original which is also used as a reference point. After the analysis, evaluation, development and testing follow. The requirements of this study are based on the social challenges observed as well as the research work already conducted such as Brewer *et al.* which suggests that there is a big concern with regards to information access [9]. The design is covered in Section 3.4 while implementation and testing are covered in Section 3.4.3 and Chapter 4 respectively. The spiral's last phase is planning, this represents future work that we identified.

3.3.2 Evaluation methods

Quantitative empirical methods are used to evaluate the project. We adopted quantitative methods since our evaluation is pragmatic, numeric and does not involve users. Quantitative empirical methods evaluate data using statistics, mathematics and computational techniques. The aim of this method is to classify features, count them and construct statistical models in an attempt to explain what is observed [40]. This kind of research method is perceived by many researchers as objective [48] as it seeks precise measurement and analysis of target concepts. This model generates rational knowledge which can be objectively used to solve real-life problems [5]. Olivier [48] however suggests that this model cannot be used to answer some pressing problems. Another disadvantage is the fact that the findings are not user-based and hence lack human perspective. Traditionally researchers use questionnaires and surveys to gather data from users but in this case we use softwares for data collection and hence performance testing (with users) is not necessary.

Quantitative empirical methods work handily with statistics. Statistical methods are used to group different data aspects and relate them to produce quantitative analysis. Statistics is the study of collecting, organizing and interpreting data [58]. Through measures such as size, mean, dispersion and graphs to present data statistically. Size indicates the number of participant members in a sample. n is used to denote it (and m when dealing with two groups). The mean is used to average a data set. Dispersion consists of variance and standard deviation which measures variability or diversity between data elements i.e. how far numbers lie from the mean. The standard deviation is the square root of the variance. The graphs consist of line and bar graphs. Line graphs show relationship in a collected data while bar graphs provide a visual comparison of values [48].

3.4 Experimental design

Through an analysis of work reviewed in Chapter 2, it was discovered that simulation is a preferred evaluation method. According to Breslau *et al.*, Simulations offer a simple, seamless and inexpensive way of testing different aspects of network protocols [8]. Simulations loyalists believe it provides efficient experimentation. They further argue that simulation provides larger scale protocol interaction in a controlled environment and easy comparison of results across different research efforts [8]. However Haq *et al.* and Cavin *et al.* state that, for simulation results to be meaningful they have to match reality as closely as possible (test bed results) [13, 21]. Haq *et al.* conclude that simulation results are only close to test bed results at lower traffic rates (bandwidth) and thus argue at high bandwidth wireless network simulators results may not evaluate the routing protocol correctly [21]. In addition, Cavin *et al.* states that there is a scarcity of real experiments that demonstrate the correctness of wireless network simulators [13]. Cavin *et al.* tested a flooding algorithm with three popular simulators namely, Network Simulator 2 (NS2), Optimized Network Evaluation Tool (OPNET), and Global Mobile Information System Simulator (GLoMoSim) and found very divergent and incomparable results [13]. They discovered that the results not only differ quantitatively but also qualitatively which suggest that the general behaviour also varied [13]. For this reason they concluded that simulation evaluation is less credible.

NS2 seems to be the most popular simulator judging by its popular utilization in the literature review. Munaretto *et al.* and Divecha *et al.* tested the proactive protocols, OLSR and DSDV respectively using this model [15, 42]. Reactive protocols such as AODV were also tested using NS2 [38]. DSR, also a reactive protocol, was tested on both NS2 and OPNET by Gupta *et al.* and Khatri *et al.* respectively [18, 35]. SHARP, a hybrid protocol and a proactive FSR were simulated using GLoMoSim by Ramasubramanian *et al.* and Pei *et al.* respectively [51, 55].

Although test beds have their limitations, we believe that they are well suited for this kind of research study project because they offer rigorous, transparent and replicable testing. They are closest to real a situation environment and offer a good platform to test the feasibility of a protocol in reality. Most test bed pessimists argue that test beds are expensive [8]. In contrast recent developments in this area offer affordable infrastructure such as mesh potatoes (see Section 3.4.1) for efficient experimental and deployment use.

The OLSR protocol has also been evaluated on a test bed by Johnson *et al.* [31]. They tested two metrics namely, standard hysteresis metric and ETX metric in a 7×7 grid of spaced Wi-Fi nodes [29]. Draves *et al.* tested the performance of OLSR and BATMAN protocols on a 7×7 grid of closely spaced WI-Fi nodes and showed that BATMAN outperformed OLSR [16]. Albolhasen *et al.* evaluated the current best proactive protocols of OLSR, BATMAN and BABEL on a test bed in an attempt to gauge the real-world performance of the trio [1].

3.4.1 Experimental test bed

The experiments were conducted in two setup phases namely, initial experiments and final experiments. The initial experiment consisted of a small scale network test bed with only four nodes while in the final experiments a larger scale network test bed of 12 nodes was used. This was aimed at evaluating impact of scalability on the performance of both protocols.

The small scale network was conducted in a single room computer laboratory with personal computers (PC) as nodes. Two PCs and 1 laptop were used as shown in Figure 3.2. All the nodes ran Linux kernel version 2.6.32-31 with IEEE 802.11bg network cards.

The system specifications of the computer nodes are as follows:

PC :

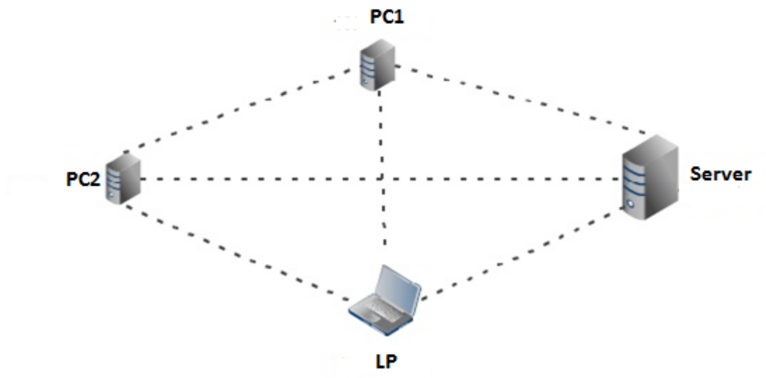


Figure 3.2: The small scale network test bed
The initial experiment consisting of four computer nodes.

Processor: Intel (R) Core (TM) i3 CPU M 540 @3.07 GHz (4 CPUs), 3.1 GHz
Memory : 4096 MB RAM
Laptop : HP ProBook 6450b
Processor: Intel(R) Core (TM) i5 CPU M 450 @2.40 GHz (4 CPUs), 2.4 GHz
Memory : 4096 MB RAM

In the larger scale experimental network test bed setup we tried as closely as possible to emulate a real life network. The construction of the mesh test bed consisted of affordable tools which are suitable for either a rural or urban environment. We used Mesh Potato (MP) devices (see Figure 3.3) in addition to the four computer nodes. Mesh potato provides inexpensive telephony and Internet services suitable for underprivileged and marginalized communities whereby these services are either non-existing or very expensive [63]. This device caters for everyone as it falls within the economic means of any ordinary working class South African. This device is a combination of a Wireless Access Point (AP) and an Analogue Telephony Adapter (ATA). The device can be seen in Figure 3.3. The device's Wireless LAN is equipped with IEEE 802.11b/g and 2.4 to 2.462 GHz Frequency Band whith a range of up to 400m [63].

Wireless mesh nodes were disseminated across the computer science departmental offices. Figure 3.4 shows the geographical floor structure and the location of the nodes while Figure 3.5 shows the mesh cloud of the same network.

The experimental test bed consisted of 12 nodes as shown in Figures 3.4 and 3.5. The network consisted of 8 mesh potatoes and four personal computers (PC) as access points running in ad-hoc mode. All nodes were initially equipped with BATMAN-adv version batman-adv-2011.2.0 and thereafter the modified version. The mesh potato nodes were labelled MP1-to-MP8, the PC nodes range from PC1-to-PC2 and laptops LP1-to-LP2.

Figure 3.3: The mesh potato.
The mesh potato device which provides telephony and wireless Internet access. A wireless mesh network created with these devices enables communities to make free local telephone calls. For internet access, a gateway node connected to the Internet Service Provider (ISP) is required.

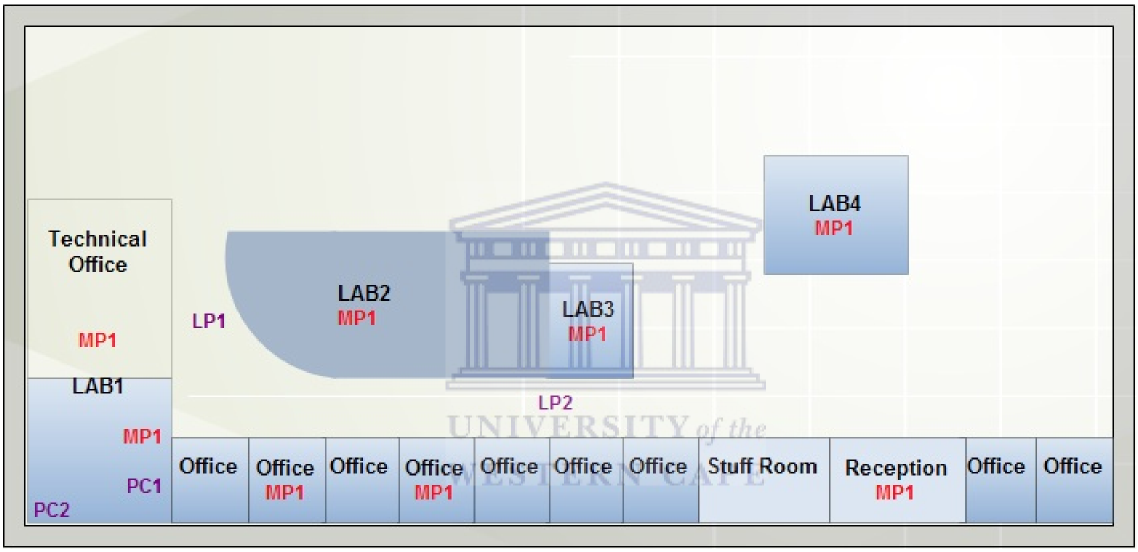


Figure 3.4: The floor structure.
The geographical floor structure of the building where the mesh nodes were located. The offices are separated by walls which reduces Wi-Fi signal range.

PC1 was configured as a server while LP, LP1 and LP2 were the laptops which were used to carry out the mobility in both setups. The 8 mesh potatoes were distributed in different rooms in the departmental offices (see Figure 3.4). Since the PCs were also in ad-hoc mode, they carried out full functional capabilities as access points and thus also joined in the mesh. Nonetheless the connectivity between the PC nodes and other distant nodes is omitted in Figure 3.5. The connectivity relationship of the nodes is shown by Table 3.1. The table shows one-to-one mapping amongst the nodes which determines the hop relationship in the network. Direct connection, i.e. one hop is denoted by the letter D while IND denotes indirect connectivity of more than 1 hop. The one-to-one connectivity was determined by the mesh potato's Asynchronous Telephone Adapter (ATA) which allows telephone calls between in range nodes. Ping was used to test the connection between PC nodes.

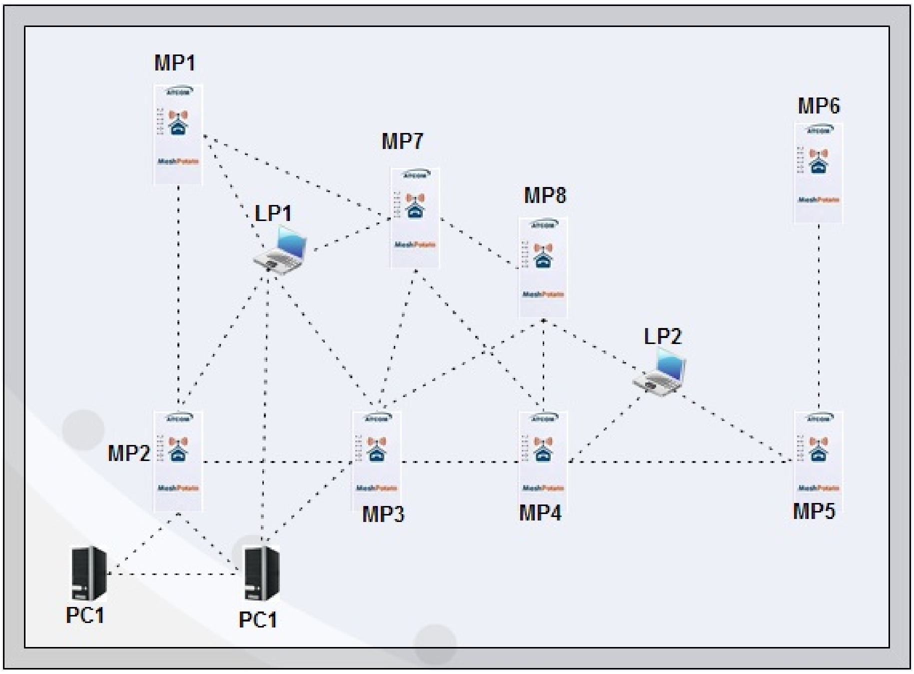


Figure 3.5: The network test bed.
The network test bed showing 8 mesh potato nodes and accompanying 4 PC nodes.

Table 3.1: Nodes connectivity.
The table shows the network connectivity between nodes, it is a one-to-one (1:1) mapping relationship.

Nodes	A	B	C	D	E	F	G	H	PC1	PC2	PC3	PC4
A		D	D	IND	IND	IND	D	D	D	D	D	IND
B	D		D	D	IND	IND	D	D	D	D	D	IND
C	D	D		D	IND	IND	D	D	D	D	IND	IND
D	IND	D	D		D	IND	D	D	IND	IND	D	D
E	IND	IND	IND	D		D	D	D	IND	IND	D	D
F	IND	IND	IND	IND	D		D	D	IND	IND	D	D
G	D	D	D	D	D	D		D	D	D	D	D
H	D	D	D	D	D	D	D		D	D	D	D
PC1	D	D	D	IND	IND	IND	D	D		D	D	IND
PC2	D	D	IND	IND	IND	IND	D	D	D		D	IND
PC3	D	D	D	D	D	D	D	D	D	D		D
PC4	IND	IND	IND	D	D	D	D	D	IND	IND	D	

3.4.2 Experimental procedure

The experiment was designed to compare the performance of unmodified BATMAN-adv with our modified version on a dynamic mesh network with and without congestion. Besides congestions, other factors such as mobility and scalability were given a special attention; Mobility to fully explore the impact of dynamic nodes in a wireless mesh network and scalability to measure the effects of network growth. BATMAN-adv's experimental results were used as a benchmark of our findings. Figure 3.6 shows the evaluation comparisons ‘within’ the protocols and in ‘between’ the protocols’ distinct test bed setups. The ‘within’ refers to the performance evaluation between the two protocols; this encompasses both congestion and mobility. The ’between’ on the other hand refers to the performance evaluation between the small scale test bed and the larger scale test bed; this evaluates the impact of scalability.

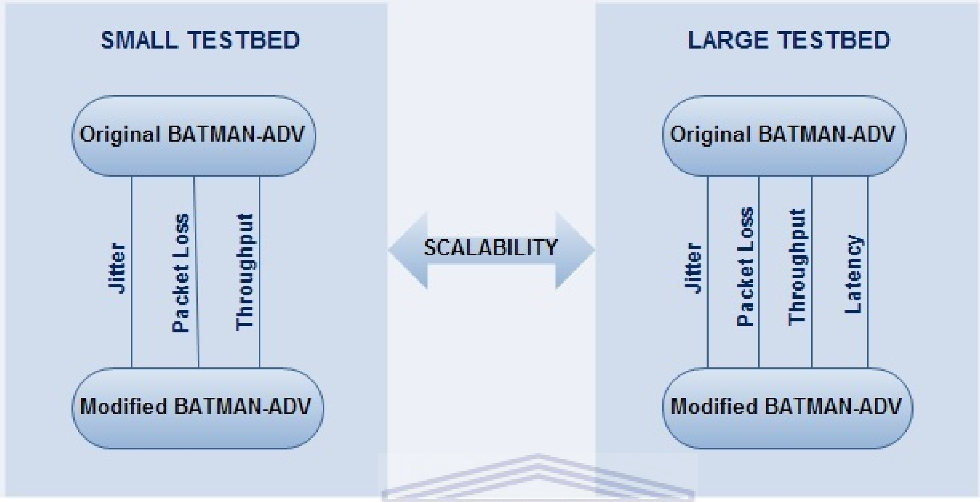


Figure 3.6: The evaluation structure.
The network test bed showing 8 mesh potato nodes and accompanying 4 PC nodes.

The main objective is to show that situation-aware routing is viable and effective in a dynamic WMN. The test parameters examined were Latency, Jitter, Packet Loss and Throughput. Although the test bed consisted of 12 nodes, we collected data on only the 4 PC nodes. The reason behind this is the existence of proven and reliable network performance monitoring tools applicable on PCs. This motivates our choice to use PC nodes for data collection whilst the mesh potato nodes act as supporting pillars of the network. This was to test the protocol’s feasibility under unstable conditions. The laptops were moved around during the experimentation’s transmitting period. Each laptop was moved around in the same room for few minutes recording the performance and then taken off the room in the vicinity of the mesh network.

IPERF was used on all of the nodes to conduct the tests. IPERF (Intelligent PERFormance Prediction) [23] is a network performance measurement tool written in C++; It is a tool used to measure the bandwidth and the quality of a network link. IPERF consists of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) measurements. TCP uses processes to check packets that are correctly sent to the receiver. TCP is reliable because packets are guaranteed to reach their target unless there is a complete connection failure due to unforeseen external circumstances. In TCP, the order in which packets are sent is upheld ensuring that packets arrive respectively in order. In case where packets arrive out of order, a ’resent request’ is sent so that packets can be resent thus making it heavyweight. In addition TCP was used to measure latency using Internet Control Message Protocol (ICMP) packets of Ping. On the other hand UDP does not check packets before they are sent; this makes it quicker than TCP. UDP is therefore unreliable, not ordered and lightweight. The performance metrics are listed below.

TCP measurements:

- **Latency**: Measures the delay of packets from source to destination i.e. one way. Round Trip Time (RTT) on the other measures the delay bidirectionally. The delay is measured using Ping Unix command.

UDP measurements:

- **Packet loss**: This is the total data packets dropped before reaching the destination (measured in percentages). it is a very critical performance metric because it affects the overall throughput on the network [55]. Packet loss has been used regularly in both the simulation and test bed experiments. Khatri *et al.* and Marina and Das applied this metric in simulation experiments using OPNET and NS2 respectively [35, 39].
- **Jitter**: This measure includes all the possible end-to-end delay caused by factors such as buffering, queuing delay, retransmission delay at MAC, propagation and transfer rate [18]. Abolhasan *et al.* evaluated real world performance of some proactive protocols using this metric in a test bed. This measure has also been used extensively in simulation environments such as NS2 [18, 24, 39], OPnet [35, 55] and GloMosim [50].
- **Throughput**: This represents the ratio of the total delivered data packets to those generated by the initiator. This is a very important determinant of an efficient routing scheme and thus used frequently in most performance evaluations. This includes simulations in NS2, OPNET and GloMoSim as well as test beds.

Node PC1 was used as a server (receiver) whilst the others were clients (senders). We configured IPERF to send packet flows representing voice packets to the server. We set a transfer interval of 60 seconds with a report back of 10 seconds. This was run 10 times for each parameter herein referred as 10 flows. During the transfer interval, IPERF sent about 4000 User Datagram protocol (UDP) packets, about 665 each 10 seconds, with a maximum fixed size of 1500 bytes. The parameters were tested with a selection of transfer rates and buffer sizes. The default settings were 1 megabytes per second (MB/s) of bandwidth and 41 kilobytes (KB) for the buffer size. The transfer rate was regulated over 1 MB, 100 MB and 150 MB speeds whilst buffer size was varied over 41 KB, 31 KB and 11 KB. The first comparison combination consisted of all the transfer speeds with the default buffer size of 41KB. The second comparison combination applied the buffer size variations to the default transfer rate of 1 MB/s. Lastly, the 150 MB/s rate was applied to the 11 KB buffer size to achieve maximum congestion of the compared rates and buffer sizes.

3.4.3 The situation-aware algorithm

BATMAN uses control packets called originator messages (OGMs) for it's routing decisions. For the proposed method the same criterion with some added situation-aware features was adopted. Given the mobility of mobile nodes, rapid topological changes in a hybrid mesh network are inevitable. Thus, the ideal approach is to take the current network situation into consideration when making routing decisions. In BATMAN, the best link is measured by the highest number of OGMs (see Chapter 2) received from the destination over a current sliding window. Much can happen within a second in an ad-hoc wireless network especially with mobile nodes. Any link with a sliding window that records a lot of OGMs at the beginning and fewer at the end due to superior link strength at the beginning stands a chance of being the best as opposed to the one that records a lot towards the end but fewer in total. Table 3.2 shows a practical example of the above scenario. Suppose one has a sliding window of 10, link L1 records [1111100000] with 6 OGMs at the front, and link L2 [0000001111] with 5 OGMs seen at the end. BATMAN will chose L1 as the best next hop because of the higher number of OGMs, but actually, the current best option would be L2 because the most OGMs would have arrived there more recently.

The situation-aware method, modified BATMAN-adv, prioritizes the recently received OGMs in the sliding window. Recently received OGMs provide a precise indication of the network state. The status of the network should be taken into account for better routing decisions. The algorithm uses the weighted moving average approach in maintaining a smooth ageing technique. The idea is to weight the OGMs according to their succession sequence. This algorithm can be seen as a Packet Ageing Algorithm since packets lose value with age. The algorithm sums the indices where the OGMs were recorded in a window. Thus from the example above, we would have link L1: 1+2+3+4+5 = 15 and link L2: 7+8+9+10 = 34 and therefore would correctly choose L2 over L1. This is a more accurate numeric representation describing the current situation of the two links.

Table 3.2: Sliding window of 10 seconds.
The table shows an example of a sliding window of 10 seconds. Link L1 records most overall OGMs to Link L2 in the sliding window but the later is the best routing option because its OGMs are more recent.

Seconds	Sequence No.	Link L1	Link L2
1	**1**	1	0
2	**2**	1	0
3	**3**	1	0
4	**4**	1	0
5	**5**	1	0
6	**6**	1	1
7	**7**	0	1
8	**8**	0	1
9	**9**	0	1
10	**10**	0	1
		6	5

The pseudo-code presented in **??** illustrates the high level description of the above algorithm (Section 3.4.3). This forms the core of the modifications we made on BATMAN-adv source code. The code used binary operations to handle the bit arrays. The algorithm is implemented directly inside the BATMAN-adv source code replacing the original algorithm in the file *bitarray.c*.

Algorithm 2 The pseudo-code of the situation-aware algorithm

```
int $i, check, count \leftarrow 0$
unsigned long word
for $i \leftarrow 0 \rightarrow Number_{o}f_{u}nsigned_{w}ords$ do
  $word \leftarrow unsigned_{w}ord$
  int $j \leftarrow numberofbitsintheunsignedword, k \leftarrow 1$
  while $j > 0$ and $k \leq j$ do
    check position is 1
    if $check = 1$ then
      $count \leftarrow count + 1$
    end if
    $j \leftarrow j - 1$
    $k \leftarrow k + 1$
  end whilereturn count
end for
```

3.5 Summary

This chapter discussed methods used to carry out the research as well as the implementation. Since this study sought to fill the void left by many existing routing protocols; we looked at those challenges at length. Limitations of routing metrics were also discussed. The challenges discussed played a major role in determining the main research question. The research question is formulated as: Can a situation-aware method improve the BATMAN routing protocol to realize better QoS in a dynamic wireless mesh network with mobile nodes? The research methods including the design and methodology used to tackle the research question were discussed in Section 3.3. The evaluation methods used involve Quantitative empirical and statistical methods. Quantitative empirical methods were used to collect the evaluation data while statistical methods were used to present the data. The experimental design section first looked at methods used in the work related to the study of this thesis. Although people used Simulations, this method did not suit the experimental structure as well as the scope of the study and therefore we used test beds. We further discussed the experimental test bed in detail, its structure, connectivity relations in between the nodes, and the procedure followed to carry out the experimental tests. The procedure involved the usage of IPERF, which is the performance monitoring tool used for this study. The chapter also presented implementation in the form of the algorithmic design. BATMAN-adv routing algorithm and its challenges as well as the proposed algorithm were also discussed. The high level design description of the algorithm, the pseudo-code for the proposed method was presented as well.

Chapter 4

Results

This chapter presents and discusses the results obtained by evaluating the situation-aware techniques based on the BATMAN-adv routing protocol on a real network test bed. The results are divided into two sections: Section 4.1 focuses on the preliminary results conducted on a small experimental network test bed and Section 4.2 discusses the final results obtained on an improved larger scale experimental network test bed with increased mobility.

4.1 Initial results: small scale test bed

The following results were obtained from the small scale experimental network test bed. The construction of the test bed is described in Section 3.4.1. The test bed consisted of four nodes based in single computer laboratory. The results presented include jitter, packet loss and throughput.

4.1.1 Jitter

The variation of packet latency across a network, known as jitter or packet delay variation (PDV) shows a significant difference between protocol sets. The BATMAN-adv original shows the best (low) PDV of less than 55ms across all variation settings as shown in Table 4.1. The PDV is consistent irrespective of the transfer rate or the buffer size. Node LP, which had mobility throughout the tests, exhibits an overall average of 30.80ms across all variations of settings while nodes PC1 and PC2 are 38.18ms and 42.40ms, respectively. On the other hand, the situation-aware BATMAN-adv lacks consistency as some points rise abruptly, reaching 336.2ms, while the lowest is 24.24ms in line with the original protocol. PDV consistently increases and appears to do so independent of the variation settings.

Table 4.1: Small scale jitter results.
The table compares the jitter results between BATMAN original and the situation-aware version across different congestion level. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented.

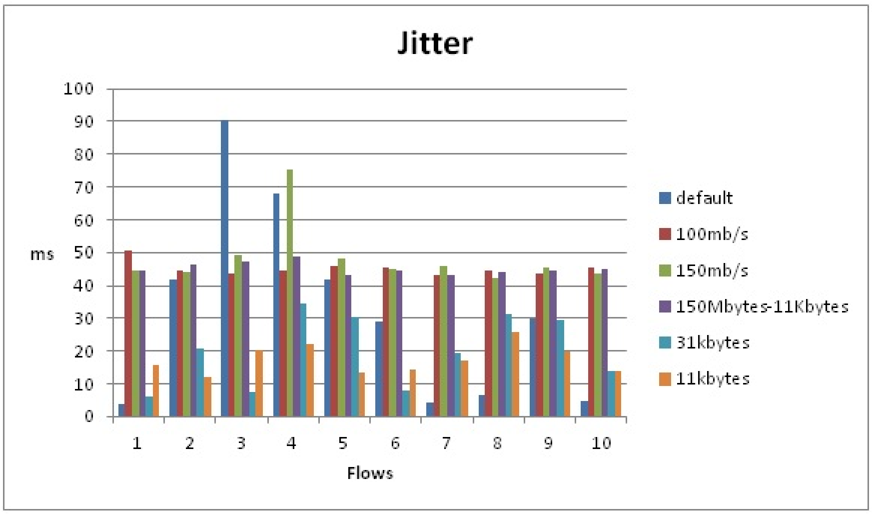
Jitter							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
PC1	32.06	45.24	48.35	45.19	20.04	17.47	38.18
LP	27.42	41.62	43.43	28.08	13.45	20.43	30.80
PC2	36.91	53.61	54.78	37.83	28.85	32.39	42.40
Situation-aware BATMAN-adv							
PC1	150.62	38.59	336.20	109.30	59.75	101.80	132.71
LP	0.25	102.6	142.50	179.50	258.40	131.30	162.76
PC2	216.58	58.18	52.80	45.80	50.72	24.24	74.72

Figure 4.1 depicts a clear picture of how jitter from the two protocols varied before they could be averaged in Table 4.1. The situation-aware protocol shows a high level on inconsistency in Figure 4.1(b). In most of the flows, the 150Mbytes transfer rate endured the highest jitter with flow 1 reaching an amazing 1486.608ms. Most of the values however fall below 50ms which is on par with the original protocol but due to it's inconsistency the overall average is high. The original protocol achieved a more consistent and standard jitter and hence is the better of the two.

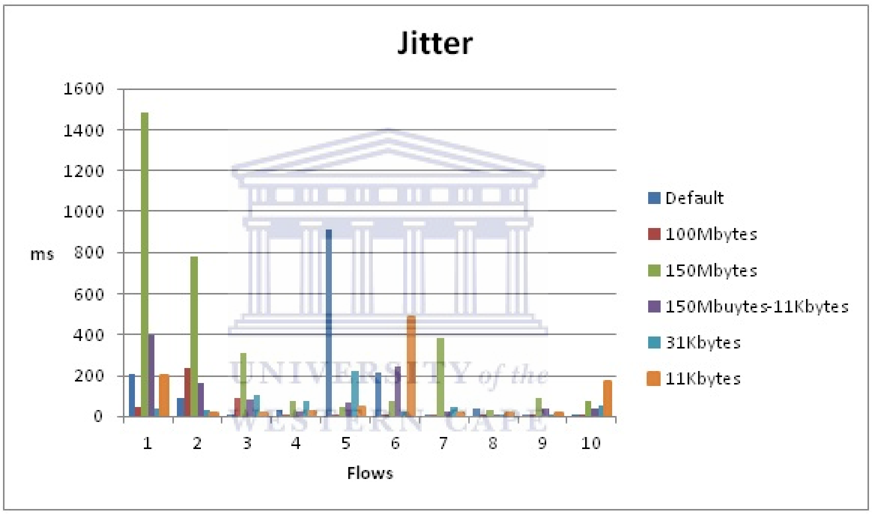
4.1.2 Packet loss

The average packet loss results of BATMAN-adv original appear inconsistent in the baseline measurement. The average across all variation settings i.e. from default 150Mbytes-11Kbits (see Table 4.2) exhibits some inconsistencies as compared to our situation-aware version which stands at an average of 3.2% for all the links (PC1 = 3.52%, LP = 3.05% and PC2 = 3.09%).

BATMAN original has values: PC1 = 2.94%, LP = 5.42%, PC2 = 0.83%. The most distinctive and significant factor in this case is the consistency of packet loss for BATMAN-adv across all settings while the modification shows reduction as per variation settings. At default, the average packet loss on the three links is about 8%. The loss rate then reduces proportionally to the transfer rate and buffer size. This shows that situation-aware routing metrics perform well on larger and inconsistent networks with congested links. The results show no practical relation between jitter/PDV and packet loss.



(a) Jitter from node PC1 to the server for BATMAN-adv original.



(b) Jitter from node PC1 to the server for Situation-aware BATMAN-adv.

Figure 4.1: Compares jitter results between the two protocols
The jitter results comparison between the BATMAN-adv original and the situation-aware version on the small scale network test bed. The original protocol has the lowest and consisted jitter as compared to the situation-aware version

Figure 4.2 depicts an illustrative example of packet loss on the mobile node LP. The packet loss rate on the situation-aware protocol was better than the original. Figure 4.2(a) shows an average of more than 4% for the original protocol while the situation-aware, (Figure 4.2(b)), is at less 2% across all congestion variation settings. The default setting (1Mbytes-41Kbits) recorded high packet loss rate compared to the default setting in the original protocol. The default was our least congestion level settings which show that the algorithm performs well on congested links.

Table 4.2: Small scale packet loss results.
The table compares the packet loss results between BATMAN original and the situation-aware version across different congestion levels. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented.

Packet Loss							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
PC1	3.80	0.77	1.13	6.07	3.13	2.77	2.94
LP	6.54	6.01	5.53	4.30	4.90	5.28	5.42
PC2	1.33	1.12	1.26	0.35	0.41	0.51	0.83
Situation-aware BATMAN-adv							
PC1	7.92	3.21	1.73	2.02	2.25	4.03	3.53
LP	9.45	1.75	2.12	1.27	2.09	1.67	3.05
PC2	8.58	0.94	2.04	5.05	1.54	0.43	3.09

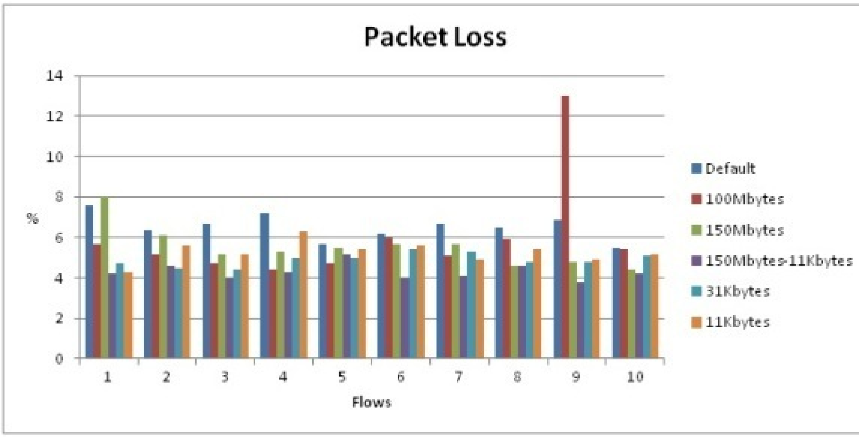
4.1.3 Throughput

Unlike packet loss, the consistency in PDV correlates well with the consistency in throughput as shown in Table 4.1 and Table 4.3. The average throughput is also independent of the variation settings. The average throughput in BATMAN-adv is consistently at 0.08 MB/s. On the other hand our situation-aware version tends to fluctuate a bit. The maximum recorded throughput in flow for BATMAN-adv is 0.09Mbytes/s while our situation-aware version could reach 3Mbytes/sec in a particular flow but due to its fluctuation tendency, the overall average amounts to 0.76Mbytes/sec. We observe that jitter/PDV and throughput are correlated, i.e. consistent PDV results in a consistent throughput. In terms of throughput, both protocol versions are at par with each.

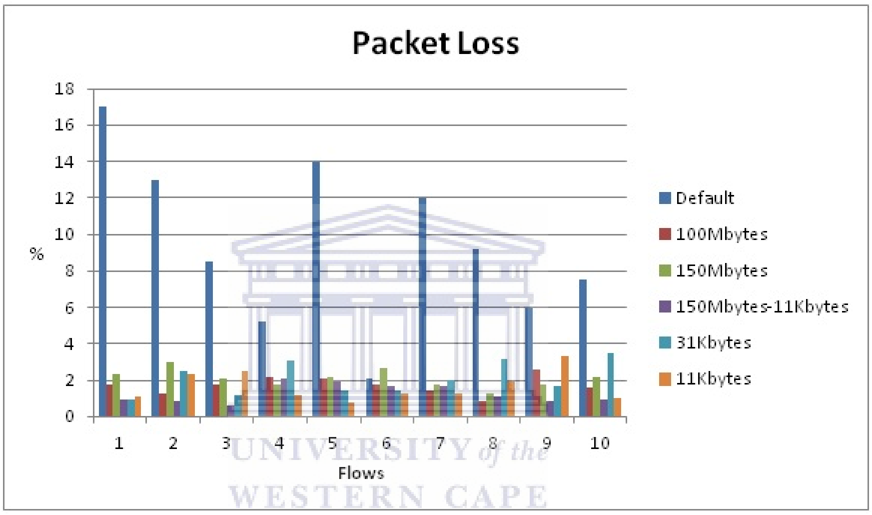
Table 4.3: Small scale throughput results.
The table compares the throughput results between BATMAN original and the situation-aware version across different congestion level. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented.

Throughput							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
PC1	0.08	0.09	0.08	0.09	0.09	0.08	0.085
LP	0.09	0.09	0.09	0.09	0.09	0.08	0.088
PC2	0.08	0.09	0.08	0.08	0.08	0.08	0.081
Situation-aware BATMAN-adv							
PC1	0.07	0.06	0.09	0.10	0.07	0.05	0.073
LP	0.11	0.05	0.05	0.09	0.06	0.06	0.07
PC2	0.06	0.09	0.08	0.09	0.09	0.09	0.083

Figure 4.3 illustrates the difference throughput between the two protocols graphically. 0.09Mbytes/s was the most recorded throughput in both protocols across all variation settings. In Figure 4.3(a) it can be seen that the maximum achieved throughput in a flow for that original protocol was 0.09Mbyte/s while the situation-aware protocol much



(a) Packet loss from the mobile node LP to the server for BATMAN-adv original.



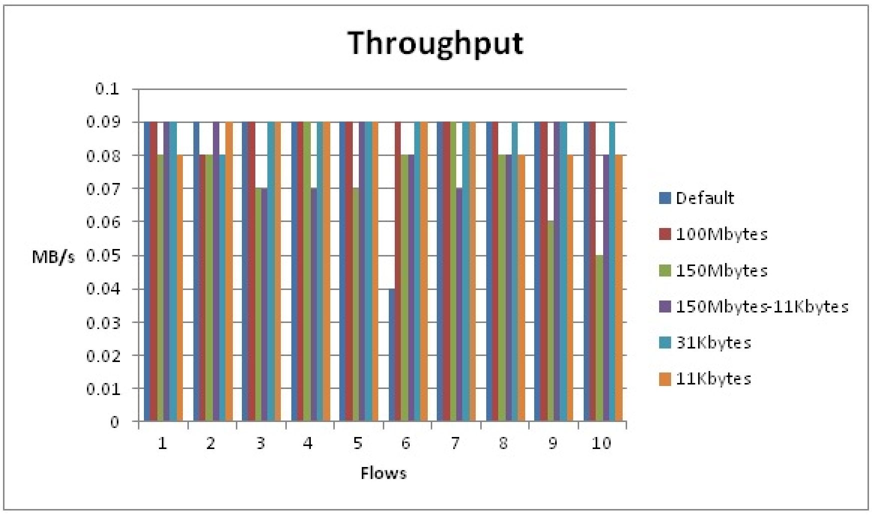
(b) Packet loss from the mobile node LP to the server for Situation-aware BATMAN-adv.

Figure 4.2: Compares packet loss results between the two protocols
The two graphs compares the packet loss rate between the BATMAN-adv original and the situation-aware version on the small scale network test bed for the mobile node. The packet loss rate for the situation-aware version reduces as the congestion level increases.

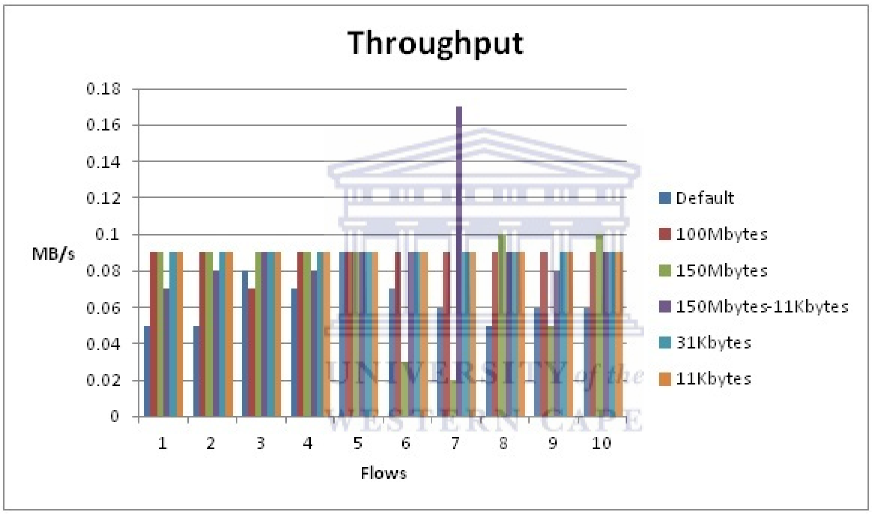
higher throughput was achieved. In flow 7 under 150Mbyte-11Kbits 0.17Mbyte/s was achieved and in flow 8 and 10 both under 150Mbyte 1Mbyte/s was reached.

4.2 Final Results: larger scale test bed

The experimental evaluations were conducted on an enlarged network test bed of 12 nodes. A network expansion is inevitable in a dynamic mesh network setup so it is critical to test the impact it has on performance. In addition to the test bed growth, two mobile nodes were used instead of only 1 which was used in the preliminary experiments



(a) Throughput from the node PC2 to the server for BATMAN-adv original.



(b) Throughput from the node PC2 to the server for Situation-aware BATMAN-adv.

Figure 4.3: Compares throughput results between the two protocols
The throughput results between the BATMAN-adv original and the situation-aware version on the small scale network test bed. The average throughput is almost the same with the original protocol slightly higher.

(see Figure 3.2 and 3.5). This was to strengthen the rigidity in mobility. The results show an enormous improvement compared to the preliminary tests.

4.2.1 Jitter

The overall jitter results show a great improvement compared to the small scale tests. In between the protocols, the situation-aware version performed much better as the average was at most 37.025ms (see Table 4.4) which was not the case in the small scale tests. The average of the averages across the three nodes highlights the overall behaviour of the two protocols under different network sizes. The original protocol recorded an average of 37.126ms in the small scale and 54.329ms in the larger scale. This shows no pragmatic improvement under scalability. However the modified protocol version equals to an average of 123.396ms in the small scale and 33.369ms in the larger scale which is a significant improvement. We can deduce that scalability does have a positive impact on jitter performance. It can also be confirmed from the jitter results, both small scale and larger scale, that congestion has little impact on jitter.

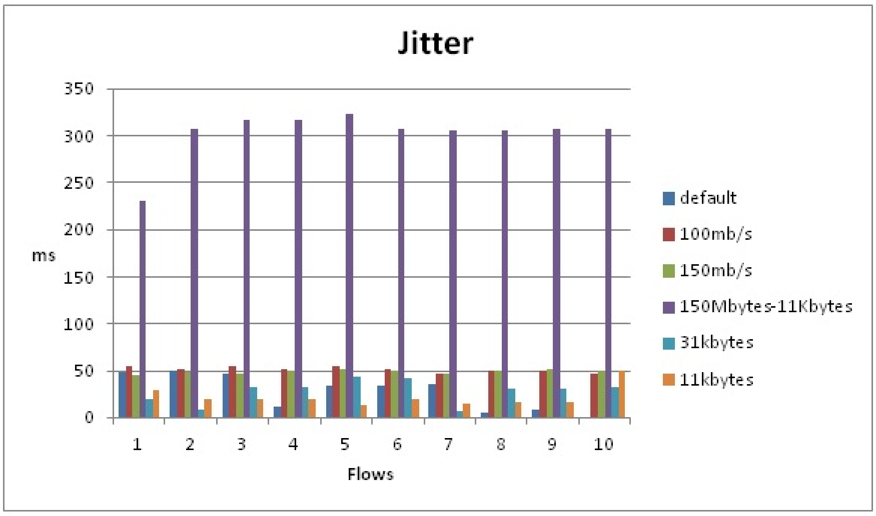
Table 4.4: Larger scale jitter results.
The table compares the packet loss results between BATMAN original and the situation-aware version across different congestion levels. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented.

Jitter							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
LP1	28.076	51.100	49.347	302.548	27.943	21.973	80.165
LP2	40.299	54.705	55.217	31.851	39.670	22.199	40.657
PC2	96.195	46.289	47.019	27.722	19.071	16.506	42.164
Situation-aware BATMAN-adv							
LP1	35.123	57.485	52.932	28.871	26.335	21.407	37.025
LP2	26.301	48.671	52.719	31.883	20.171	21.62	33.561
PC2	2.067	48.671	52.719	31.883	20.171	212.62	29.522

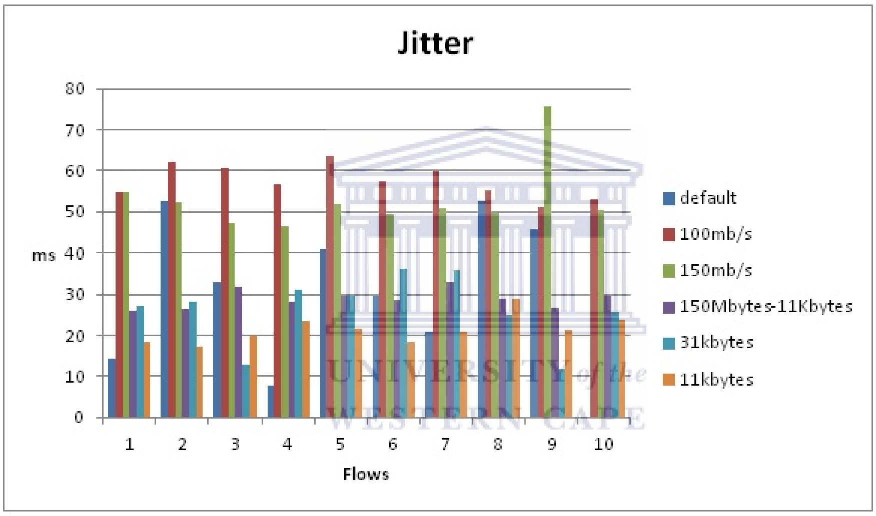
Figure 4.4 highlights the effects of congestion on jitter graphically, LP1 is used for illustration. In the original BATMAN-adv most of the jitter values consistently falls below 50ms except of 150Mbytes-11Kbits which also does not behave the same in the larger scale tests. Furthermore, jitter in the larger scale test bed is consistently below 30ms with the exception of 100Mbytes and 150Mbytes which are both on an average of 50ms. The common factor in this regard is the transfer rate and hence we can deduce that congestion has least effect on jitter.

4.2.2 Packet loss

In the larger-scale test bed, packet loss rate reduced significantly as compared to the small scale tests. The overall packet loss rate as seen in Table 4.5 is less than 1% in both protocols. In most cases only 1 packet was lost out of a possible 4000 sent by IPERF; voice and other data types can tolerate this loss level. The situation-aware protocol performed much better when comparing the two in the larger scale. The packet loss rate of situation-aware protocol is consistently below that of the original protocol across all congestion variation settings and both are at an acceptable level. The average loss rate on the mobile nodes is at 0.029% and 0.08% for LP1 and LP2 respectively. This is much better than 0.173% and 1.96% on the same nodes respectively for the



(a) Jitter from mobile node LP1 to the server for BATMAN-adv original.



(b) Jitter from mobile node LP1 to the server for Situation-aware BATMAN-adv.

Figure 4.4: Compares jitter results between the two protocols
The jitter results comparison between the BATMAN-adv original and the situation-aware version on the larger scale network test bed. The situation-aware version achieved the best (lowest) jitter compared to the original version; there is a huge improvement from the small scale where the original.

original protocol version. Another notable and significant factor is the performance of the stationery nodes in both protocols. The average packet loss rate in BATMAN-adv original is 0.189% while the situation-aware version is 0.158%. The slight difference results from the fact that they were in the same room and at a constant signal reach with the server node.

The average of the three node averages for the situation-aware protocol and the original protocol in the larger scale is 0.0911% and 0.769% respectively while the original is

Table 4.5: Larger scale packet loss results.
The table compares the packet loss results between BATMAN original and the situation-aware version across different congestion level. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented..

Packet Loss							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
LP1	0.173	0.201	0.121	0.183	0.224	0.139	0.173
LP2	0.765	1.93	1.39	3.3	3.47	0.822	1.946
PC2	0.006	0.446	0.254	0.204	0.100	0.126	0.189
Situation-aware BATMAN-adv							
LP1	0.022	0.027	0.027	0.018	0.042	0.040	0.029
LP2	0.141	0.029	0.045	0.110	0.041	0.145	0.085
PC2	0.002	0.021	0.079	0.163	0.276	0.223	0.158

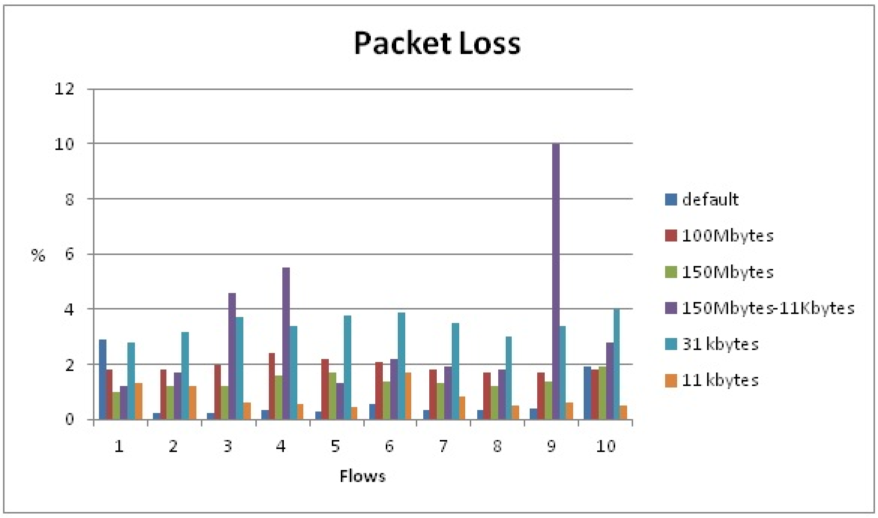
3.063% and 3.223% in the same respective order. This proves the significant improvement that scalability brought in both protocols. In Figure 4.5, an illustration from the mobile LP2 to the server is graphically presented. Figure 4.5(b) shows that the highest packet loss endured in the situation-aware protocol is less than 0.25% while more than 1% was encountered in the original with some points even reaching 10% at some stage. This also show the superiority of the situation-aware protocol as well as the effectiveness of situation-aware methods in combating high levels of packet loss in growing networks.

4.2.3 Throughput

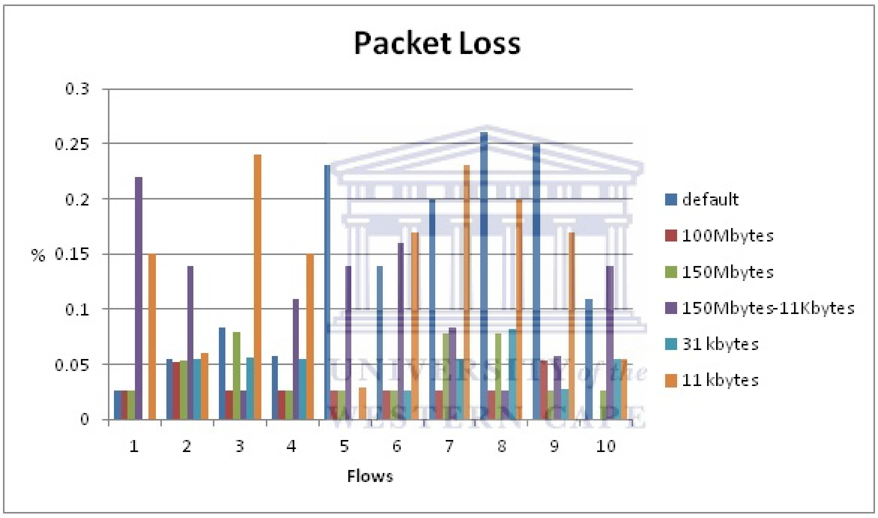
Throughput is one of the goals this study aimed to improve by using situation-aware methods. The throughput results are presented in Table 4.6. The situation-aware protocol achieved a much higher throughput to the original protocol, a huge improvement from the small scale tests.

Table 4.6: Larger scale throughput results.
The table compares the throughput results between BATMAN original and the situation-aware version across different congestion level. Each value presented is an average of 10 flows between two nodes and the average of the averages is also presented..

Throughput							
Node	**1Mbytes 41Kbits**	**100Mbytes 41Kbits**	**150Mbytes 41Kbits**	**150Mbytes 11Kbits**	**1Mbytes 31Kbits**	**1Mbytes 11Kbits**	**Average**
BATMAN-adv							
LP1	0.83	0.085	0.016	0.09	0.088	0.083	0.074
LP2	0.081	0.08	0.08	0.006	0.07	0.08	0.066
PC2	0.04	0.088	0.09	0.09	0.09	0.09	0.081
Situation-aware BATMAN-adv							
LP1	0.082	0.078	0.086	0.09	0.08	0.09	0.084
LP2	0.081	0.081	0.09	0.082	0.086	0.081	0.083
PC2	0.12	0.086	0.085	0.081	0.083	0.087	0.090



(a) Packet loss from the mobile node LP2 to the server for BATMAN-adv original.

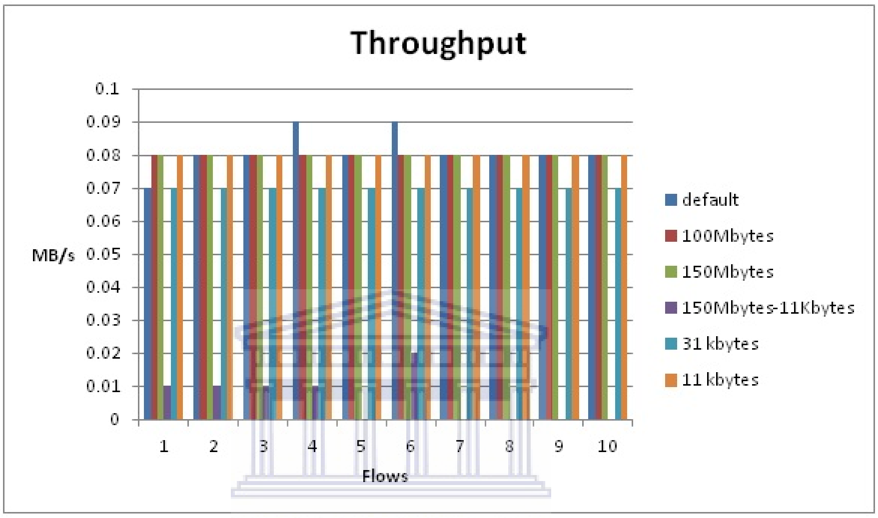


(b) Packet loss from the mobile node LP2 to the server for BATMAN-adv situation-aware.

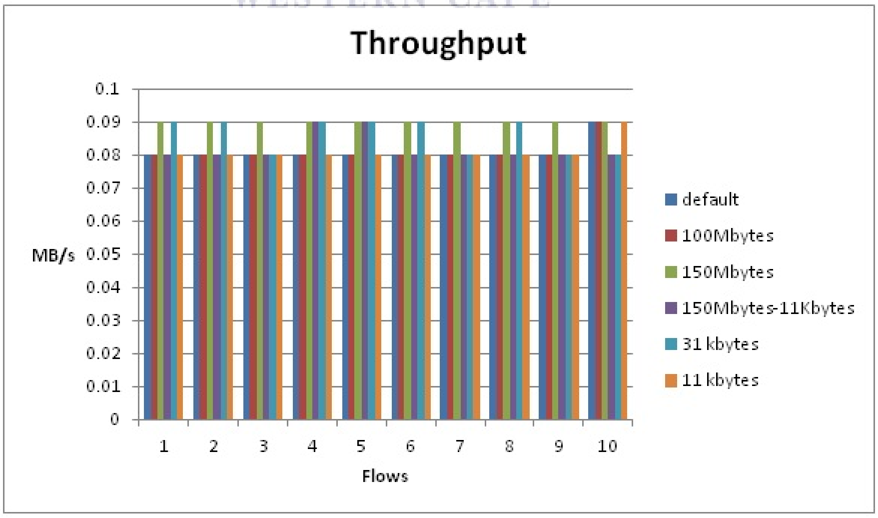
Figure 4.5: Compares packet loss results between the two protocols
The two graphs compares the packet loss rate between the BATMAN-adv original and the situation-aware version on the larger scale network test bed for the mobile node. The overall packet loss rate reduced immensely for both protocols compared to the small scale test bed. The situation-aware protocol achieved the lowest loss rate compared to the original protocol.

The two mobile nodes, LP1 and LP2 in BATMAN-adv original achieved an average of 0.0741 Mbytes/s and 0.0661 Mbytes/s respectively (see Table 4.6). This is far less as compared to the 0.0843 Mbytes/s and 0.0835 Mbytes/s of the same nodes respectively for the situation-aware version. Thus, the consistent and high throughput on the situation-aware protocol affirms the effectiveness and efficiency of situation-aware methods in dynamic situations. The size of the network, however, makes a huge difference as the average of the node's averages in the small scale test bed is 0.075 Mbytes/s for the situation-aware protocol compared to 0.084 Mbytes/s for the original protocol. In the larger scale tests, the average for the situation-aware protocol is 0.0861 Mbytes/s compared to 0.0738Mbytes/s of the original protocol. This proves the effectiveness of scalability in mesh network performance, thus the more nodes the better the performance.

Figure 4.6 graphically presents an illustrative example of this fact from mobile node LP2 to the server. In 4.6(b) all value points are more than 0.08Mbytes/s while in 4.6(a) only few reached 0.08Mbytes/s. This shows the level of consistency in the situation-aware protocol which brings a sense of reliability in a network.



(a) Throughput from the mobile node LP2 to the server for BATMAN-adv original.



(b) Throughput from the mobile node LP2 to the server for Situation-aware BATMAN-adv.

Figure 4.6: Compares throughput results between the two protocols
The throughput results between the BATMAN-adv original and the situation-aware version on the larger scale network test bed. The difference of the average throughput between the two protocols is not much; the situation-aware version achieved the best of the two.

4.2.4 Latency

Latency is regarded as a measure of delay because it measures the time a packet takes to reach a destination and it is different from Jitter which is the variation of delay. The Latency results were obtained by using the PING command. The measure of latency is unidirectional, i.e. only one way. Round Trip Time (RTT) on the other hand measures the time it takes for a packet to reach a destination and back. Table 4.7 provides PING statistics measured simultaneously from node LP1, LP2 and PC2 to the server. 5000 ping request were sent.

Table 4.7: Latency.
This table shows detailed latency results attained through PING.

Latency							
Node	**Transmit**	**Packet Loss (%)**	**Time (ms)**	**Round Trip Time (RTT)**			
				Min. Time	**Avg. Time**	**Max. Time**	**Std. Deviation**
BATMAN-adv							
LP1	5000	0	5005739	1.733	3.997	158.909	4.183
LP2	5000	23	5023120	1.545	3.954	140.473	3.216
PC2	5000	0	5005682	1.785	3.892	133.027	3.264
BATMAN-adv situation-aware							
LP1	5000	0	5005500	1.776	4.168	272.654	8.628
LP2	5000	0	5004661	2.726	4.892	232.238	8.125
PC2	5000	0	5005640	1.663	4.353	414.961	11.555

The latency results in Table 4.7 shows 100% packet delivery rate with the exception of LP2 on the original BATMAN-adv which suffered 23% packet loss. This packet loss results should not be confused with the results in Section 4.1.2 and 4.2.2 because this was solely based on control packets (ICMP) not real data packets. The latency times do not differ that much between the two protocols but the situation-aware version performed better. The RTT results show a better response in time for the original BATMAN-adv. The minimum time and average time also differ slightly in both protocols. The maximum time and the standard deviation of the situation-aware protocol are both almost double that of the original protocol. This is due to the fact that our algorithm has a dynamic way of routing based on the network state. Thus a route to a destination changes a lot faster than it does in the original. However the fact that the one-way transmission time in the situation-aware protocol is better than the original proves the effectiveness of the algorithm in terms of packet delivery speed.

4.3 Summary

The results were obtained from two experimental network test beds namely: small scale and larger scale. The small scale consisted of four nodes with one mobile node while the larger scale consisted of 12 nodes, 8 mesh potato nodes and 4 computer nodes composed of 2 mobile nodes. The small scale tests showed that BATMAN-adv original had an upper hand in terms of the overall jitter and packet loss while the situation-aware protocol lacked consistency in jitter. The packet loss rate from the situation-aware protocol however showed a responsive reaction to different congestion levels. The packet loss rate reduced as the links got congested. The overall packet loss for the two protocols in the small scale test bed was at an unacceptable level. Throughput in the small scale did not vary that much between the two protocols with the original protocol slightly high. The larger scale results showed a huge improvement in all performance metrics for both protocols. In terms of jitter, BATMAN-adv situation-aware performed better, a huge improvement compared to the small scale. The packet loss results in both protocols improved significantly from the small scale with the overall packet rate of less than 1%, which is acceptable and tolerable. There was a notable constant packet loss rate of about 0.025% which could possibly be resulting from external factors such as interference. The situation-aware protocol was superior in this instance.The overall throughput achieved in both protocols from the two test beds does not vary that much just like the two protocols in the respective test bed tests. The throughput also saw a significant improvement for the situation-aware protocol from its small scale performance.The latency was only measured in the larger scale tests. The situation-aware protocol achieved less delay compared to the original. The original protocol showed a better RTT. This is so because our algorithm allows a quick change of best path towards a destination.

Chapter 5

Conclusion

This chapter presents the conclusion of the study together with limitations of the study, recommendations and future work. Section 5.1 analyses the findings obtained from the results. Section 5.2 discusses the limitations of the experimental design. Section 5.3 makes recommendations based on lessons learned from this, study and finally, Section 5.4 suggests future work.

5.1 Analysis of the findings

Situation-aware routing has proven to improve QoS on dynamic wireless mesh networks by making routing decisions based on the current situation of the network. BATMAN-adv counts OGMs received as a link quality measurement. This study gives the more recently received OGMs more weight in deciding the link quality by summing their indices in a sliding window rather than counting OGM quantity like BATMAN does. Therefore, more recently received OGMs contribute more to the metric in order to give a more precise indication of the current state of a link. The results show little relation between jitter/PDV and packet loss. Jitter is, however, proportional to throughput. The original BATMAN-adv achieved the best jitter in the small scale experiment. The average throughput achieved on both protocols was almost the same. The modified protocol suffered from packet loss at low bandwidth rates but this reduced as the transfer rate increased and buffer size shrunk, i.e. it performed well with congestion.

The overall results in the larger scale network test bed improved markedly from the initial small scale network test bed. The jitter results for the situation-aware protocol improved more than the original. The improvements in jitter performance have shown the flexibility of the situation-aware algorithm in dealing with scalability. The results from the larger test bed have shown that congestion has little impact on jitter. The impact of mobility on jitter is also unnoticeable on both protocols from both test beds. The jitter results suggest that more nodes in a network results in easy propagation of packets. The best jitter achieved by the modified protocol showed how well situation-aware methods deal with scalability towards reducing jitter.

The rate of packet loss reduced significantly in the larger test bed. The rate of packet loss encountered by both protocols was at an acceptable and tolerable level because many data streams such as voice and video are least affected by such a low level loss. Both protocols proved to be scalable but the situation-aware protocol was superior. A certain portion of the low packet loss could be a result of factors such as interference. The packet loss results in the larger scale showed no response to congestion. It can be concluded that situation-aware techniques thus reduce packet loss rate in a dynamic network with mobile nodes.

The larger test bed throughput did not vary much from the small test bed results. The original protocol endured a decline in throughput while the situation-aware protocol improved in the larger test bed. This proved the effectiveness of the algorithm and further showing that it can indeed improve network performance under dynamic situations. The two mobile nodes running the situation-aware protocol achieved almost the same throughput but the static nodes had a slightly higher throughput. This has showed that mobility can indeed affect the stability of the link quality in general.

In terms of latency, the situation-aware protocol showed less delay compared to the original protocol. The original however achieved a better RTT than the modified protocol. This is caused by the nature of the algorithm which is based on the current network state and therefore the best outgoing path towards a destination changes fast. Since one-way delay is better, it means the suffered hitch was in the transition of best links.

We can conclude that the situation-aware algorithm performs well under both mobility and moderate scalability. The impact of congestion was only experienced in the small test bed and this is where the situation-aware algorithm performed well. Since the congestion settings were the same in both test beds, the small impact shows that scalability affected it. The congestion lessened because there were more routes to choose from. We have demonstrated that situation-aware routing offers great potential to address issues pertaining to mobility, congestion and scalability in dynamic mesh networks. We can also say with confidence that the number of nodes in any mesh network is directly proportional to its performance.

5.2 Limitations of the study

There were some limitations in both experimental design setups, which consequently had a negative impact on the results. The preliminary test bed was in a single computer laboratory room with only four nodes. The distance between the nodes was short. Also, there were several other wireless networks in that same room. Although we tried our best to confine our network to a free channel spread, the noted persistent packet loss could possibly result from network interference. Despite having a larger test bed, ideally as close to our target NGO headquarters' setup, it was still not enough. The challenges in this case were the concrete walls dividing the building's rooms where we distributed the mesh potato devices. The walls interfered with the network signal, detracting the performance. The results are thus limited to our 'real world' test bed. The experiments could only scale up to 12 nodes which might not necessarily be the case in real networks like the rural NGO headquarters. The two experiments also had the same congestion enhancement settings which proved futile in the larger experiment. We could not test using mobile phones because BATMAN-adv is a Layer 2 protocol and does not yet run on mobile phones. This explains why laptops were used, running BATMAN-adv.

5.3 Recommendations

Mesh networks enable 'bottom up' networks to be built quickly and inexpensively [66], and they *can* be independent of telecoms providers like Telkom or Vodacom. A mesh potato network provides a seamless model of communication and information access in communities. This model works such that local calls can be made for free and people only pay for breakout calls, which is how telcos can benefit with interconnection. A village telco also needs a gateway for Internet access. A typical operator's business model tends to cater for wealthy customers thereby marginalizing the poor. It is highly recommended that these big communication giants seize the opportunity of being part of this revolutionary development. This presents an opportunity for them to connect their existing infrastructure to bottom-up small enterprise networks in an 'over-the-top' fashion. This will mutually benefit both parties as the companies can charge for Internet provision and interconnect voice calls.

On a technical level, it is important to be familiar with kernel level debugging in order to implement optimization algorithms such as this. The BATMAN-adv algorithm depends a lot on numeric values and for that one needs to always validate by printing them out. For C developers use `printk` (print kernel). This will print the routing decision making values in the kernel. This can be displayed by using `dmesg` | less command. All these values will appear on the wireless interface used.

5.4 Future work

In the future, we would like to see how the protocol performs under a wider range of traffic patterns, and also in a more geographically spread mesh network, with more nodes. Our algorithm experiences retardation when dealing with round trip requests. We would like to use bit shifting instead of summing the indices of the sliding window. Bit shifting is faster because it deals directly with the binary numbers which are faster to compute than decimal indices. Our initial plan was to use mobile phones as the mobile nodes but could not because BATMAN-adv has not yet been ported to the mobile phone. Hence, any future work can orient toward that scenario. Besides, this situation-aware algorithm described herein can easily be ported to standard BATMAN to be used in BATphone scenario.

Bibliography

[1] M. Abolhasan, B. Hagelstein, and J. C. Wang, “Real-world performance of current proactive multi-hop mesh protocols,” in *Proc. 15th Asia-Pacific Conference on Communications, APCC’09.* Piscataway, NJ, USA: IEEE Press, 2009, pp. 42–45.

[2] P. Acharya, A. Sharma, E. Belding, K. Almeroth, and K. Papagiannaki, “Congestion-aware rate adaptation in wireless networks: A measurement-driven approach,” in *Proc. of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON’08.* IEEE Press, 2008, pp. 1–9.

[3] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A multi-radio unification protocol for ieee 802.11 wireless networks,” in *Proc. First International Conference on Broadband Networks, BROADNETS ’04.* Washington, DC, USA: IEEE Computer Society, 2004, pp. 344–354.

[4] D. Aguayo, J. Bicket, and R. Morris, “Srcrr: A high throughput routing protocol for 802.11 mesh networks (draft),” *ACM Whitepaper*, 2003.

[5] J. Bertrand and J. Fransoo, “Operations management research methodologies using quantitative modeling,” *International Journal of Operations and Production Management*, vol. 22, no. 2, pp. 241–264, 2002.

[6] B. Boehm and W. Hansen, “The spiral model as a tool for evolutionary acquisition,” *CrossTalk*, vol. 14, no. 5, pp. 4–11, 2001.

[7] B. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[8] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, “Advances in network simulation,” *Computer*, vol. 33, pp. 59–67, 2000.

[9] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall, “The case for technology in developing regions,” *Computer*, vol. 38, no. 6, pp. 25–38, 2005.

[10] M. Caleffi and L. Paura, “Bio-inspired link quality estimation for wireless mesh networks,” in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks and Workshops, WoWMoM’09.* IEEE, 2009, pp. 1–6.

[11] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Saade, and M. Rubinstein, “Routing metrics and protocols for wireless mesh networks,” *IEEE Network*, vol. 22, no. 1, pp. 6–12, 2008.

[12] P. M. Carthy and D. Grigoras, “Multipath associativity based routing,” in *Proc. Second Annual Conference on Wireless On-demand Network Systems and Services.* Washington, DC, USA: IEEE Computer Society, 2005, pp. 60–69.

[13] D. Cavin, Y. Sasson, and A. Schiper, “On the accuracy of manet simulators,” in *Proc. second ACM international workshop on Principles of mobile computing.* ACM, 2002, pp. 38–43.

[14] T. Chen and M. Gerla, “Global state routing: A new routing scheme for ad-hoc wireless networks,” in *Proc. 1998 IEEE International Conference on Communications, Conference Record, ICC’98*, ser. ICC’98, vol. 1. IEEE, 1998, pp. 171–175.

[15] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, “Analysis of dynamic source routing and destination-sequenced distance-vector protocols for different mobility models,” in *Proc. First Asia International Conference on Modelling and Simulation.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 224–229.

[16] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *Proc. 10th annual international conference on Mobile computing and networking.* ACM, 2004, pp. 114–128.

[17] P. Esposito, M. Campista, I. Moraes, L. Costa, O. Duarte, and M. Rubinstein, “Implementing the expected transmission time metric for olsr wireless mesh networks,” in *Wireless Days, 2008. WD’08. 1st IFIP.* IEEE, 2008, pp. 1–5.

[18] A. Gupta, H. Sadawarti, and A. Verma, “Performance analysis of aodv, dsr and tora routing protocols,” *IACSIT International Journal of Engineering and Technology*, vol. 2, no. 2, pp. 226–231, 2010.

[19] Z. Haas, “A new routing protocol for the reconfigurable wireless networks,” in *Proc. 6th International Conference on Universal Personal Communications, ICUPC’97*, vol. 2. IEEE, 1997, pp. 562–566.

[20] Z. Haas and M. Pearlman, “The performance of query control schemes for the zone routing protocol,” *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 4, pp. 427–438, 2001.

[21] F. Haq and T. Kunz, “Simulation vs. emulation: Evaluating mobile ad hoc network routing protocols,” in *Proc. International Workshop on Wireless Ad-hoc Networks, IWWAN’05*, 2005.

[22] R. Harris, “Link state routing,” *Computer Networks*, vol. 1, p. 2, 2011.

[23] C. Hsu and U. Kremer, “Iperf: A framework for automatic construction of performance prediction models,” in *Proc. Workshop on Profile and Feedback-Directed Compilation (PFDC), PFDC’98, Paris, France*, 1998.

[24] Y. Hu, D. Johnson, and A. Perrig, “Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks,” *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.

[25] L. Iannone and S. Fdida, “Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks,” in *Proc. IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN)*, 2005, pp. 103–110.

[26] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in *Proc. IEEE International In Multi Topic Conference, Technology for the 21st Century, INMIC’01, IEEE.* IEEE, 2001, pp. 62–68.

[27] W. Jiang, S. Liu, and Y. Zhu, “A high-throughput routing metric for large-scale multi-radio mesh networks,” in *Proc. Wireless Communications and Networking Conference, WCNC’07, IEEE*, 2007, pp. 3598–3602.

[28] M. Joa-Ng and I. Lu, “A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks,” *IEEE Journal on Selected Areas in Communications,*, vol. 17, no. 8, pp. 1415–1425, 1999.

[29] D. Johnson and G. Hancke, “Comparison of two routing metrics in olsr on a grid based mesh network,” *Ad Hoc Networks*, vol. 7, no. 2, pp. 374–387, 2009.

[30] D. Johnson, N. Ntlatlapa, and C. Aichele, “Simple pragmatic approach to mesh routing using batman,” in *Proc. 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, 2008.

[31] D. Johnson, D. Maltz, J. Broch *et al.*, “Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks,” *Ad hoc networking*, vol. 5, pp. 139–172, 2001.

[32] M. Kaosar, M. Hafiz, M. Tarek, R. Sheltami, and A. Mahmoud, "Simulation-based comparative study of on demand routing protocols for manet," in *Proc. International Conference on Wireless Networking and Mobile Computing, ICWNMC05, Chennai, India*, vol. 1, 2005, pp. 201–206.

[33] B. Karp and H. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proc. 6th annual international conference on Mobile computing and Networking.* ACM, 2000, pp. 243–254.

[34] M. Khalil, "The wireless internet opportunity for developing countries," *I-WAYS-The Journal of E-Government Policy and Regulation*, vol. 27, no. 1, pp. 17–20, 2004.

[35] P. Khatri, M. Rajput, A. Shastri, and K. Solanki, "Performance study of ad-hoc reactive routing protocols," *Journal of Computer Science*, vol. 6, no. 10, pp. 1130–1135, 2010.

[36] S. Kumar and S. Ramachandram, "The performance evaluation of genetic zone routing protocol for manets," in *Proc. TENCON 2008 IEEE Region 10 Conference.* IEEE, 2008, pp. 1–6.

[37] T. Liu and W. Liao, "On routing in multichannel wireless mesh networks: challenges and solutions," *Network, IEEE*, vol. 22, no. 1, pp. 13–18, 2008.

[38] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. Network Protocols Ninth International Conference on ICNP 2001.* IEEE, 2001, pp. 14–23.

[39] ——, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[40] M. Miles and A. Huberman, *Qualitative data analysis.* Sage Publ., 1999.

[41] P. Misra, "Routing protocols for ad hoc mobile wireless networks," Deparment of Computer Science, Washington University at St.Louis, 1999.

[42] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle, "A link-state qos routing protocol for ad hoc networks," in *Proc. 4th International Workshop on Mobile and Wireless Communications Network*, IEEE. IEEE, 2002, pp. 222–226.

[43] C. Murphy, "Moving averages." [Online]. Available: http://investopedia.com/inv/pdf/tutorials/MovingAverages.pdf

[44] D. Murray, M. Dixon, and T. Koziniec, "An experimental comparison of routing protocols in multi hop ad hoc networks," in *Australasian Telecommunication Networks and Applications Conference, ATNAC*, 2010.

[45] A. Neumann, C. Aichele, and M. Lindner, “Lindner: Batman status report, 2007.”

[46] N. Nikaein, C. Bonnet, and N. Nikaein, “Harp-hybrid ad hoc routing protocol,” in *Proc. International Symposium on Telecommunications, IST’01*, vol. 1, no. 3, September 2001.

[47] U. of Luxembourg SECAN-Lab, “Interoperability laboratory for security in ad-hoc networks.” [Online]. Available: http://wiki.uni.lu/secan-lab/

[48] M. Olivier, *Information technology research: a practical guide for computer science and informatics.* Van Schaik, 2009.

[49] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *Proc. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, INFOCOM ’97*, vol. 3. Washington, DC, USA: IEEE Computer Society, 1997, pp. 1405–1413.

[50] G. Pei, M. Gerla, X. Hong, and C. Chiang, “A wireless hierarchical routing protocol with group mobility,” in *Proc. Wireless Communications and Networking Conference, WCNC’99, IEEE*, vol. 3. IEEE, 1999, pp. 1538–1542.

[51] G. Pei, M. Gerla, and T. Chen, “Fisheye state routing: A routing scheme for ad hoc wireless networks,” in *Proc. IEEE International Conference on Communications, ICC’00*, vol. 1. IEEE, 2000, pp. 70–74.

[52] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *Proc. Conference on Communications Architectures, Protocols and Applications, SIGCOMM ’94.* New York, NY, USA: ACM, 1994, pp. 234–244.

[53] R. Pinto, “Wmm-wireless mesh monitoring,” Technical report, INESC-ID, Tech. Rep., 2009.

[54] R. Pressman, *Software engineering: a practitioner’s approach*, 7th ed. McGraw-Hill Higher Education, 2010.

[55] V. Ramasubramanian, Z. Haas, and E. Sirer, “Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks,” in *Proc. 4th ACM international symposium on Mobile ad hoc networking & computing.* ACM, 2003, pp. 303–314.

[56] E. Royer and C. Perkins, “An implementation study of the aodv routing protocol,” in *Proc. Wireless Communications and Networking Conference, WCNC’00, IEEE*, vol. 3. IEEE, 2000, pp. 1003–1008.

[57] E. Royer and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE*, vol. 6, no. 2, pp. 46–55, 1999.

[58] R. Scheaffer, "Statistics education: Perusing the past, embracing the present, and charting the future," *Newsletter for the section on statistical education*, vol. 7, no. 1, 2001.

[59] S. Shaar, A. Fawaz, A. Murad, M. Ayman, R. Al-Shalabi, and G. Kanaan, "Analysis of enhanced associativity based routing protocol," *Journal of Computer Science*, vol. 2, no. 12, pp. 853–858, 2006.

[60] C. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," in *Proc. 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications.* IEEE, 1996, pp. 480–486.

[61] A. Tsirigos and Z. Haas, "Multipath routing in the presence of frequent topological changes," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 132–138, 2001.

[62] M. Vainio, "Link reversal routing," *Nokia Group*, 2002.

[63] Village-Telco, "Village telco." [Online]. Available: http://www.villagetelco.org/

[64] H. Vo and C. Hong, "Hop-count based congestion-aware multi-path routing in wireless mesh network," in *Proc. International Conference on Information Networking, ICOIN'08.* IEEE, 2008, pp. 1–5.

[65] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations," *Multimedia tools and Applications*, vol. 29, no. 3, pp. 285–303, 2006.

[66] R. Westerveld, "Inverse telecommunications: The future for rural areas in developing countries?" in *Inverse Infrastructures*, T. Egyedi and D. Mehos, Eds. Edward Elgar Publishing Ltd, 2011.

[67] S. Wunderlich, M. Lindner, and W. Tsai, "Batman advanced," 2011. [Online]. Available: http://www.open-mesh.org/wiki/batman-adv

[68] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *Proc. IEEE Workshop on Wireless Mesh Networks, WiMesh'05*, 2005.

[69] H. Yu, D. Wu, and P. Mohapatra, "Experimental anatomy of packet losses in wireless mesh networks," in *Proc. 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON'09.* Piscataway, NJ, USA: IEEE Press, 2009, pp. 126–134.

Appendix A

SATNAC 2010 work in progress paper (proposal)

HI Kobo and WD Tucker (2010). Quality of Service-aware Routing for Static Mesh Networks with Mobile Nodes. Proc. South African Telecommunications Networks & Applications Conference, (SATNAC 2010), Stellenbosch, South Africa.

Quality of Service-aware Routing for Static Mesh Networks with Mobile Nodes

Hlabishi I. Kobo and William D. Tucker
Department of Computer Science
University of the Western Cape, Private Bag X17, Bellville 7535 South Africa
Tel: +27 21 9592461, Fax: +27 21 9593006
E-mail: {**2654952**, btucker}@uwc.ac.za

Abstract- This paper describes work in progress to explore a quality of service based routing protocol suitable for a hybrid wireless mesh network. A hybrid network has both static and mobile wireless mesh routers. The combination of static and mobile routers can increase the reach and redundancy of an ad hoc network. Such networks ideally suite a rural environment where constant management and maintenance is unaffordedable. The network would therefore be extremely dynamic and require optimization of the routing protocol to adapt to frequent topological changes. The latest smart mobile phones, such as those running Android, can act as routers to support mesh protocols. We believe the use of these high end mobile handsets amongst the rural populace may become common place on village telco-type networks. However, adding mobile phones to a mesh network complicates the link structure and the stability of the network. The routing protocol should therefore know the topological situation, and the quality of its links, before making a routing decision. We will use statistical methods to monitor the stability of links and use the media access control layer to measure link signal strength to compute a situation-aware next-hop.

Index Terms TCP/IP & Layer 3 Protocols, Mobile/wireless protocols, Wireless mesh networks, Quality of service, Situation-aware routing.

I. INTRODUCTION

Quality-aware routing in mesh networks is about making routing decisions based on link quality. This paper discusses work in progress to study quality-aware routing on static wireless mesh networks with mobile nodes for quality of service (QoS) optimization. We call such a network a hybrid mesh network, as shown in Fig. 1.

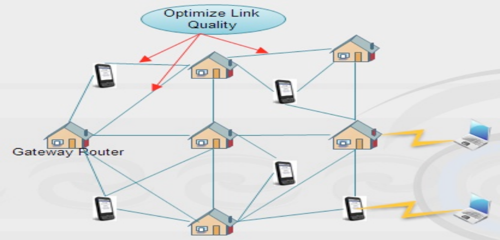


Fig. 1: A hybrid mesh network, applicable to a rural area, with static mesh routers inside homes, and mobile routers on cell phones. The goal is to optimize link quality within such dynamic topologies.

Wireless mesh networks (WMNs) have recently gained much popularity, as well as research attention, due to their inexpensive deployment and interesting characteristics. A wireless mesh network is a distributed network that can self-discover and self-heal [1]. WMNs offer an ideal infrastructure for providing affordable wireless Internet access to the less privileged. Such end-users still rely on 'push' type information such as newspapers, television news and radio. In our view, all people should have access to on-demand information such as the Internet, and mesh networks offer an affordable way to provide that access.

Routing protocols are at the centre of ongoing research on WMNs. This paper describes an aim to optimize BATMAN, a mesh routing protocol, to be feasible for a hybrid mesh network. Due to the participation of mobile nodes/routers, the protocol should be able to adapt to rapid topological changes. Situation-aware methods are proposed to improve the routing decisions in the hybrid mesh network to optimize the link quality for better throughput. BATMAN's routing algorithm checks for the existence of a link and increases the probability of delivering a packet through that link [7].

We propose two methods to optimize BATMAN's routing algorithm: statistical situation monitoring and MAC layer situation monitoring. The first uses standard deviation and average calculations to estimate the stability of a link, and the second method queries the MAC layer of the WMN to check the strength and congestion rate of a link. The investigation will test a situation-aware modification of BATMAN through simulation as well as on an actual network. CPU cycles will be taken into consideration for the examination of the computation complexity.

The rest of this paper is organized as follows: Section II previews the related literature, Section III discusses the methods and Section IV highlights conclusions and future work.

II. RELATED WORK

Routing is a process of delivering data packets from a source (sender) node to destination (receiver) node on a network. Routing protocols deal with the maintenance, creation, establishment and discovery of such routes [2]. Routing protocols are based on three protocol classification categories: proactive, reactive and hybrid. In proactive protocols, each node in the network maintains a table containing routing information of the entire network, which is updated periodically. Reactive protocols, also referred to as on-demand protocols, create a route from source to destination only when needed, e.g. when there is data to be sent. Hybrid protocols exhibit behavioural design aspects of both approaches.

Ad-hoc on demand distance vector (AODV) is one of the popular reactive protocols and hence creates routes on demand. AODV is a single path routing that is based on hop-by-hop routing [3]. Ad-hoc on-demand multipath distance vector (AOMDV) routing is based on AODV. Unlike AODV, AOMDV utilizes multipath routing alleviating link failures and link breakage suffered in AODV [3][4]. MeshDv is a hybrid protocol that uses a combination of proactive route computation for the routers and on-demand path request for clients [2] [5]. Optimized link state routing (OLSR) protocol is a proactive protocol that is based on a link state algorithm and uses hop-by-hop routing [6]. OLSR's objective is to reduce the size of control packets as well as the overhead cost by broadcasting control packets. Multipoint relays (MPR) are key to OLSR [7], and are subsets of the neighbours that a node uses to forward broadcast messages. A 'better approach to mobile ad-hoc networks' (BATMAN) is another proactive protocol that only maintains information about the best next-hop towards the destination [1][2] and thus reduces signal overhead. The objective of this protocol is to enhance the probability of delivering a packet [8]. The protocol maintains information about the existence of a node and thus does not check the quality of the packet [8]. BATMAN protocol stack has been successfully ported on an Android platform by the village-telco team (www.villageteco.org).

Routing protocols use metrics to decide how to select the best path to the next hop. Hop count routing metrics count the number of hops between the sender and the destination. This metric is simple to compute but does not consider packet loss or bandwidth. Expected transmission count (ETX) is a quality-aware metric that considers the number of MAC layer transmissions needed to successfully deliver a packet through a link [9][10]. An expected transmission time (ETT) metric was developed to overcome the shortcomings of ETX and hence it is an optimization of ETX [9]. Weighted cumulative ETT was designed to overcome the shortcomings of both ETX and ETT in order to reduce interference [9].

III. METHODS

We now turn attention solely on how to optimize BATMAN based on the related work. BATMAN uses control packets called originator messages (OGMs) in routing decisions. The proposed methods adopt the same criterion with some added QoS-oriented features. Each node in BATMAN periodically broadcasts OGMs to its neighbours who further rebroadcast the packets. The best link is measured by the highest number of OGMs received from the destination over a current sliding window. Given the mobility of mobile nodes, rapid topological changes to the hybrid mesh network are inevitable and thus the ideal approach is to take the current network situation into consideration before making routing decisions.

The first proposed method uses statistical methods to ensure the stability and reliability of the link. By computing statistical standard deviation of the number of OGMS recorded in the current sliding window, the variability of the link will be evaluated. Depending on the network's behavioural data, we will consider the top ranked links by BATMAN and apply statistical quality to check on them.

The second proposed method uses OSI Layer 2 information to estimate the signal strength and the congestion rate of the links. The Received Signal Strength Indicator (RSSI) obtained from the MAC layer will be used in this regard. The signal-to-noise ratio (SNR) will also be used to check the quantity of signal affected by noise. Again the top ranked links by BATMAN will be considered together with RSSI and SNR in to make routing decisions.

IV. CONCLUSIONS AND FUTURE WORK

According to the literature reviewed, BATMAN outperforms many other WMN protocols. However BATMAN does not check the quality of the link to make routing decisions. Considering the rapid topological changes in hybrid WMNs with mobile routers, quality-aware methods based on the network situation at any given time were proposed. These methods look at link stability and reliability to optimize routing decisions.

REFERENCES

[1] M. Abolhasan, B. Hagelstein, and J.C.P. Wang, "Real-world performance of current proactive multi-hop mesh protocols," *Faculty of Informatics-Papers*, 2009, p. 736.

[2] R. Pinto, "WMM-Wireless Mesh Monitoring", *Technical report, INESC-ID*, 2009.

[3] M. Marina and S. Das, "On-Demand Multi Path Distance Vector Routing in Ad Hoc Networks.", *in Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2001, pp. 14-23.

[4] E. Royer and C. Perkins " An Implementation Study of the AODV Routing Protocol.", *In Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 3, 2000, pp. 1003-1008.

[5] L. Iannone and S. Fdida, "Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks," *IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN)*, 2005, pp. 103–110.

[6] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, " Optimized link state routing protocol for ad hoc networks." *in Proceeding of IEEE International In Multi Topic Conference*, 2001, pp. 62-68.

[7] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle, "A link-state QoS routing protocol for ad hoc networks," *IEEE Conference on Mobile and Wireless Communications Networks-MWCN*, 2002.

[8] D. Johnson, N. Ntlatlapa, and C. Aichele, "A simple pragmatic approach to mesh routing using BATMAN," *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, 2008.

[9] Y. Yang, J. Wang and R. Kravets "Designing Routing Metrics for Mesh Networks." *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh). IEEE Press*, 2005

[10] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations," *Multimedia Tools and Applications*, vol. 29, 2006, pp. 285-303.

Hlabishi I. Kobo obtained a BSc Honours degree in Computer Science from the University of the Western Cape (UWC) in 2009. He is presently studying towards an MSc degree with the Bridging Applications and Networks Group (BANG) at the same institution. His main research interest is now wireless mesh routing protocols on mobile phones.

William D. Tucker is a Senior Lecturer of Computer Science at UWC and leads BANG research there. His research interest includes Internet Protocol networks and their applications in developing regions.

Appendix B

SATNAC 2011 full paper (initial results)

HI Kobo, WD Tucker and XM Liu (2011). Situation-aware Routing Based on Link Quality for Static Mesh Networks with Mobile Nodes. Proc. South African Telecommunications Networks & Applications Conference, (SATNAC 2011), East London, South Africa, 299-304.

Quality of Service-aware Routing for Static Mesh Networks with Mobile Nodes

Hlabishi I. Kobo and William D. Tucker
Department of Computer Science
University of the Western Cape, Private Bag X17, Bellville 7535 South Africa
Tel: +27 21 9592461, Fax: +27 21 9593006
E-mail: {**2654952**, btucker}@uwc.ac.za

Abstract- This paper describes work in progress to explore a quality of service based routing protocol suitable for a hybrid wireless mesh network. A hybrid network has both static and mobile wireless mesh routers. The combination of static and mobile routers can increase the reach and redundancy of an ad hoc network. Such networks ideally suite a rural environment where constant management and maintenance is unaffordedable. The network would therefore be extremely dynamic and require optimization of the routing protocol to adapt to frequent topological changes. The latest smart mobile phones, such as those running Android, can act as routers to support mesh protocols. We believe the use of these high end mobile handsets amongst the rural populace may become common place on village telco-type networks. However, adding mobile phones to a mesh network complicates the link structure and the stability of the network. The routing protocol should therefore know the topological situation, and the quality of its links, before making a routing decision. We will use statistical methods to monitor the stability of links and use the media access control layer to measure link signal strength to compute a situation-aware next-hop.

Index Terms TCP/IP & Layer 3 Protocols, Mobile/wireless protocols, Wireless mesh networks, Quality of service, Situation-aware routing.

I. INTRODUCTION

Quality-aware routing in mesh networks is about making routing decisions based on link quality. This paper discusses work in progress to study quality-aware routing on static wireless mesh networks with mobile nodes for quality of service (QoS) optimization. We call such a network a hybrid mesh network, as shown in Fig. 1.

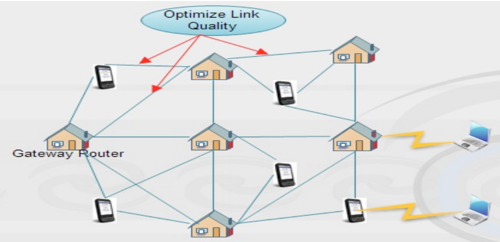


Fig. 1: A hybrid mesh network, applicable to a rural area, with static mesh routers inside homes, and mobile routers on cell phones. The goal is to optimize link quality within such dynamic topologies.

Wireless mesh networks (WMNs) have recently gained much popularity, as well as research attention, due to their inexpensive deployment and interesting characteristics. A wireless mesh network is a distributed network that can self-discover and self-heal [1]. WMNs offer an ideal infrastructure for providing affordable wireless Internet access to the less privileged. Such end-users still rely on 'push' type information such as newspapers, television news and radio. In our view, all people should have access to on-demand information such as the Internet, and mesh networks offer an affordable way to provide that access.

Routing protocols are at the centre of ongoing research on WMNs. This paper describes an aim to optimize BATMAN, a mesh routing protocol, to be feasible for a hybrid mesh network. Due to the participation of mobile nodes/routers, the protocol should be able to adapt to rapid topological changes. Situation-aware methods are proposed to improve the routing decisions in the hybrid mesh network to optimize the link quality for better throughput. BATMAN's routing algorithm checks for the existence of a link and increases the probability of delivering a packet through that link [7].

We propose two methods to optimize BATMAN's routing algorithm: statistical situation monitoring and MAC layer situation monitoring. The first uses standard deviation and average calculations to estimate the stability of a link, and the second method queries the MAC layer of the WMN to check the strength and congestion rate of a link. The investigation will test a situation-aware modification of BATMAN through simulation as well as on an actual network. CPU cycles will be taken into consideration for the examination of the computation complexity.

The rest of this paper is organized as follows: Section II previews the related literature, Section III discusses the methods and Section IV highlights conclusions and future work.

II. RELATED WORK

Routing is a process of delivering data packets from a source (sender) node to destination (receiver) node on a network. Routing protocols deal with the maintenance, creation, establishment and discovery of such routes [2]. Routing protocols are based on three protocol classification categories: proactive, reactive and hybrid. In proactive protocols, each node in the network maintains a table containing routing information of the entire network, which is updated periodically. Reactive protocols, also referred to as on-demand protocols, create a route from source to destination only when needed, e.g. when there is data to be sent. Hybrid protocols exhibit behavioural design aspects of both approaches.

Ad-hoc on demand distance vector (AODV) is one of the popular reactive protocols and hence creates routes on demand. AODV is a single path routing that is based on hop-by-hop routing [3]. Ad-hoc on-demand multipath distance vector (AOMDV) routing is based on AODV. Unlike AODV, AOMDV utilizes multipath routing alleviating link failures and link breakage suffered in AODV [3][4]. MeshDv is a hybrid protocol that uses a combination of proactive route computation for the routers and on-a particular moment in time so we give more recently received packets more weight in the routing decision. We tested the protocol enhancement on a small WMN with some mobility. The results are encouraging, and not too far off the mark from the original BATMAN.

The rest of the paper is organized as follows: Section II reviews some related work. Section III presents the methods for the protocol enhancement and the experimental setting. Section IV presents preliminary results, and Section V discusses them. Section VI concludes the paper and recommends some future work.

II. RELATED WORK

A. Routing Protocols

Routing is a process of delivering data packets from a source (sender) node to destination (receiver) node on a network. Routing protocols deal with the maintenance, creation, establishment and discovery of such routes [4]. Routing protocols are based on three protocol classification categories: reactive, proactive and hybrid.

Reactive protocols also referred to as on-demand protocols create a route from source to destination only when needed, i.e. when there is actual data to be sent. This scheme uses network flooding to find the routes [4] [5]. This protocol scheme is suited for mobile ad-hoc networks where there are frequent topological changes due to the mobility of routers [1] [5]. According to [5], flood based route discovery provides high network connectivity and low message overhead. More importantly, the method does not waste bandwidth by propagating control packets when it is not necessary [1]. This scheme, however, leads to higher latency on the network because of route discovery. [4] Argues that reactive protocols are more suitable for a network with static traffic patterns whilst proactive protocols suit dense networks with bursty traffic patterns [4].

Ad-hoc on demand distance vector (AODV) is one of the popular reactive protocols and hence creates routes on demand. AODV has single path routing and is based on hop-by-hop routing [3]. Single path routing means that a node can only have one path towards a destination [3]. The AODV routing table only stores information about the best next-hop towards a destination [4]. Sequence numbers are used to ensure loop-free routes and to ensure the freshness of the routing information [6] [7]. The AODV protocol uses unicast, broadcast, as well as multicast for communication on the network. It uses broadcast to flood route requests, then the intermediate nodes and the destination nodes send a unicast route reply [7]. There are multicast groups where a multicast of sequence numbers takes place [6]. On-demand multipath distance vector routing (AOMDV) was developed to alleviate link failures and link breaking suffered in AODV by using multipath routes [6] [7].

Dynamic source routing (DSR) and SrcRR are other reactive protocols which are based on source routing. DSR as defined by Jonhson et al. is a simple and efficient routing protocol designed specifically for multihop wireless ad hoc networks with mobile nodes [8]. DSR has two fundamental mechanisms: route discovery and route maintenance. These two components allow the nodes to self-discover and self-maintain routes to dynamic destinations in the ad hoc network. SrcRR is based on DSR but differs because it uses ETX routing metric (see next section).

Other on demand routing protocols are based on link reversal routing (LRR). LRR suits ad hoc networks due to their ease on adaptability and scalability with more emphasis on fast changing topology networks [9].

Associativity Based Routing (ABR) is a source-initiated reactive protocol. It is a bandwidth efficient distributed routing protocol used in ad hoc networks [10]. ABR uses periodic beacons to let neighbours know about other neighbours' existence. Another beacon based reactive protocol is the Signal Stability Routing protocol (SSR). This protocol selects routes based on signal strength and nodes location stability.

In proactive protocols, each node in the network maintains a table containing routing information of the entire network. Each node then periodically broadcasts control packets (hello packets) to the whole network to let other nodes know about its existence. The routing information is periodically updated to maintain the adequacy of the routing information and thus the network will always be up to date with respect to topological changes. The biggest advantage of this scheme is the minimization of route discovery delay and consequently lower latency in delivering a packet. However because of the periodic updates of control messages that get propagated through the entire network, the overhead increases. Thus, bandwidth consumption also rises. Proactive protocols are also known as table-driven protocols and consume memory space.

Optimized link state routing (OLSR) protocol is a proactive protocol based on a link state algorithm. OLSR's objective is to reduce the size of the control packets as well as the overhead cost by broadcasting control packets [3]. This protocol is an optimization of the link state protocol for mobile ad-hoc networks [3]. It uses a hop by hop routing metric. Multipoint relays (MPR) are the key concepts in OLSR. MPRs are the subsets of the neighbours of which a node uses to forward broadcast messages. MPRs reduce duplicate retransmission in the same region and thus minimize flooding overhead [3].

Destination-Sequenced Distance Vector (DSDV) is also a proactive routing protocol developed by Perkins et al. based on the classic Bellman-Ford routing algorithm [11]. Global State Routing (GSR) is a link state MAC-efficient protocol similar to DSDV. The main goal of this protocol is to address the shortcomings endured in many LS (link state) protocols such as flooding of routing messages. Thus GSR controls the size and the number of the control packets in order to achieve optimized MAC throughput.

The Wireless Routing Protocol (WRP) is a proactive distance vector routing protocol aimed at maintaining routing information on the network [12]. Each node in the network maintains four routing tables: distance table, routing table, link cost table and message retransmission list (MRL) table [11] [12]. Fisheye State routing (FSR) is also an LS routing protocol inspired by the fish-eye technique created to reduce the size of information required for graphical data representation [13]. Clusterhead Gateway Switch Routing (CGSR) is a cluster based proactive protocol which uses the DSDV routing algorithm. Nodes are grouped into clusters where cluster-heads are elected.

The BATMAN algorithm (described in Section I) is also a proactive protocol. However, it experiences serious flaws in dealing with asymmetric links. BATMAN advanced, referred to as BATMAN-adv, is a Layer 2 protocol introduced to overcome this setback by using a Transmit Quality (TQ) algorithm. BATMAN-adv consists of two fundamental functions: receiving link quality (RQ) and transmit link quality. Receiving link quality deals with the probability of transmitting a packet successfully towards a node [15]. The transmitting link illustrates the probability of transmitting a packet successfully towards a neighbor [15]. TQ is the most important because RQ does not influence the routing decision. RQ is determined by the by the number of received OGMs. Echo link quality (EQ) is the number of the rebroadcasted OGMs from neighbors. TQ is calculated by dividing the EQ by the RQ i.e. $TQ = EQ/RQ$ [15].

Hybrid protocols exhibit the behavioural design of the two above mentioned protocols. Hybrid protocols are very challenging because the switch from one protocol to another needs to be very sharp. However, this is still a major concern and thus hybrid protocols are still theoretical rather than practical due to their complex implementation [1].

MeshDv is a hybrid protocol which uses the combination of proactive route computation for the routers and on-demand path request for clients [16]. The proactive route is based on the destination-sequenced distance vector (DSDV) protocol [16].

Zone Routing Protocol (ZRP) is a zone based hybrid protocol. ZRP proactively maintains routing information for the local neighbourhood, referred to as the routing zone. It reactively acquires routes to destinations that are outside the routing zone. Zone-based Hierarchical Link State (ZHLS) routing protocol is another zone based hybrid routing protocol, and is based on global positioning system (GPS). Other hybrid protocols includes SHARP (Hybrid Adaptive Routing Protocol) and HARP (Hybrid ad hoc routing).

B. Routing metrics

Routing protocols use metrics to select the best routing path. Several situation-aware routing metrics have been proposed, as well as applied, in many routing protocols.

The hop count routing metric counts the number of hops between a sender and its destination. Hop count is commonly used in routing protocols such as AODV, DSR and DSDV [5]. Hop count is simple to compute when compared to other metrics, and this is the main reason it has been preferred by many routing protocols. However, hop count does not consider packet loss or bandwidth, and hence results in low throughput [5].

Expected transmission count (ETX) is a situation-aware metric which considers the number of MAC layer transmissions needed to successfully deliver a packet through a link [5] [17]. The ETX metric captures the effects of packet loss and path length. Each node broadcasts probe packets to its neighbors and they send a back a reply/report [17]. The metric is calculated by the number of probe packets received by its neighbor in both directions [12]. ETX is isotonic, thus ensures easy calculations of minimum weight paths [5]. The ETX metric does not consider bandwidth of the links, interference, or the link transmission variance [5].

The Expected transmission time (ETT) metric was developed to overcome the shortcomings of ETX and hence it is an optimization of ETX. ETT takes bandwidth and link transmission difference into consideration for its path selection computation. The ETT of a link is measured by the expected Layer 2 durations it takes to successfully transmit a packet through that link [5]. However, since ETT uses a single path channel interference (both inter-flow and intra-flow), this remains a major drawback in ETT, like in ETX [5]. Intra-flow interference is interference between intermediate routers sharing the same path while inter-flow is between neighboring routers competing for the same channel.

Weighted cumulative ETT (WCETT) was designed to overcome the shortcomings of both ETX and ETT in order to reduce intra-flow interference [5]. This is done through the use of multi path channels. However, since it is not isotonic, it has not been used by any algorithm [5]. Another drawback of WCETT it does not consider inter-flow interference and its effects. The metric of interference and channel-switching (MIC) addresses the shortcomings of WCETT by considering inter-flow interference and as well as solving the some of the non-isotonic effects.

C. Moving averages

The idea of using moving averages is not new. It is commonly used in economic systems for computing and plotting stock markets. Moving average (MA) is an arithmetic result calculated by averaging a number of past data points [18]. A simple moving average (SMA) is calculated using the mean of a given set of values. The sum of the set is divided by the number of elements in that set. It is similar to a statistical computation of a mean yet different by the fact that only a recent n number of data values are considered.

Exponential moving average (EMA) is another type of MA that gives more weight to recent values in order to make it more responsive newer information [MA]. In relation to routing protocols more weight is applied to more recent OGMs, for example, for precise current link quality estimation. In stock market analysis, EMA uses the formula: $EMA = (P*\alpha) + (\text{previous } EMA*\ (1-\alpha))$ where P = current price, α = smooth factor = $2/(1+N)$ and N is the number of time periods. [] applied this in BATMAN protocol in terms of round trip-time formulated as: $estimateRTT = (1-\alpha)* estimateRTT + \alpha*sampleRTT$ where sampleRTT is the RTT measured with the last packet, α is a smoothing operator with a constant value of~0.125m [www.open-mesh.org].

Weighted moving average (WMA) is another type of MA that also gives more weight to recently received data values. WMA multiplies the most recent value with its sequence value and monotonically decreases with iterations. For example, given a set of 10 values, WMA would multiply the value at index 10 with 10 and value at 9 with 9 and so forth.

D. Routing in mobile phones

BATMAN has been successfully deployed on a new routing device called the mesh potato by Village Telco Project (www.villagetelco.org). A mesh potato is a wireless access point combined with an Asynchronous Telephone Adapter (ADT) suitable for a rural network. The Village Telco team has also successfully ported the BATMAN stack on a selection of Android mobile phones. We aim to use such phones, known as Batphones, as mobile nodes in our hybrid WMN.

III. METHODS

This section describes how we optimized the BATMAN protocol to make situation-aware routing decisions based on link quality. We also describe the experimental design to compare baseline performance to that of the modification.

Given the mobility of mobile nodes, rapid topological changes in a hybrid mesh network are inevitable. Thus, the ideal approach is to take the current network situation into consideration when making routing decisions. In BATMAN, the best link is measured by the highest number of OGMs received from the destination over a current sliding window. Much can happen within a second in an ad hoc wireless network. Any link with a sliding window that records a lot of OGMs at the beginning and fewer at the end due to superior link strength at the beginning stands a chance of being the best as opposed to the one that records a lot towards the end but fewer in total. For example, suppose one has a sliding window of 10, link L1 records [1111100000] with 5 OGMs at the front, and link L2 [0000001111] with 4 OGMs seen at the end. BATMAN will chose L1 as the best next hop because of the higher number of OGMs, but actually, the current best option would be L2 because the most OGMs have arrived there more recently.

Our method prioritizes the recently received OGMs in the sliding window, and would therefore correctly choose L2 over L1. We sum the indices on which OGMs were recorded in a given window. From the example above, we would have link L1: 1+2+3+4+5 = 15 and link L2: 7+8+9+10 = 34. This is a more accurate numeric representation describing the current situation of the two links.

This section explains the experimental design and procedure to evaluate our BATMAN modification. We created a mesh network composed of four nodes as shown in Fig. 2. All nodes ran Linux version 2.6.32-31 with 802.11bg network cards. We used BATMAN advanced version BAD2010.1.0. Note that node B is the server, and node C is a laptop that we can move around during the tests.

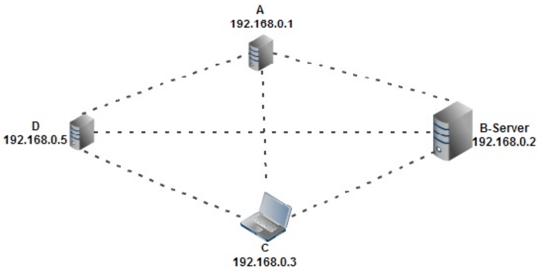


Fig. 2 The experiment test bed.

The experiment was designed to compare the performance of unmodified BATMAN-adv with our modified version on a dynamic mesh network with and without congestion. Our main objective is to show that situation-aware routing is viable and effective in a hybrid WMN. The test parameters examined were jitter, packet loss and throughput. We assume that jitter also covers latency.

We installed Iperf on all of the nodes to conduct the tests. Node B was used as a server (receiver) whilst the others were clients (senders). We configured Iperf to send packet flows representing voice packets to the server. We set a transfer interval of 60 seconds with a report back of 10 seconds. This was run 10 times for each parameter (herein referred as 10 flows). During the transfer interval, Iperf sent about 4000 UDP (User Datagram protocol) packets, about 665 each 10 seconds, with a maximum size of 1500 bytes. The parameters were tested with a selection of transfer rates and buffer sizes. The default settings were 1MB/s (megabytes per second) of bandwidth and 41 KB (kilobytes) for the buffer size. The transfer rate was regulated over 1MB, 100MB and 150MB speeds whilst buffer size was varied over 41KB, 31KB and 11KB. The first comparison combination consisted of all the transfer speeds with the default buffer size of 41KB. The second comparison combination applied the buffer size variations to the default transfer rate of 1MB/s. Lastly, the 150MB/s rate was applied to the 11KB buffer size to achieve maximum congestion of the compared rates and buffer sizes.

IV. RESULTS AND DISCUSSION

The results of jitter, packet loss and throughput comparisons in these combinations are presented in Table 1, Table 2 and Table 3, respectively. We measured average jitter, packet loss and throughput with the rate/buffer size combinations mentioned above. Fig. 3 and Fig. 4 are illustrative examples of packet loss only.

Jitter						
	1MB 41KB	100MB 41KB	150MB 41KB	150MB 11KB	1MB 31KB	1MB 11KB
BATMAN-ADV						
A	32.06	45.24	48.35	45.19	20.04	17.47
C	27.42	41.62	43.43	28.08	13.45	20.43
D	36.91	53.61	54.78	37.83	28.85	32.39
BATMAN-ADV modified						
A	150.62	38.59	336.20	109.30	59.75	101.80
C	0.25	102.60	142.00	179.50	258.40	131.30
D	216.58	58.18	52.80	45.80	50.72	24.24

Table 1: Jitter comparisons.

Packet Loss						
	1MB 41KB	100MB 41KB	150MB 41KB	150MB 11KB	1MB 31KB	1MB 11KB
BATMAN-ADV						
A	3.80	0.77	1.13	6.07	3.13	2.77
C	6.54	6.01	5.53	4.30	4.90	5.28
D	1.33	1.12	1.26	0.35	0.41	0.51
BATMAN-ADV modified						
A	7.92	3.21	1.73	2.02	2.25	4.03
C	9.45	1.75	2.12	1.27	2.09	1.67
D	8.58	0.94	2.04	5.05	1.54	0.43

Table 2: Packet loss comparisons.

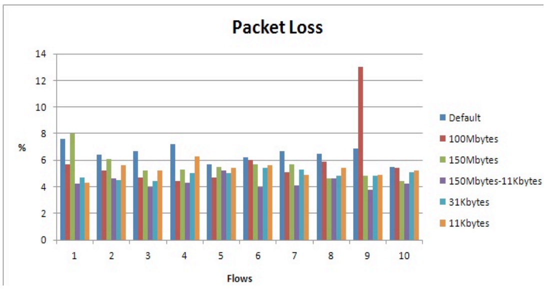


Fig. 3 : Packet loss from node C to B in a congested scenario using BAMTAN-adv original.

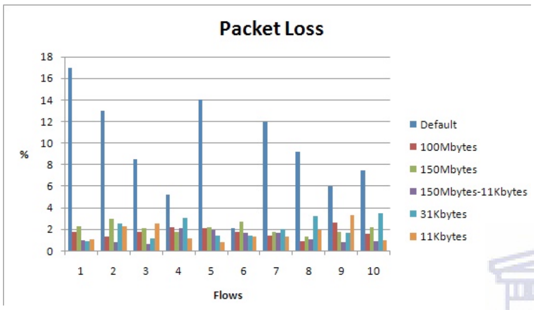


Fig. 4 : Packet loss from node C to node B in a congested scenario using the modified BAMTAN-adv.

Throughput						
	1MB 41KB	100MB 41KB	150MB 41KB	150MB 11KB	1MB 31KB	1MB 11KB
BATMAN-ADV						
A	0.08	0.09	0.08	0.09	0.09	0.08
C	0.09	0.09	0.09	0.09	0.09	0.86
D	0.08	0.09	0.08	0.08	0.08	0.08
BATMAN-ADV modified						
A	0.07	0.06	0.09	0.10	0.07	0.05
C	0.11	0.05	0.05	0.09	0.06	0.06
D	0.06	0.09	0.08	0.09	0.09	0.09

Table 3: Throughput comparisons.

The results show that our metric is well suited for unstable and dynamic networks under strenuous circumstances. The variation of packet latency across a network, known as jitter or packet delay variation (PDV) shows a significant difference between protocol sets. The BATMAN-adv original shows the best (low) PDV of less than 55ms across all variation settings as show in Table 1. The PDV is consistent irrespective of the transfer rate or the buffer size. Node C, which had mobility throughout the tests, exhibits an overall average of 30.80ms across all variation settings while nodes A and D are 38.18 and 42.40, respectively. On the other hand, the modified BATMAN-adv lacks consistency as some points rise abruptly, reaching 336.2ms, while the lowest is 24.24ms in line with the original protocol. PDV consistently increases and appears to do so independently of the variation settings.

The average packet loss results of BATMAN-adv original appear inconsistent in the baseline measurement. The average across all variation settings (i.e. from default 150MB-11KB see Table 2) exhibits some inconsistencies as compared to our modified version which stands at an average of 3.2% for all the links (A = 3.52, C = 3.05 and D = 3.09). BATMAN original has values: A = 2.94, C = 5.42, C 0.83. The most distinctive and significant factor in this case is the consistency of packet loss for BATMAN-adv across all settings while the modification shows reduction as per variation settings. At default, the average packet loss on the three links is about 8%. The loss rate then reduces proportionally to the transfer rate and buffer size. This shows that situation-aware routing metrics perform well on large and inconsistent networks with congested links. The results show no practical relation between jitter/PDV and packet loss.

Unlike packet loss, the consistency in PDV correlates well with the consistency in throughput as shown in Table 1 and Table 3. The average throughput is also independent of the variation settings. The average throughput in BATMAN-adv is consistently at 0.08 MB/s. On the other hand our modified version tends to fluctuates a bit. The maximum recorded throughput in flow for BATMAN-adv is 0.09MB/s while our modified version could reach 3MB/sec in a particular flow but due to its fluctuation tendency, the overall average amounts to 0.76MB/sec. We observe that jitter/PDV and throughput are correlated, i.e. consistent PDV results in a consistent throughput. In terms of throughput, both protocol versions are at par with each.

V. CONCLUSION AND FUTURE WORK

Situation-aware routing seeks to improve QoS on hybrid wireless mesh networks by making routing decisions based on the current situation of the network. BATMAN-adv counts OGMs received as a link quality measurement. We apply a prioritization technique to calculate the link quality metric. We give the more recently received OGMs more weight in deciding the link quality by summing their indices in a given window rather than counting their quantity. Therefore, more recently received OGMs contribute more to the metric in order to give a more precise indication of the current state of a link.

The results show little relation between jitter/PDV and packet loss. Jitter is, however, proportional to throughput. The average throughput achieved on both protocols was almost the same but we noticed that the throughput on our modified version increases as the network grows, and therefore appears to be scalable. Our protocol modification suffered from packet loss at low bandwidth rates but this reduces as the transfer rate increases and buffer size shrinks, i.e. it performs well with congestion. We can infer that increasing the transfer rate with a smaller buffer size is poorly handled by BATMAN-adv original while our modification performs admirably. We conclude that our situation-aware protocol modification shows potential to address issues pertaining to scalable and congested static mesh networks with mobile nodes.

There are some limitations to our experimental design, which possibly had negative impact on the results. The test bed was in a single computer laboratory room with only four nodes. The distance between the nodes was small. Also, there are several other wireless networks accessible in the same room. Although we tried our best to confine our network to free different channel spread, the noted inconsistent packet loss could possibly result from network interference.

In the future, we would like to see how the protocol performs under a wider range of traffic patterns, and also in a more geographically spread mesh network (with more nodes). Our initial plan was to use mobile phones as the mobile nodes but could not because BATMAN-adv has not yet been ported to the mobile phone. Hence our future work can orient toward that scenario.

VI. ACKNOWLEDGEMENTS

We thank Telkom, Cisco, and the THRIP (Technology and Human Resources for Industry Partnership) initiative of the South African Department of Trade and Industry for financial support via the Telkom Centre of Excellence (CoE) programme. THRIP funding is managed by the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF does not accept any liability in regard thereto.

VII. REFERENCES

[1] M. Abolhasan, B. Hagelstein, and J. C. P. Wang, "Real-world performance of current proactive multi-hop mesh protocols," in *Communications, 2009. APCC 2009. 15th Asia-Pacific Conference on*, p. 44–47.

[3] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle, "QoS-enhanced OLSR protocol for mobile ad hoc networks," in *Proceedings of ANWIRE 1st International Workshop, Glasgow Scotland*, 2003, p. 171–183.

[4] R. Pinto, *WMM-Wireless Mesh Monitoring*, Technical report, INESC-ID, 2009.

[5] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[6] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, p. 92–93, 2002.

[7] E. Royer and C. Perkins , "An Implementation Study of the AODV Routing Protocol.", *In Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 3, 2000, pp. 1003-1008.

[8] D. B. Johnson, D. A. Maltz, J. Broch, and others, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad hoc networking*, vol. 5, p. 139–172, 2001.

[9] M. Vainio, "Link reversal routing," *Nokia Group*.

[10] S. A. Shaar, F. A. M. Masoud, A. Murad, R. Al-Shalabi, and G. Kanaan, "Topic," 2006.

[11] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, "Impact of node mobility on MANET routing protocols models," *Journal of Digital Information Management*, vol. 5, no. 1, p. 19, 2007.

[12] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE*, vol. 6, no. 2, p. 46–55, 1999.

[13] G. Pei, M. Gerla, and T. W. Chen, "Fisheye state routing in mobile ad hoc networks," in *ICDCS Workshop on Wireless Networks and Mobile Computing*, 2000, p. D71–D78.

[14] D. Johnson, N. Ntlatlapa, and C. Aichele, "Simple pragmatic approach to mesh routing using BATMAN," in *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, 2008.

[15] S. Wunderlich, M. Lindner and W. Tsai "B.A.T.M.A.N advance" http://www.open-mesh.org/wiki/batman-adv.

[16] L. Iannone and S. Fdida, "Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks," in *IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN)*, 2005, p. 103–110.

[17] M. Caleffi and L. Paura, "Bio-inspired link quality estimation for wireless mesh networks," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009*, 2009, p. 1–6.

[18] C. Murphy, "Moving Averages." http://investopedia.com/inv/pdf/tutorials/MovingAverages.pdf.

Hlabishi I. Kobo obtained a BSc Honours degree in Computer Science from the University of the Western Cape (UWC) in 2009. He is presently studying towards an MSc degree with the Bridging Applications and Networks Group (BANG) at the same institution. His main research interest is now wireless mesh routing protocols on mobile phones.

William D. Tucker is a Senior Lecturer of Computer Science at UWC and leads BANG research there. His research interest includes Internet Protocol (IP) networks and their applications in developing regions.

Xiaoming "Andy" Liu obtained an MSc degree in Computer Science from UWC in 2008. He is now a PhD student interested in IP networks. His research interests include intelligent wireless network management with situation awareness and AI methods, distributed computing, and time series modeling.

Appendix C

Draft paper(final results)

Situation-aware Routing Based on Link Quality for Static Mesh Networks with Mobile Nodes

Hlabishi I. Kobo and William D. Tucker
Department of Computer Science
University of the Western Cape, Private Bag X17, Bellville 7535 South Africa
Tel: +27 21 9592461, Fax: +27 21 9591274
E-mail: {**2654952**, btucker}@uwc.ac.za

Abstract- This paper demonstrates that a situation-aware algorithm based on the current situation of the network improves quality of service for dynamic mesh networks with mobile nodes running BATMAN-adv. BATMAN-adv is a mesh routing protocol that counts beacons as a link quality metric. We modified BATMAN-adv to give more recently received beacons more weight, thereby giving a more precise indication of the current state of a link. We then compared the original protocol with our modification in a small laboratory testbed and later on a larger testbed. Results show little relation between jitter and packet loss. Jitter is, however, proportional to throughput. BATMAN-adv achieved the best jitter and packet loss on the small testbed. The average throughput achieved on both protocols was almost the same. The larger scale results, which had an increased scalability and mobility rate showed a significant improvement. Thus our algorithm performed better than the later in all performance metrics. We conclude that our situation-aware protocol modification offers potential solution to address issues pertaining to mobility, congestion and scalability in dynamic mesh networks with mobile nodes.

Categories and Subject Descriptors
C.2 [**COMPUTER-COMMUNICATION NETWORKS**]: Wireless Communication-Routing protocols

I. INTRODUCTION

This paper presents a situation-aware routing metric calculation prioritizing the most recent link quality data to inform routing decisions on static wireless mesh networks (WMN) with mobile nodes. This type of network can be referred to as dynamic mesh network. We believe these types of mesh networks will become prevalent as mesh network protocols improve and mobile devices become more powerful and able to run such protocols. Fig. 1 illustrates the dynamic mesh network concept. Our goal is to improve the effective usage of link quality using situation-aware routing in BATMAN (described below) in order to optimize quality of service (QoS) and throughput on such networks. We chose BATMAN because it has high stability level and high packet delivery ratio [1].

A Better approach to mobile ad-hoc network (BATMAN) is a proactive mesh network routing protocol. BATMAN's control messages, called originator messages (OGMs), are relatively small packets of about 52 bytes. BATMAN's nodes do not maintain the routing information of the entire network [1]. Rather, each node only maintains information about the best next-hop towards the destination [1][2]. This reduces the signal overhead and avoids unnecessary knowledge about the whole network. The objective of this protocol is to enhance the probability of delivering a packet. The protocol maintains information about the existence of a node and thus does not check the quality of a link [3].

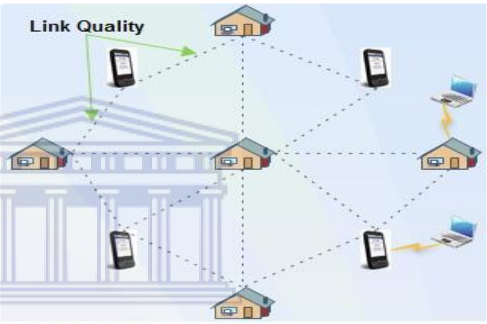


Fig. 1: A dynamic mesh network, applicable to a rural area, with static mesh routers inside homes, and mobile mesh nodes on cell phones. Note that link quality in such networks is continually changing as phones move around.

All BATMAN nodes periodically send/broadcast OGMs. Each OGM contains the original sender's address, address of the node rebroadcasting the OGM, TTL (time to live) and a sequence number. The sequence number is incremented for each OGM, i.e. the first OGM gets 1 and so on. Thus, BATMAN also keeps track of the freshness of an OGM. Any sequence number received with a value lower than the previous one gets dropped [2]. The TTL is used to limit the number of hops on which the packets must pass through before it expires (gets dropped). Upon receiving the OGM, each node then rebroadcasts it to its neighbors. However, each node only rebroadcasts OGMs coming through the current best next-hop. The number and the reliability of the OGMs determine the route discovery as well as neighbor selection.

This paper describes our efforts to optimize BATMAN for a dynamic mesh network. The crux of the problem is to optimize the routing protocol so it can adapt and react quickly to rapid and dynamic topological changes. We propose situation-aware methods to improve the routing decisions based on link quality to achieve better QoS and throughput. BATMAN's routing algorithm checks for the existence of a link and increases the probability of delivering a packet through that link [4]. Our method adapts the routing protocol to use a simple weighting mechanism. We believe recent packets provide a clearer indication of link quality at a particular moment in time so we give recently received packets more weight in the routing decision. We tested the protocol enhancement on two experimental testbed setups: small scale and large scale. The core factors of this research include mobility, congestion and scalability. The results from the small testbed are encouraging, and not too far off the mark from the original BATMAN. They are better than the original BATMAN in the large scale network.

The rest of the paper is organized as follows: Section II reviews some related work. Section III presents the methods for the protocol enhancement and the experimental setting. Section IV presents preliminary results and final results, and discusses them. Section V concludes the paper and recommends some future work.

II. RELATED WORK

A. Routing Protocols

Routing is a process of delivering data packets from a source (sender) node to destination (receiver) node on a network. Routing protocols deal with the maintenance, creation, establishment and discovery of such routes [2]. Routing protocols are based on three protocol classification categories: reactive, proactive and hybrid.

Reactive protocols are on-demand; they create a route from source to destination only when needed, i.e. when there is actual data to be sent. This scheme uses network flooding to find the routes [2][5]. It is suited for mobile ad-hoc networks where there are frequent topological changes due to the mobility of routers [1][5]. According to [5], flood based route discovery provides high network connectivity and low message overhead. More importantly, the method does not waste bandwidth by propagating control packets when it is not necessary [1]. This scheme, however, leads to higher latency on the network because of route discovery. [2] Argues that reactive protocols are more suitable for a network with static traffic patterns whilst proactive protocols suit dense networks with bursty traffic patterns.

Ad-hoc on demand distance vector (AODV) is one of the popular reactive protocols and hence creates routes on demand. AODV has single path routing and is based on hop-by-hop routing [2]. Single path routing means that a node can only have one path towards a destination [2]. The AODV routing table only stores information about the best next-hop towards a destination [2][6]. Sequence numbers are used to ensure loop-free routes and to ensure the freshness of the routing information [7]. The AODV protocol uses unicast, broadcast, as well as multicast for communication on the network. It uses broadcast to flood route requests, then the intermediate nodes and the destination nodes send a unicast route reply [7]. There are multicast groups where a multicast of sequence numbers takes place [6][7]. Other reactive protocols include Ad On-demand multipath distance vector routing (AOMDV), Dynamic Source Routing (DSR), Link Reversal Routing (LRR) and Associativity Based Routing (ABR).

In proactive protocols, each node in the network maintains a table containing routing information of the entire network. Each node then periodically broadcasts control packets to the whole network to let other nodes know about its existence. The routing information is periodically updated to maintain the adequacy of the routing information. The biggest advantage of this scheme is the minimization of route discovery delay and consequently lower latency in delivering a packet. However because of the periodic updates of control messages that get propagated through the entire network, the overhead increases. Thus, bandwidth consumption also rises and also requires a lot of memory for the tables.

Optimized link state routing (OLSR) protocol is a proactive protocol based on a link state (LS) algorithm. OLSR's objective is to reduce the size of the control packets as well as the overhead cost by broadcasting control packets [4]. This protocol is an optimization of the link state protocol for mobile ad-hoc networks [4]. It uses a hop by hop routing metric. Multipoint relays (MPR) are the key concepts in OLSR. MPRs are the subsets of the neighbours of which a node uses to forward broadcast messages. MPRs reduce duplicate retransmission in the same region and thus minimize flooding overhead [4].

Fisheye State routing (FSR) is also a LS routing protocol inspired by the fish-eye technique created to reduce the size of information required for graphical data representation [8]. Other proactive protocols include Destination-Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP).

The BATMAN algorithm (described in Section I) is also a proactive protocol. However, it experiences serious flaws in dealing with asymmetric links. BATMAN advanced, referred to as BATMAN-adv, is a Layer 2 protocol introduced to overcome this setback by using a Transmit Quality (TQ) algorithm. BATMAN-adv consists of two fundamental functions: receiving link quality (RQ) and transmit link quality. Receiving link quality deals with the probability of transmitting a packet successfully towards a node [9]. The transmitting link illustrates the probability of transmitting a packet successfully towards a neighbor [9]. TQ is the most important because RQ does not influence the routing decision. TQ is determined by the by the number of received OGMs. Echo link quality (EQ) is the number of the rebroadcasted OGMs from neighbors. TQ is calculated by dividing the EQ by the RQ i.e. TQ = EQ/RQ [9].

Hybrid protocols exhibit the behavioural design of the two above mentioned protocols. Hybrid protocols are very challenging because the switch from one protocol to another needs to be very sharp. However, this is still a major concern and thus hybrid protocols are still theoretical rather than practical due to their complex implementation [1].

MeshDv is a hybrid protocol which uses the combination of proactive route computation for the routers and on-demand path request for clients [10]. The proactive route is based on the destination-sequenced distance vector (DSDV) protocol [10].

Zone Routing Protocol (ZRP) is a zone based hybrid protocol. ZRP proactively maintains routing information for the local neighbourhood, referred to as the routing zone [11]. It reactively acquires routes to destinations that are outside the routing zone. Zone-based Hierarchical Link State

(ZHLS) routing protocol is another zone based hybrid routing protocol, and is based on global positioning system (GPS) [12]. Other hybrid protocols includes SHARP (Hybrid Adaptive Routing Protocol) and HARP (Hybrid ad hoc routing).

B. *Routing metrics*

Routing protocols use metrics to select the best routing path. Several situation-aware routing metrics have been proposed, as well as applied, in many routing protocols.

Expected transmission count (ETX) is a situation-aware metric which considers the number of MAC layer transmissions needed to successfully deliver a packet through a link [5][13]. The ETX metric captures the effects of packet loss and path length. Each node broadcasts probe packets to its neighbors and they send a back a reply/report [13]. The metric is calculated by the number of probe packets received by its neighbor in both directions. ETX is isotonic, thus ensures easy calculations of minimum weight paths [5]. The ETX metric does not consider bandwidth, interference, or the link transmission variance [5]. Other metrics includes expected transmission time (ETT), weighted cumulative ETT (WCETT) and the metric of interference and channel-switching (MIC)

III. METHODS

This section describes how we optimized the BATMAN protocol to make situation-aware routing decisions based on link quality.

Given the mobility of mobile nodes, rapid topological changes in a dynamic mesh network are inevitable. Thus, the ideal approach is to take the current network situation into consideration when making routing decisions. In BATMAN, the best link is measured by the highest number of OGMs received from the destination over a current sliding window. Much can happen within a second in an ad hoc wireless network. Any link with a sliding window that records a lot of OGMs at the beginning and fewer at the end due to superior link strength at the beginning stands a chance of being the best as opposed to the one that records a lot towards the end but fewer in total. For example, suppose one has a sliding window of 10, link L1 records [1111100000] with 5 OGMs at the front, and link L2 [0000001111] with 4 OGMs seen at the end. BATMAN will chose L1 as the best next hop because of the higher number of OGMs, but actually, the current best option would be L2 because the most OGMs have arrived there more recently.

Our method prioritizes the recently received OGMs in the sliding window, and would therefore correctly choose L2 over L1. We sum the indices on which OGMs were recorded in a given window. From the example above, we would have link L1: 1+2+3+4+5 = 15 and link L2: 7+8+9+10 = 34. This is a more accurate numeric representation describing the current situation of the two links.

This section explains the experimental design and procedure to evaluate our BATMAN modification. We created a preliminary mesh network testbed composed of four nodes as shown in Fig. 2. Subsequently the mesh testbed was extended to 12 nodes (see Fig. 3).

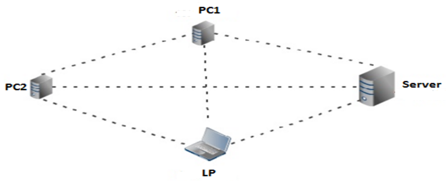


Fig. 2: The small scale experimental testbed, consist of 2 static and 1 mobile nodes. It is controllable and easy to debug. Adopted from [14] where they also tested BATMAN.

Eight mesh potato nodes were added to the initial testbed. A mesh potato is a wireless access point combined with an Asynchronous Telephone Adapter (ADT). Two laptops were used in this case to take the mobility evaluation to length. All nodes ran Linux version 2.6.32-31 with 802.11bg network cards. BATMAN advanced version 2011.2.0 was used.

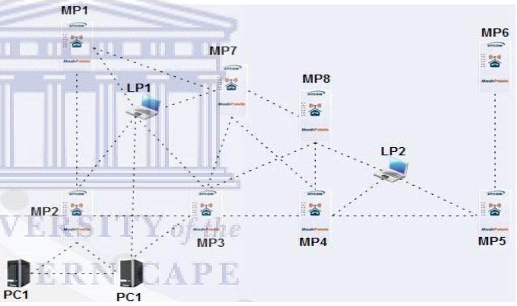


Fig. 3: The larger scale testbed with 12 nodes used to evaluate the impact of network growth on BATMAN oriented mesh network.

The experiment was designed to compare the performance of unmodified BATMAN-adv with our modified version on a dynamic mesh network. Our main objective is to show that situation-aware routing is viable and effective in a dynamic WMN. The test parameters examined were jitter, packet loss and throughput. We assume that jitter also covers latency. These performance metrics were used to investigate the impact of mobility, congestion and scalability on dynamic networks. The mobility was realized by moving the laptops around in hand during the experimentation. The congestion procedure is explained below while the scalability was evaluated by comparing the overall performance between the small scale and the large scale testbed.

Iperf was used to generate packets and monitor performance. We set a transfer interval of 60 seconds with a report back of 10 seconds. This was run 10 times for each parameter (herein referred as 10 flows). During the transfer interval, Iperf sent about 4000 UDP (User Datagram protocol) packets, with a maximum size of 1500 bytes. The parameters were tested with a selection of transfer rates and buffer sizes. The default settings were 1MB/s (megabytes per second) of bandwidth and 41 KB (kilobytes) for the buffer size. The transfer rate was regulated over 1MB, 100MB and 150MB speeds whilst buffer size was varied over 41KB, 31KB and 11KB. The first comparison combination consisted of all the transfer speeds with the default buffer size of 41KB. The second comparison combination applied the buffer size variations to the default transfer rate of 1MB/s. Lastly, the 150MB/s rate was applied to the 11KB buffer size to achieve maximum congestion of the compared rates and buffer sizes.

IV. RESULTS AND DISCUSSION

The results presented in Table 1, Table 2 and Table 3 of jitter, packet loss and throughput respectively are from the larger scale testbed. The small scale results are summarized. We measured average jitter, packet loss and throughput with the rate/buffer size combinations mentioned above. Each cell is an average of 10 flows e.g. the value 28.08ms in Table 1is an average of 10 flows fromLP1 to the server under 1MB-41KB settings.

Table 1: Jitter comparison from larger scale testbed.

Jitter						
	1M 41K	100M 41K	150M 41K	150M 11K	1M 31K	1M 11K
BATMAN-adv						
LP1	28.08	51.10	49.35	302.5	27.94	21.97
LP2	40.29	54.70	55.22	31.85	39.67	22.19
PC2	96.19	46.29	47.01	27.72	19.07	16.51
BATMAN-adv modified						
LP1	35.12	57.49	52.93	28.87	26.34	21.46
LP2	26.30	48.67	52.72	31.88	20.17	21.66
PC2	2.067	48.67	52.72	31.88	20.17	21.66

Table 2: Packet loss comparison from larger scale testbed

Packet Loss						
	1M 41K	100M 41K	150M 41K	150M 11K	1M 31K	1M 11K
BATMAN-ADV						
LP1	0.17	0.20	0.12	0.18	0.22	0.14
LP2	0.77	1.93	1.39	3.3	3.47	0.82
PC2	0.006	0.45	0.25	0.20	0.10	0.13
BATMAN-ADV modified						
LP1	0.02	0.03	0.03	0.018	0.042	0.040
LP2	0.14	0.03	0.05	0.11	0.04	0.15
PC2	0.002	0.21	0.08	0.16	0.28	0.22

Table 3: Throughput comparison from larger scale testbed

Throughput						
	1M 41K	100M 41K	150M 41K	150M 11K	1M 31K	1M 11K
BATMAN-adv						
LP1	0.08	0.08	0.02	0.09	0.08	0.08
LP2	0.08	0.08	0.08	0.006	0.07	0.08
PC2	0.04	0.08	0.09	0.09	0.09	0.09
BATMAN-adv modified						
LP1	0.08	0.08	0.09	0.09	0.08	0.09
LP2	0.08	0.08	0.09	0.08	0.09	0.08
PC2	0.12	0.09	0.09	0.08	0.083	0.08

The results show that our metric is well suited for unstable and dynamic networks under strenuous circumstances. Jitter which is a variation of packet delay, showed a huge difference between the protocols in the small scale testbed. BATMAN-adv original shows the best (low) jitter of less than 55ms across all variation settings. The jitter is consistent irrespective of the transfer rate or the buffer size. Node LP, which had mobility throughout the tests, achieved the best jitter compared to PC1 and PC2. On the other hand, the modified BATMAN-adv lacks consistency as some points rise abruptly, reaching 336.2ms. The results from the larger testbed are however different. The average jitter from BATMAN-adv original is (LP1 = 80.16, LP2 = 40.66, PC2 = 42.13) while the modified version is (LP1 = 37.02, LP2 = 33.56, PC2 = 29.52). The modified protocol recorded the best and consistent jitter compared to the original and thus improving significantly from small scale testbed.

The average packet loss results for BATMAN-adv original appear inconsistent in the preliminary tests. The most distinctive and significant factor in this case is the consistency of packet loss for BATMAN-adv across all settings while the modification shows reduction as per variation settings. At default, the average packet loss on the three links is about 8%. The loss rate then reduces proportionally to the transfer rate and buffer size. This shows that situation-aware routing metrics perform well on large and inconsistent networks with congested links. In the larger scale tests, the packet loss reduced significantly with the average in BATMAN-adv original (LP1 = 0.17, LP2 = 1.95, PC2 = 0.19) and BATMAN-adv modified at (LP1 = 0.029, LP2 = 0.085, PC2 = 0.158). The lowest recorded packet loss can be seen on the mobile nodes, LP1 and LP2 which proves the effectiveness of situation-aware routing technique in dynamic situations. The results show no practical relation between jitter and packet loss.

Unlike packet loss, the consistency in jitter correlates well with the consistency in throughput. The average throughput is also independent of the variation settings. The average throughput for BATMAN-adv in the small scale is consistently at 0.08 MB/s. On the other hand our modified version tends to fluctuates a bit. The maximum recorded throughput in flow for BATMAN-adv is 0.09MB/s while our modified version could reach 3MB/sec in a particular flow but due to its fluctuation tendency, the overall average amounts to 0.76MB/sec. However in the larger scale testbed, the maximum recorded in a flow was 1.12Mbyte per second in both protocols. The original BATMAN-adv recorded average (LP1 = 0.074, LP2 = 0.066, PC2 = 0.081) while the modified version recorded (LP1 = 0.084, LP2 = 0.084, PC2 = 0.090). The overall average between the three averages is 0.074 and 0.086 for BATMAN-adv original and BATMAN-adv modified respectively. Thus the modified protocol proved to be superior.

V.CONCLUSION AND FUTURE WORK

This paper demonstrated that the situation-aware method improves QoS on dynamic WMNs by making routing decisions based on the current situation of the network.

BATMAN-adv counts OGMs received as a link quality measurement. We apply a prioritization technique to calculate the link quality metric. We give the more recently received OGMs more weight in deciding the link quality by summing their indices in a given window rather than counting their quantity. Therefore, more recently received OGMs contribute more to the metric in order to give a more precise indication of the current state of a link.

The results show little relation between jitter/PDV and packet loss. Jitter is, however, proportional to throughput. The original BATMAN-adv achieved the best jitter in the small scale experiment. The average throughput achieved on both protocols was almost the same. Our protocol modification suffered from packet loss at low bandwidth rates but this reduces as the transfer rate increases and buffer size shrinks, i.e. it performs well with congestion. However in the larger scale experiment, the modified version outperformed the original version in all performance metrics used. This was in the presence of an increased mobility and scalability. The congestion level was the same in both experimental setups. We have demonstrated that situation-aware routing offers a great potential solution to address issues pertaining mobility, congestion and scalability in dynamic mesh networks. We can also deduce that the number of nodes in any mesh network is directly proportional to its performance.

There were some **limitations** in both experimental design setups. The preliminary (small scale) testbed was in a single computer laboratory room with only four nodes. The distance between the nodes was small. Also, there are several other wireless networks accessible in the same room. Despite having a larger scale network testbed ideally perceived close to a real rural network it was still not enough. The challenge in this case being was the walls and the glasses dividing the building inside, the walls have built-in steals. These elements dissipate the network signal negating the performance. The persistent packet loss rate encountered in the larger scale testbed could possibly result from network interference.

In **future**, we would like to see how the protocol performs under a wider range of traffic patterns, and also in a more geographically spread mesh network and therefore a rural area is ideal environment to take this forward. It ought to be quite straightforward to a) recode the decimal-based index sliding window with binary bit-shifting to get it to consume fewer clock cycles and b) port it to BATMAN running on mobile phones such as Batphone.

VI. ACKNOWLEDGEMENTS

We thank Telkom, Cisco, and the THRIP (Technology and Human Resources for Industry Partnership) initiative of the South African Department of Trade and Industry for financial support via the Telkom Centre of Excellence (CoE) programme. THRIP funding is managed by the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF does not accept any liability in regard thereto.

REFERENCES

[1] M. Abolhasan, B. Hagelstein, and J. C. P. Wang, “Real-world performance of current proactive multi-hop mesh protocols,” *15th Asia-Pacific Conference on Communications, 2009. APCC 2009*, pp. 44–47.

[2] R. Pinto, *WMM-Wireless Mesh Monitoring*, Technical report, INESC-ID, 2009.

[3] D. Johnson, N. Ntlatlapa, and C. Aichele, “Simple pragmatic approach to mesh routing using BATMAN,” *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, 2008, pp. 6–7.

[4] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle, “QoS-enhanced OLSR protocol for mobile ad hoc networks,” *Proceedings of ANWIRE 1st International Workshop, Glasgow Scotland*, 2003, pp. 171–183.

[5] Y. Yang, J. Wang, and R. Kravets, “Designing routing metrics for mesh networks,” *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[6] E. M. Royer and C. E. Perkins, “An implementation study of the AODV routing protocol,” in *Wireless Communications and Networking Conference, 2000. WCNC. IEEE*, vol. 3, 200, pp. 1003–1008.

[7] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, 2000, pp. 92–93.

[8] G. Pei, M. Gerla, and T. W. Chen, “Fisheye state routing in mobile ad hoc networks,” *ICDCS Workshop on Wireless Networks and Mobile Computing*, 2000, pp. D71–D78.

[9] S. Wunderlich, M. Lindner and W. Tsai “B.A.T.M.A.N advance” http://www.open-mesh.org/wiki/batman-adv.

[10] L. Iannone and S. Fdida, “Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks,” *IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN)*, 2005, pp. 103–110.

[11] Z. J. Haas and M. R. Pearlman, “The performance of query control schemes for the zone routing protocol,” *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 4, 2001, p. 427–438.

[12] M. Joa-Ng and I. T. Lu, “A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, 1999, pp. 1415–1425.

[13] M. Caleffi and L. Paura, “Bio-inspired link quality estimation for wireless mesh networks,” *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009*, 2009, pp. 1–6.

[14] R. G. Garroppo, S. Giordano, and L. Tavanti, “Experimental evaluation of two open source solutions for wireless mesh routing at layer two,” *5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC)* , 2010, pp. 232–237.