GSAR: GREEDY STAND-ALONE POSITION-BASED ROUTING PROTOCOL TO AVOID HOLE PROBLEM OCCURRENCE IN MOBILE AD HOC NETWORKS

MAHMOUD ALI AL-SHUGRAN

DOCTOR OF PHILOSOPHY
UNIVERSITY UTARA MALAYSIA
2014

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Proses penentuan laluan di dalam Rangkaian Mudah Alih Ad Hoc (MANET) adalah sukar disebabkan kekerapan perubahan topologi serta keterbatasan sumber. Oleh itu, mereka bentuk protokol laluan yang boleh dipercayai, dinamik serta mampu memenuhi kehendak MANET amatlah diperlukan. Strategi Penghantaran Rakus (GFS) merupakan strategi yang paling banyak digunakan dalam protokol laluan berasaskan posisi. Algoritma GFS direka bentuk sebagai protokol berprestasi tinggi yang menggunakan kiraan hop untuk mendapatkan laluan paling dekat. Walau bagaimanapun, GFS tidak mengambil kira kehendak MANET yang lain. Oleh itu, ianya tidak mencukupi untuk membuat pengiraan laluan yang boleh dipercayai. Kajian ini bertujuan mempertingkatkan GFS sedia ada kepada protokol laluan yang dinamik, kendiri, boleh bertindak balas dengan pantas terhadap kehendak MANET, serta berupaya menyediakan laluan yang boleh dipercayai dalam kalangan nod yang berhubung. Untuk mencapai matlamat ini, dua mekanisme telah diusulkan sebagai penambahbaikan terhadap GFS yang sedia ada iaitu Mekanisme Pengemaskinian Mata Arah Dinamik (DBUM) dan Mekanisme Keandalan Anggaran Dinamik dan Reaktif dengan Metrik Terpilih (DRESM). Fungsi utama algoritma DBUM adalah untuk menyediakan nod dengan maklumat baru tentang status nod di sekitarnya. Fungsi algoritma DRESM pula adalah untuk membuat keputusan penghantaran berdasarkan pelbagai metrik laluan. Kedua-dua mekanisme ini telah disepadukan di dalam GFS konvensional bagi membentuk protokol Laluan Kendiri Rakus (GSAR). Penilaian ke atas GSAR telah dilakukan menggunakan simulator rangkaian Ns2 berdasarkan set metrik prestasi, senario dan topologi yang telah ditetapkan. Hasil penilaian menunjukkan bahawa GSAR dapat mengetepikan keperluan menggunakan mod pemulihan dan mencapai peningkatan menyeluruh pada prestasi rangkaian berbanding dalam GFS. Dalam pelbagai keadaan pergerakan nod yang diuji, GSAR dapat mengurangkan masalah lubang perangkap kira-kira 87% dan 79% berbanding Protokol Laluan Tanpa Keadaan Perimeter Rakus dan Protokol Laluan Oportunistik Berasaskan Posisi. Kesimpulannya, protokol GSAR merupakan alternatif munasabah kepada protokol laluan berasaskan posisi dalam MANET.

Kata Kunci: Rangkaian Mudah Alih Ad-hoc, Strategi Penghantaran Rakus, Protokol Laluan Berasaskan Posisi, Protokol Laluan Kendiri Rakus

Abstract

The routing process in a Mobile Ad Hoc Network (MANET) poses critical challenges because of its features such as frequent topology changes and resource limitations. Hence, designing a reliable and dynamic routing protocol that satisfies MANET requirements is highly demanded. The Greedy Forwarding Strategy (GFS) has been the most used strategy in position-based routing protocols. The GFS algorithm was designed as a high-performance protocol that adopts hop count in soliciting shortest path. However, the GFS does not consider MANET needs and is therefore insufficient in computing reliable routes. Hence, this study aims to improve the existing GFS by transforming it into a dynamic stand-alone routing protocol that responds swiftly to MANET needs, and provides reliable routes among the communicating nodes. To achieve the aim, two mechanisms were proposed as extensions to the current GFS, namely the Dynamic Beaconing Updates Mechanism (DBUM) and the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM). The DBUM algorithm is mainly responsible for providing a node with up-to-date status information about its neighbours. The DRESM algorithm is responsible for making forwarding decisions based on multiple routing metrics. Both mechanisms were integrated into the conventional GFS to form Greedy Stand-Alone Routing (GSAR) protocol. Evaluations of GSAR were performed using network simulator Ns2 based upon a defined set of performance metrics, scenarios and topologies. The results demonstrate that GSAR eliminates recovery mode mechanism in GFS and consequently improve overall network performance. Under various mobility conditions, GSAR avoids hole problem by about 87% and 79% over Greedy Perimeter Stateless Routing and Position-based Opportunistic Routing Protocol respectively. Therefore, the GSAR protocol is a reasonable alternative to position-based unicast routing protocol in MANET.

Keywords: Mobile Ad hoc Networks, Greedy Forwarding Strategy, Position-based Routing Protocols, Greedy Stand-alone Routing Protocol

Declaration

Some of the works presented in this thesis have been published or submitted as listed below.

[1] Mahmoud Al-Shugran, Osman Ghazali, Suhaidi Hassan, Omar M. Almomani, and Kashif Nisar, "Adaptive and Fuzzy Management for Greedy Routing in Mobile Ad-hoc Networks," in *the Proceeding of 3ed International Conference on Network Applications, Protocols and Services* (*NetApps2012*), Sintok, Malaysia, 19-20 Sep. 2012, pp. 36-41.

[2] Mahmoud Al-Shugran, Osman Ghazali, Suhaidi Hassan, Omar M. Almomani, and Kashif Nisar, "Comparative Performance Evalution of Unicast Routing Protocol in Mobile Ad-hoc Networks," in *the Proceeding of 3ed International Conference on Network Applications, Protocols and Services* (*NetApps2012*), Sintok, Malaysia, 19-20 Sep. 2012, pp. 42-47.

[3] Mahmoud Al-Shugran, Osman Ghazali and Suhaidi Hassan," A General Framework for Greedy Routing in Mobile Ad-hoc Networks, " in t*he Proceeding of International Conference on Advanced Computer Science Applications and Technologies* (*ACSAT2012*), Kuala Lumpur, Malaysia, Indexed by the IEEE Xplore, 26-28 Nov. 2012.

[4] Mahmoud Al-Shugran, Osman Ghazali and Suhaidi Hassan, " Performance Comparison of Position-Based Routing Protocol in the Context of Enhancing Greedy Failure," in *the Proceeding of International Conference on Advanced Computer Science Applications and Technologies* (*ACSAT2012*), Kuala Lumpur, Malaysia, Indexed by the IEEE Xplore, 26-28 Nov. 2012.

[5] Mahmoud Al-shugran, Osman Ghazali, Suhaidi Hassan, Kashif Nisar, and A. Suki M. Arif "A Qualitative Comparison Evaluation of the Greedy Forwarding Strategies in Mobile Ad Hoc Network," *Journal of Network and Computer Applications*, vol. 36, issue 2, pp. 887–897, Impact factor 1.467, Publisher Elsevier, Mar. 2013.

Acknowledgements

Alhamdullilah and thanks Allah (s.w.t) for His blessings to successfully complete this Ph.D thesis. I would like to express my deepest gratitude to my supervisors, Assoc. Prof. Dr. Osman Ghazali and Prof. Dr. Suhaidi Hassan. Their guidance, support, and kindness made this work possible. I feel honoured to have had the opportunity to work with both of them. Osman was always encouraging. Although he was far away on a post-doctorate programme, whenever I had problems, he was always ready to help me resolve them. I would also like to express my sincere gratitude to my examiners Prof. Dr. Sabira Khatun and Prof. Dr. Rahmat Budiarto. Thank you for your invaluable comments and the time you took despite your hectic and busy schedule. You not only enabled me to go on the right path, but also helped me to converge my scattered and incoherent findings to structure them in harmony and consistency and bringing them to a designated end. I thank you for being so knowledgeable, enthusiastic, reassuring, committed and supportive.

This thesis is the result of many years of effort and would not have been possible without the contributions of my friends and my family, which I would like to acknowledge. First, I would like to thank Ns2 users, for improving my understanding of the tools I selected to use. I would also like to express my sincere gratitude to Mr. Tariq Alshugran, I am deeply indebted to him for his insight and suggestion to the thesis topic. He has helped me in many ways and has developed my programming skills to modify the GPSR code to meet the specific alteration for GSAR. I am grateful to Dr. Shengbo Yang, for showing me the way to better simulations with his own work on extending GPSR protocol. Grateful acknowledgment should be mentioned here to Dr. Omar Almomani who guided me through a better way to run a series of simulation experiments. In the same vein, I remain thankful to Dr. Tracy Camp for helping in the venue of selecting the proper mobility models for GSAR. Also, I thank the TEX-LATEX users for improving my understanding of the LYX tool that I selected to text this thesis. I am thankful to Dr. Lian Tze Lim for her help to write this thesis using LYX. To many

others in the Networks Research Lab, I will forever cherish the many useful conversations and insights that helped to shape this study. I would like to thank Dr. A. Suki M. Arif for his help in translating the thesis abstract to Bahasa Melayu. My thanks go to Omar Alshugran, Salah Alshugran, and Tariq Alshugran for their financial support without which this PhD work would not have been possible.

I would like to thank my mother, Jameelah, my father, Ali, and my only sister, Basma, for their love and enthusiasm, and for offering me the stability I needed to accomplish my project. My mother taught me in my formative years and gave me the most solid foundation on which I could develop my skills. I thank her for teaching me her moral principles, for all her loving and caring attention, and her enormous effort to make sure I would survive away from home. My father has been the best example to guide me through life; he taught me the importance of hard work, dedication, persistence and honesty. My parents were the first persons that believed I was able to pursue this project - I thank them for encouraging me to expand my horizons, and for making me a better person. I thank my sister for being my eternal friend and ally, for drowning me with her kindness, and for patiently helping me to focus on my work. I hope I can bring as much happiness to her life as she has brought to mine. Also, I would like to express my gratitude to my brothers, for their moral support and prayers. Their support kept me motivated and inspired me to strive and complete my Ph.D. journey.

Finally, yet importantly, the people who deserve the greatest acknowledgement are the ones to whom this thesis is dedicated: my loving and beautiful wife, Wejdan, my stunning daughters, and my sons. Wejdan is an extraordinary person and her contribution to this work is apparent in every word, she pushed me forward at all times, she gave me constant support and stimulation, she fostered my best ideas, she listened patiently to my constant divagations, and she offered me more understanding than I could ever imagine. I love her dearly and I wish I could one day have the opportunity to help her realise her most ambitious dreams in the way she helped me to realise mine. During my PhD, I was extremely fortunate to experience one of the most rewarding and joyful moments of my life: the birth of my daughter Nor. I only wish I can offer her the same wisdom that her grandparents gave to me.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

ACK	Acknowledgement Packet
AI	Artificial Intelligence
APU	Adaptive Position Update
BCF	Beacon-based Cooperative Forwarding
BPIT	Beacon Packet Interval Time
BP	Beacon Packet
BPsize	Beacon Packet Size
BTA	Backtracking Based Approach
Bw	Bandwidth
CHT	Check Time
COH	Control Overhead
CR	Compass Routing
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance protocol
CTF	Clear To Forward message
CUT	Compulsory Update Technique
DFLCH	Dynamic Fuzzy Logic Controller Check-time
DARPA	Defence Advanced Research Projects Agency
DATA	Data Packet
DLI	Destination Location Information
DBUM	Dynamic Beaconing Updates Mechanism,
DCF	Distributed Coordination Function
DIFS	Distributed Inter Frame Space
DPsize	Data Packet Size
DPS	Destination Prediction Scheme
DRESM	Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism
DREAM	Distance Routing Effect Algorithm for Mobility
DRM	Dynamic Route Maintenance algorithm
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing Protocol

Du	Duration for MAC usage
EED	Average End to End Delay
ELT	Neighbourhood Entry Lifetime
FBPIT	Fixed Beacon Packet Interval Time
FIFO	First-In-First-Out policy
FLDRE	Fuzzy Logic Dynamic Reliability Estimation technique
FLC	Fuzzy Logic Controller
GEDIR	Geographic Distance Routing
GAs	Genatic Algorithms
GFS	Greedy Forwarding Strategy
GG	Gabriel Graph Algorithm
GLS	Scalable Location Service for Geographic Ad Hoc Routing
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
ID	Node Identity
IDOTM	Information Distribution and Outgoing Traffic Control Management technique
IDPI	Inaccuracy in Destination Position Information
INM	Inconsistency of Neighbourhood Matrix
LAR	Location Aided Routing
LLT	link lifetime
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MFR	Most Forward Within Transmission Range
MNs	Wireless Mobile Nodes
MP	Message Packet (used with the RTF, CTF, etc.)
MPsize	Message Packet size
MPDM	Mobility Prediction Using Dead-reckoning Model
NAM	Network Animator
NAV	Network Allocation Vector
NBL	Neighbour Break Link
NFP	Nearest With Forward Progress

NLM Neighbourhood's Location-Matrix

NMEM Neighbourhood Matrix Entries Management

NPN Number of a node's Positive Neighbours

Ns2 Network Simulator 2

NS Node Speed

OTcl Object-Oriented Tool Command Language

PDn Packet Distinction Number

PDR Packet Delivery Ratio

POR Position-based Opportunistic Routing protocol

QoS Quality of Service

RSGF Recovery Strategies with Greedy Failure

REEF REliable and Efficient Forwarding mechanism

RIN Reliability Index

RLT Residual Links Lifetime

RNG Relative Neighbourhood Graph algorithm

RPF Random Progress Forwarding

RSN Reliability Sequence Number of the candidate node

RTF Request To Forward message

RWP Random WayPoint mobility model

SEGF Supportive Enhancement for Greedy Forwarding

SIFS Short Interframe Space

SLPS Self Location Prediction Scheme

Tcl Tool Command Language

TOD TOlerance Deviation distance

TSF Local Timing Synchronization Function

TTL Time To Live

UBM Urgent Beacon Message

VDVH Virtual Destination-based Void Handling

WTSA Waiting Time to Send ACK packet

CHAPTER ONE

INTRODUCTION

1.1 Overview

This thesis proposes a new extension to the current Greedy Forwarding Strategy (GFS) in the Mobile Ad hoc Network (MANET). In this chapter, Section 1.2 provides a general background. Section 1.3 presents the motivation and research problem. Sections 1.4 and 1.5 present the research objectives and the research scope respectively. Sections 1.6 and 1.7 present research assumptions and key research steps respectively. Finally, Section 1.8 presents the organization of the thesis.

1.2 Background

Interest in mobile computing has grown immensely over the last decade. Mobile computing aims to provide users access to information and communication from anywhere and at any time [1]. Mobile Ad Hoc Network (MANET) is a subset of mobile computing [2]. MANET is a spontaneous network because it does not need a pre-fixed infrastructure such as a base station or access points to provide the capacity for communication [3]. MANET is a rapidly deployable, self-organized, multi-hop wireless network that is set up for a limited period of time and for a particular purpose [4].

MANET consists of wireless mobile nodes such as laptops, tablets and personal digital assistants [2]. These mobile nodes may reside in vehicles, instruments and mobile machines, thus, making the network topology highly dynamic [5]. Nodes in MANET may move arbitrarily while communicating over wireless links [3]. In MANET, mobile nodes capable of connecting and communicating with each other use limited-bandwidth radio links. They are incorporated with routing functionality and computational power so that they can perform the operations of host and router simultaneously. Mobile nodes have limited resources including CPU capacity, buffer capacity, and battery power [4]. A schematic illustration of MANET is shown in Figure 1.1 below.

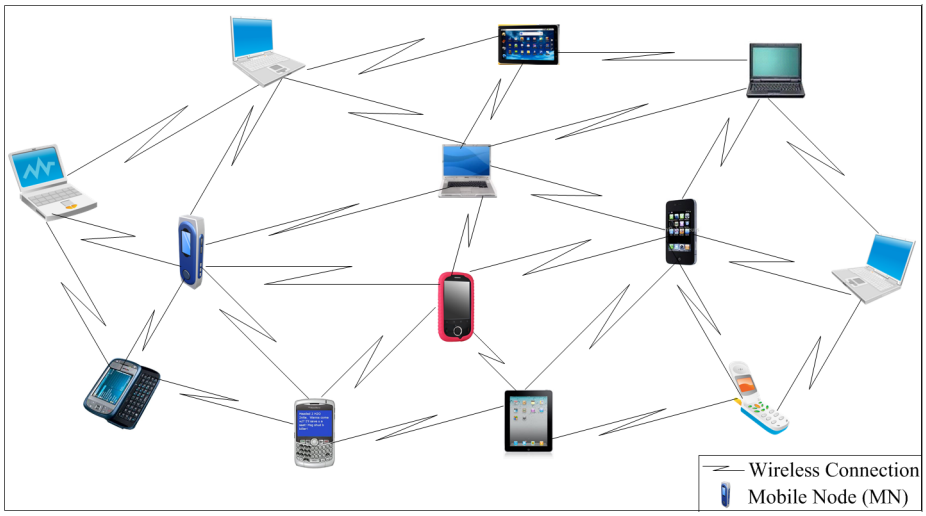


Figure 1.1. An Example of Mobile Ad Hoc Wireless Network Topology

Due to its mobility feature, MANET is considered to be a valuable network for its flexibility and availability for various applications. Also, due to its deployment ease, MANET is considered to be a robust network [3]. However, the flexibility and robustness of MANET come at the cost of more complications and challenges. Besides the features and complexities of traditional wireless networks, the unpredictable and frequent topology changes, resources scarcity, bandwidth constraints, multihop nature, network scalability, and lack of pre-existing infrastructure are new challenges added to MANET [6]. In general, these challenges result in creating serious challenges on the MANET's protocol design and particularly on the routing protocols [6, 7, 8, 9, 10].

1.3 Motivation and Research Problem

To overcome the routing issue, position-based routing protocols have been proposed for MANET. Their main goal is to design a reliable routing protocol that not only can find a route between communicating nodes, but also can respond swiftly to the frequent topology changes, while optimally using limited resources [11, 12, 13]. Early proposals for position-based routing protocols were based on pure greedy strategies [14, 15, 16, 17]. As stated in [18, 19, 20, 21, 22], Greedy Forwarding Strategy (GFS) [23] has been the most-used strategy in position-based routing protocols.

Greedy Forwarding Strategy (GFS) is a hop-by-hop strategy; its algorithm adopts hop counts to achieve the shortest path in fixed interconnection networks [17, 22, 24]. With

a GFS algorithm, a source node forwards packets by applying a geometric calculation to determine the neighbouring node whose distance to the destination is the smallest. Compared to other greedy strategies, GFS incurs relatively low overhead, results in the lowest packet latency, and expends the least energy [21, 24, 25].

The advantage of using GFS [23] in most current position-based routing protocols comes from the operation mode using hop counts criterion by which it achieves the shortest path routing objective [15, 17, 18, 19, 20, 21, 22, 25]. On the down side, a single routing metric, that is hop counts, makes the GFS algorithm vulnerable to repeated failures that, in turn, lead to packet loss [24, 26, 27, 28, 29, 30, 31, 32]. This failure is because routing in MANET requires certain factors for the proper operation of GFS in MANET [6, 8, 9, 10]. These factors mainly include the conditions of the participating nodes, their mobility attributes, and the accuracy of their position information [33, 34, 35, 36]. Ignorance of these factors is the main reason behind the decrease of GFS performance in MANET.

The shortest path adoption makes the GFS algorithm unable to distribute the routing load fairly among mobile nodes [33, 34, 37]. The failure to distribute the routing load fairly creates a hot-spot area at the centre of the MANET that contains the overloaded nodes [38]. The nodes located in the hot-spot area soon create the most congested area [39, 40] with the least battery power [41, 42, 43]. In turn, those highly congested and very low battery power nodes often die fast and create geographical holes in the network [37]. As consequence, a packet sent to a node located in a hole area will be lost [44, 45]. In addition, due to its single routing metric adoption, the GFS algorithm is unable to track the mobility status of the selected next relay-node [27, 46]. Any considerable difference of motion characteristics between a sender and the next relay-node will cause a routed packet to be lost [47, 48]. In other words, if a sender node moves with very high speed compared to the selected next relay-node the link between the two will have a short lifetime and will break quickly, resulting the packet being lost. Therefore, ignorance of node conditions and mobility status during the forwarding decision-making will result in packet loss.

The availability of accurate location information is a critical demand for performing

any position-based routing protocol in MANET. Frequent node mobility causes unpredictable changes in a network's topology [27, 46]. To track topology changes, every node transmits a beacon packet at fixed interval time to its neighbours announcing its latest position information [49, 50]. A node removes a neighbour's entry if the originating node fails to receive a beacon packet in a pre-specified time from that neighbour [51, 52, 53]. In highly dynamic networks, the information stored in a node's neighbours list is often outdated and no longer reflects the actual network topology [46, 51]. The consequence is that, if the selected next relay-node is one of those with an inaccurate position, the forwarded packet will be dropped [47, 48, 54]. These drops lead to retransmissions and rerouting of the dropped packet, which results in increased communication cost. Therefore, a key factor for good performance of GFS in MANET while making a forwarding decision is neighbour location accuracy in a node's neighbours list.

To retain the advantages of position-based routing protocols when using GFS, reducing the GFS failure rate and making GFS compatible with MANET constraints are important. To help achieve these goals, researchers have improved GFS by using two methods [27, 30, 31]. The first method proposed recovery strategies to guarantee delivery when GFS fails. Examples of the method are in [28, 44, 45, 55, 56]. The second method, designed to mitigate GFS failure, includes improvements in GFS itself and using a recovery strategy. Examples of the second method are in [41, 49, 50, 57, 58, 59, 60, 61, 62, 63, 64].

In the first method, when a forwarded packet is stuck at specific node due to a GFS failure, an alternative, but less efficient, strategy is used [65, 66]. This strategy is known as a recovery strategy, a recovery mode, or a rescue mode [67]. While a recovery mode can be used to solve GFS failure and can guarantee delivery, the drawback costs often prohibit the ubiquitous use of a recovery mode as an alternative solution. The simulation results in [68] proved that currently developed recovery modes suffer from trade-offs between (and among) three important factors. First, performing a recovery mode requires more effort by each participating node, which drains available resources. Second, performing a recovery mode creates a complex, long route. A longer route requires additional overheads such as consuming more bandwidth and

power and leads to increased delay due to the added hop count increments. Third, a recovery algorithm handles GFS failure on demand, that is a recovery mode is initiated only when a packet gets stuck leading to more delay.

As stated in [27, 30, 31], the drawbacks costs of a recovery mode have motivated other researchers to develop the second method of solving GFS failure. This method involves improving the next relay-node selection process and improving the beaconing-update approaches. Unfortunately, such solutions inherit the same limitations and weaknesses inherent in the recovery mode method. In mitigating GFS failure, researchers have tried utilizing other routing metrics in addition to that of the hop count. For example, to improve MANET's lifetime researchers have utilized residual battery power of the next relay node. Such improvement cannot achieve the intended enhancement as well as sustain excellent performance for MANET. The reason is because achieving one objective in addition to that of the shortest path may affect other, overlooked objectives [57, 58, 60].

To conclude, motivated by the reasons behind GFS failures in MANET and the drawbacks of current proposed solutions, the demand for an efficient solution is obvious. This research proposes a new extension of the position-based routing protocol based on current GFS. The purpose of this proposal is to create a method to improve the GFS success rate and to make GFS a standalone routing protocol eliminating the need for a recovery mode.

1.4 Research Goal and Objectives

The main goal of this research is to design an extension to the current Greedy Forwarding Strategy (GFS) to transform GFS into a standalone position-based routing protocol for MANET. The intended improvement seeks to make GFS compatible with MANET features, to decrease its failure rate, and to dispense with the need for using a recovery mode. In order to fully accomplish this goal, a set of concrete objectives have been formulated:

i. To propose two new mechanisms capable of improving the GFS's performance in MANET. To achieve the objective, the following specified objectives are identified:

a) To design a beaconing-update mechanism capable of dynamically preserving up-to-date status information of nodes in the network. This mechanism takes into account the mobility's topology changes and the reliability of a node's neighbours-matrix.

b) To design a multi-routing mechanism capable of dynamically controlling outgoing traffic and making the forwarding decision based on multi-metric routing.

ii. To integrate the two proposed mechanisms into the current GFS to mitigate GFS failure and dispense with the need for using a recovery mode.

iii. To empirically validate and verify the correctness of the new algorithms through a series of simulations.

iv. To empirically evaluate the performance of the new routing protocol against the existing position-based routing protocols using Network Simulator 2.

1.5 Research Scope

This research focuses on the unicast routing protocols for MANET. As illustrated in Figure 1.2 below, this research is limited to position-based routing protocols. The position-based routing protocol functions in two different modes: the Greedy Forwarding Strategy (GFS) mode and the recovery mode. This thesis concentrates on the primary mode, which is the GFS mode. This research considers improving the performance of GFS mode to dispense with the necessity of using the recovery mode.

In this particular area, this research focuses on designing an extension to the current GFS. This research considers the reasons behind GFS failure and suggests approaches to solve these failures. The first suggested solution focuses on the periodic beaconing-update that increases the level of the nodes' information accuracy in the network. This solution comes up with two approaches which are a new updated technique and a new neighbourhood entries management technique. The second suggested solution focuses on the next relay-node prioritization and selection process that can assist the nodes in controlling the outgoing traffic and making a reliable routing decision. This solution comes up with two approaches which are a new status information and outgoing traffic management technique and a new multi-routing objectives technique.

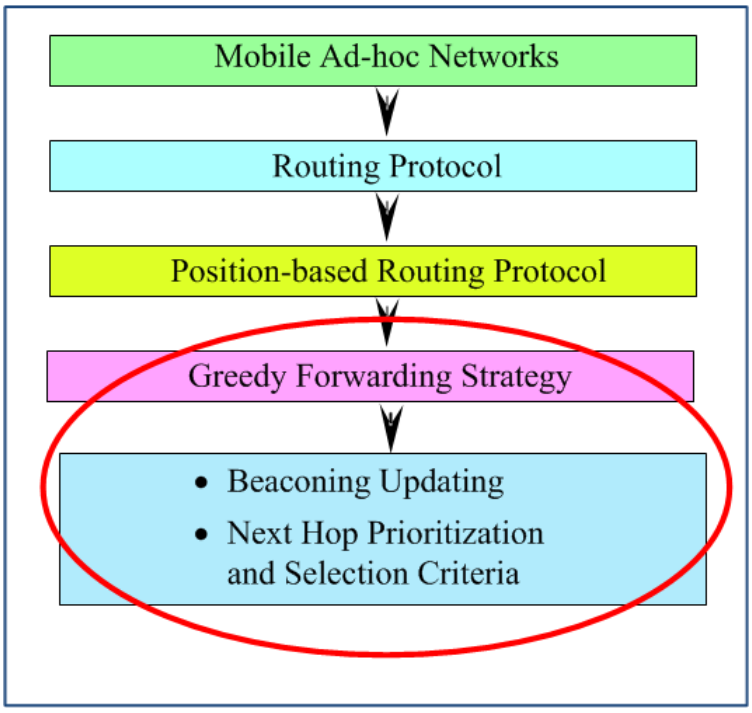


Figure 1.2. Research Scope and Key Contributions (Bounded Area)

1.6 Research Assumptions

This research considers that the only communications paths available are via MANET and that no other communications infrastructure exists. This research assumes that a MANET exists in a two-dimensional (2-D) coordinates plan. In MANET, precision is critical; thus, this research assumes that all clocks in the nodes are synchronized (all nodes maintain a local Timing Synchronization Function (TSF). All nodes have a fixed equal Maximum Transmission Range R (250 m), which is the nominal transmission range of IEEE 802.11. This research uses the IEEE.802.11 Medium Access Control (MAC) Distributed Coordination Function (DCF) protocol. Thus, the link between nodes is bidirectional. This means two neighbouring nodes can send packets to each other.

As in [44, 45], this research assumes that a source node can obtain the position information in (x, y) coordinates, velocity and the motion direction of the destination node by using an available and efficient Location-based Service such as the Hierarchical Location Service (HLS) [69] as suggested by [70]. Also, this research assumes that all mobile nodes are equipped with a Global Positioning System (GPS). Thus, once a

node invokes the localization process, a localization algorithm provides a node with its current status information with a reasonable accuracy. GPS is used to measure a node's own locations in (x, y) coordinates, velocity and motion direction. In using this approach, it is necessary to assum that each node knows its position in (x, y) coordinates, velocity and motion direction with respect to x-axis. This can be accomplished by using a periodic beaconing update.

1.7 Key Research Steps

To achieve the research goal, this research work is carried out through the following steps:

1. Perform two main surveys with the objective of providing a framework of the key concepts upon which to base the design of an architecture that addresses the issue of GFS when adopted in MANET as a forwarding strategy:
    a) Perform an extensive literature review to identify problems with the current Greedy Forwarding Strategy (GFS) when is adopted for use in MANET; and
    b) Perform a comprehensive analysis of previously published work on position-based routing protocols for MANETs, mainly including protocols proposed to mitigate GFS failure and highlighting the issues to be tackled in the proposed protocol.
2. Design a new position-based routing protocol algorithm based on the current Greedy Forwarding Strategy (GFS).
3. Implement the newly proposed position-based routing protocol algorithm in a simulated environment.
4. Perform a performance evaluation of the new algorithm by comparing the new algorithm with existing position-based routing protocols algorithms.

1.8 Thesis Organization

The remainder of this thesis is organized as follow. Chapter 2 presents the background and related work that mainly focus on current GFS and its relevant problems. It also presents the current state-of-the-art approach to solve the greedy forwarding problem. It also presents brief information about the application tools and concepts that are used in this research flow. Chapter 3 provides steps and details of the proposed dynamic beaconing-update mechanism design. In addition, the chapter introduces the implementation of the proposed mechanism. Chapter 4 follows the same procedures as in Chapter 3 for the second proposed mechanism. Hence, it provides steps and details of the proposed dynamic and reactive reliability estimation with selective metrics mechanism design. Finally, Chapter 4 introduces the implementation of the proposed mechanisms. Chapter 5 presents the integration of the two proposed mechanisms given in previous two chapters to shape the new proposed routing protocol. This chapter also discusses the simulation results, provides performance analysis and a detailed performance comparison of the proposed routing protocol with other position-based routing protocols. Chapter 5 also presents the features of the newly proposed protocol based on the simulation results. Chapter 6 concludes the research work, outlines research contributions, points out its shortcomings, and recommends future research work.

CHAPTER TWO

BACKGROUND AND RELATED WORK

2.1 Overview

Routing is one of the key challenges in the design of Mobile Ad Hoc Network (MANET) [6, 7, 8, 9, 10]. Greedy Forwarding Strategy (GFS) [23], which is proposed to be used in static networks is the most famous strategy adopted with position-based routing protocols in MANET [18, 19, 20, 21, 22]. GFS was not used until the very early 2000's. Since this date and up to now, several studies have been conducted to show the reasons behind GFS failure in MANET. A number of proposed solutions also have been put forward as potential approaches towards overcoming the identified problems in conventional GFS [27, 30, 31]. As shown in Figure 2.1 below, this chapter is directed towards capturing a set of potential enhancements to conventional GFS. The goal behind that potential enhancement is to shape the basics of this conceptual research framework that target improving GFS to be compatible with the MANET environment.

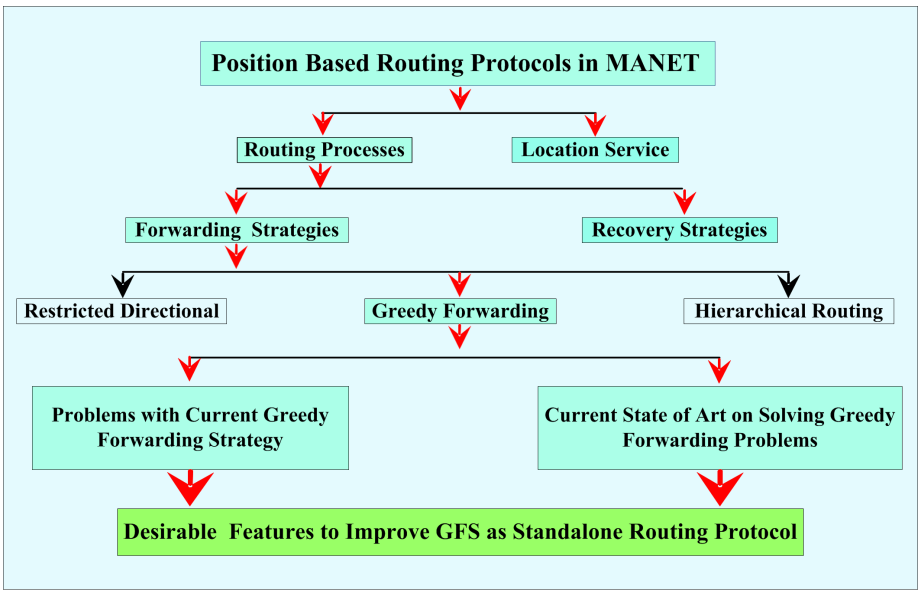


Figure 2.1. Literature Review Framework

This chapter is organized as follows: An introduction and brief knowledge of routing protocols in MANET is presented in Section 2.2. Position-based routing protocols' and packet forwarding strategies are presented in Sections 2.3 and 2.4 respectively.

Section 2.5 investigates the reasons behind GFS failure when used in MANET. Section 2.6 investigates in previous works that have been done to solve GFS problems when used in MANET. Section 2.7 presents a qualitative discussion of the proposed routing protocols to alleviate GFS failure and the direction of this research. Section 2.8 presents the research application tools and concepts. The chapter's summary is presented in Section 2.9.

2.2 Routing in Mobile Ad Hoc Networks

Generally, the function of a routing protocol involves the process of exchanging packets between the communicating nodes of the network [11]. As stated in [71], a reliable and efficient routing protocol should has two main distinctive characteristics. First, the routing protocol algorithm should has the ability to establish correctly a route between a pair of nodes, so a packet can be delivered in a timely manner. Second, the established route should be found with a minimum consumption of both overhead and bandwidth.

In MANET, a routing process runs on every node and is affected by its resources and mobility degree [11]. Due to the limited radio transmission range of wireless devices, nodes can communicate directly if they are within transmission range of each other. Otherwise, they communicate indirectly by using intermediate nodes [72]. Thus, the route in MANET is usually formed from a sequence of intermediate nodes. These nodes collaborate to transfer the data packet from the source node to the targeted destination node [71]. Due to node mobility, the links between intermediate nodes suffer frequent failure and reactivation. As a consequence, the constructed communication route between a source and a targeted destination node has to change frequently. Frequent route changes cause significant number of routing packets to be lost, leading to retransmission and rerouting the lost packet that increase communication cost. Therefore, routing has been recognized as the fundamental research problem in MANET [10].

In MANET, a need exists for an appropriate dynamic multi-hops routing protocol to facilitate communication between the participating nodes. The goal is to ensure that

the routing protocol is able to facilitate efficiency and scalability with both sparse and dense MANET. When in such mode, the protocol can make fast response to the frequent topology change as well as adapt to MANET resource constraints [12, 73]. In accordance with this logic of its efficiency, the used routing algorithm is considered to be a sign of the performance and reliability of mobile nodes communications [71, 72, 73].

Recent years have seen a burst of proposed routing protocols for MANET, which can be divided broadly into Position-based and Topology-based routing protocols [11, 12, 13, 74].

Topology-based routing protocols need information about links that exist in the network. Such information is used to establish and maintain routs to perform packet forwarding [75]. These protocols discover routes either proactively such as the work presented in [76] or reactively such as the work presented [77]. On one hand, proactive route discovery cannot maintain accurate routes in a highly mobility's topology change and has high overhead. On the other hand, reactive route discovery relies on flooding. Flooding can congest the network, has high delay, and does not scale to large MANET [78, 79, 80].

Position-based routing is the task of delivering a data packet to a specific position in a MANET [21]. These protocols utilize the geographical location information of participating nodes to deliver packets over a network. Thus, in all proposed position-based routing protocols, the underlying principle is to send the data packet ahead from sender node to a neighbour within its transmission range that is closer to the destination than itself. With this principle, they make positive progress towards the destination [20]. The Greedy Perimeter Stateless Routing (GPSR) [44] protocol is a prominent example of this category.

The simulation results in [81, 82] showed that position-based routing protocols scale better than other routing protocols in MANET. Additionally, the simulation results showed that position-based routing protocols offer a number of advantages over topology-based routing protocols. Also, as stated in [83], position-based routing algorithms eliminate some drawbacks of topology-based routing protocols.

Because the scope of this research is concerned with position-based routing protocols, this research does not address any topology-based routing protocols for MANET. A survey of topology-based routing protocols can be found in [75]. The following section present more details about position-based routing protocols.

2.3 Position-based Routing Protocols' Structure and Operations

The overall idea of position-based routing is to select the next relay-node from a node's neighbours-list, which is the closest to the destination, other than itself. To make such a routing decision, every node in MANET must have local knowledge of its neighbourhood location information. The principal goal of position-based routing algorithms is to ensure that a short path can be found if one exists between two communicating nodes whose locations are known [14, 16, 17, 18].

A position-based routing protocol is constructed of two main elements: a location service and a geographic routing process [20, 21, 24]. The location service is used to determine the location of the packet's ultimate destination. The geographic routing process is used to discover and establish routes between communicating nodes in the network. The next sub-section presents a brief review of position information and location services methods used in position-based routing protocol.

2.3.1 Position Information and Location Services Methods

The availability of location information is the main issue for performing any position-based routing protocol in MANET. Thus, the ability of the sender node to make effective routing decisions is based totally on the following factors [15].

i. The ability to obtain its own position information;
ii. The ability to maintain up-to-date information of its immediate neighbours; and
iii. The ability to know the targeted destination position information.

A node's self-position information can be obtained via Global Positioning System (GPS) [84], or by using location-sensing techniques as in [85]. After it gets its own location information, a node emits an update packet in order to inform all neighbours

of its presence and position information. The node, which receives the update packet, will update its neighbours-list regarding the node sent the update packet [49, 50].

Position-based routing protocols always need the location information knowledge about the targeted node. The determination of the position of a node in the network is the responsibility of the Location-Service, which dynamically maps a node's ID to its geographical information. In other words, Location-Service is responsible for answering a node's location query such as: "Where is the node with id X?". Because of this, a major challenge in position-based routing protocols has been the design of scalable and effective location services to track the geographical locations of the participating nodes and quickly respond to their location queries [70].

Broadly speaking, existing location services in MANET can be classified into two main categories which are: Flooding-based and Location-server Based [70]. The first category comprises two classes: reactive flooding-based and proactive flooding-based location services [86]. In the proactive class, every node periodically floods its geographic information to all other nodes in the network. Thus, all nodes are able to update their location tables. The flooding sending rate can be optimized according to the distance effect or node's mobility [87]. DREAM Location Service (DLS) is an example of the proactive flooding-based location service in the Distance Routing Effect Algorithm for Mobility (DREAM) [88] routing protocol. In the reactive class, a node sends its location information when it receives a location request. Reactive Location Service (RLS) [89] is an example of the reactive flooding-based location service. Proactive flooding-based incurs heavy loads because of using periodical flooding update locations. Additionally, if a nodes' mobility is considered to determine the interval of flooding sending rate, a broadcast storm problem can be happen in the case of high nodes' mobility. Thus, proactive class is inefficient in terms of scalability and risks of congestion. Although reactive class mitigates the overhead of useless location information in proactive class it adds high latency that is not suitable in case of high mobility in MANET [70, 86, 87].

In the location-server based method, the nodes in the network agree on mapping of a node's ID to other specific nodes [70]. The elected nodes are called the "location

servers". Selected nodes maintain the received nodes' location information for later use, as on demand. When a node wishes to send a data packet, it queries a location server node to get the location information of a targeted destination [87]. The location server will replay as soon as it receives this request. In the location-server-based category, there are two major classes of methods: the quorum-based and hierarchical-based methods.

In the quorum-based method, the location update is sent to a group of nodes called the update quorum. Later, the location query is sent to a different group called the query quorum. The two groups are designed such that they are not necessarily disjointed, and, thus, some node in the update quorum will satisfy the query. Examples of this class are the methods discussed in [90].

In the hierarchical approach, the network is divided into smaller and smaller levels. For a node, in each level of the hierarchy, one or more nodes are selected as its location servers. Location updates from a node and its queries traverse up and down the hierarchy. Hierarchical Location Service (HLS) [69] and the Grid Location Service (GLS) [91] are two prominent examples of the hierarchical approach. Using the hierarchical approach avoids flooding. Also, a major benefit of using a hierarchy approach is that, when the requester and destination nodes are nearby, the forwarded packets to the lower levels of the hierarchy are limited. Although, hierarchical approaches reduce their overhead by taking advantage of localized mobility, in case of global communication, the queries have to traverse high up in the hierarchy. Faraway location servers may suffer from inconsistencies that degrade the performance of the hierarchical approach and the routing protocol itself [92]. However, as mentioned in [93], from amongst the current location service techniques, HLS [69] technique is the most suitable candidate to be used in an ad hoc network under high mobility conditions.

2.3.2 Geographic Routing Process

The second main part of position-based routing protocol involves a routing process. It can be further subdivided into the basic forwarding algorithm (primary mode) and a secondary repair strategy (recovery mode) [24]. For the sake of this research flow, the

former mode is introduced in the next section and the later in Sub-section 2.6.1.

2.4 Packet Forwarding Strategies (Primary Mode)

Geographic Forwarding can be defined as the process involved in making the routing decision locally at each node [24]. Making the forwarding decision by a sender node is primarily based on two main factors [18, 30]. The first one concerns accurate knowledge of intermediate nodes' location, and the position of the ultimate target. The second is that the sender will be concerned about the forwarding method that will be used to select the next relay-node. These packet-forwarding strategies typically can be categorized into: Restricted Directional Flooding such as the works presented in [88, 94], Hierarchical Approaches such as the work presented in [95] and Greedy Forwarding Strategies such as the work presented in [23].

Because the scope of this research is concerned with Greedy Forwarding Strategies, this research does not address any Restricted Directional Flooding or Hierarchical Approaches for MANET. The following section present more details about Greedy Forwarding Strategies. A survey of Restricted Directional Flooding and Hierarchical Approaches can be found in [24, 32].

2.4.1 Greedy Forwarding Strategies

Greedy forwarding strategies are the most popular unicast position-based routing protocols in MANET [15, 24]. In the greedy forwarding strategies, a node tries to forward the packet to a one one-hop neighbour closer to the destination than itself [17, 21, 25]. To determine the next-hop to be selected, the greedy forwarding strategies use different optimization criteria. A node can use one of those approaches to decide which of the neighbours for a given packet should be forwarded [16, 22, 26]. As illustrated in Figure 2.2 below, certain optimization criteria can be identified as follows [24]: Progress-based, such as the Most Forward within Transmission Range (MFR) [96], and the Random Progress Forwarding (RPF) [97], Direction-based such as the Compass Routing (CR) [98], and Distance-based such as the Nearest With Forward Progress (NFP) [99], and GFS [23].

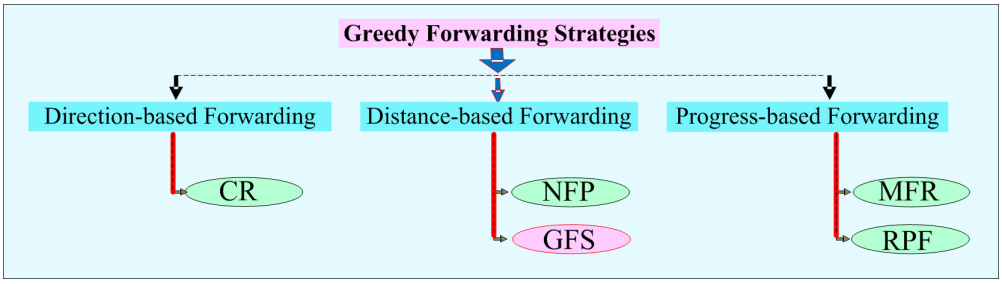


Figure 2.2. Main Greedy Forwarding Strategies Used in Position-based Routing Protocols

2.4.1.1 Most Forward within Transmission Range

In [96], Takagi and Kleinrock proposed the Most Forward within Radius (MFR). MFR belongs to the progress-based strategy. With the algorithm criterion, a sender node makes its forwarding decision by selecting a neighbour which possesses the shortest Euclidean distance to the destination. This strategy tries to minimize the number of hops a packet has to traverse to reach the destination. Compared to the other criteria, MFR improves energy consumption but incurs more interference [16, 22, 26].

2.4.1.2 Random Progress Forwarding

In [97], Nelson and Kleinrock proposed the Random Progress Forwarding (RPF) method. RPF enables the sender to randomly choose one of the neighbour nodes that is closer to the destination than itself as the next-relay node. By means of this approach, RPF has the advantage of distributing the traffic load. In comparison to the other geographical forwarding strategies, this approach does not use any measure of progress to differentiate any single candidate's next hop as better than the other. In effect, this strategy minimizes the accuracy of information needed about the position of the neighbours as well as reduces the number of operations required to forward a packet [16, 22, 26].

2.4.1.3 Compass Routing

Kranakis et al. proposed the Compass Routing (CR) algorithm in [98]. The CR algorithm belongs to the direction-based strategy. In CR, a source makes its decision to forward a packet towards its destination, by selecting a neighbour that has the minimum angle between the imaginary lines from the source to the neighbour, with the line

from the source to the destination. This CR approach tries to minimize, as much as possible, the spatial distance that the packet travels. Although compared to the other criteria this approach reduces the energy consumption, it incurs more end-to-end delay [16, 22, 26].

2.4.1.4 Nearest with Forward Progress

In [99], Hou and Li proposed the Nearest with Forward Progress (NFP) algorithm. NFP criterion belongs to the distance-based approach. In this approach, a sender node selects the next relay node that has the minimum distance to the destination from amongst other neighbours. As a result, the NFP routing protocol is designed to ensure that it stays within a selected route as long as it exists. This strategy is meant to decrease the energy consumption and the bandwidth usage for each hop. However, NFP criterion incurs more end-to-end delay compared to other approaches [16, 22, 26].

2.4.1.5 Greedy Forwarding Strategy

Because this research is concerned with GFS, the functionality of this criterion is covered comprehensively in this section. Finn's GFS that belongs to distance-based category is considered as one of the earliest proposed criteria [16, 22, 26]. The current GFS algorithm [23] considers the shortest path as a measure of route cost to make routing decisions. To satisfy this objective, the sender node selects the next relay neighbour as the farthest node from it and the closest to the destination. Figure 2.3 below shows the case of GFS when establishing the route between the mobile nodes.

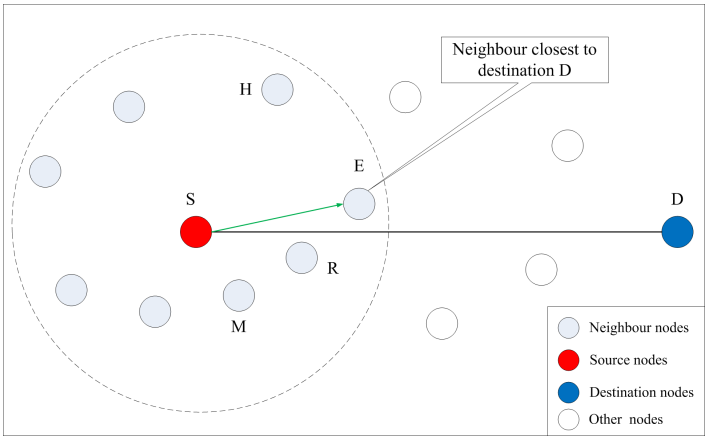


Figure 2.3. The GFS Criterion Functionality

The process in the criteria is as follows. Node *S* has data to be forwarded to node *D*. Node *S* has a group of neighbours, *E*, *H*, *M*, and *R* in the direction of the target. One function of the node, as it applies to GFS algorithm, is to compare distance between the neighbours, *E*, *H*, *M*, and *R*, and the destination node *D* and decide which is the closest neighbour to the destination. According to the distance condition (the shortest path), node *E* is the closest node to the destination *D*. Hence, node *S* selects *E* as the optimal next-relay node amongst other neighbours.

2.4.2 Advantages and Disadvantages of GFS over Others Criteria

Current study in [18, 19, 20, 21, 22] has shown that GFS is the top-most means to route data packets in MANET. With GFS, the sender node forwards the data packets to the neighbour who is closer to the destination than itself.

Its practice is favoured because GFS makes each forwarding decision per hop per packet. This feature makes the GFS algorithm more robust. This is demonstrated by the fact that, even when the failure of a node may cause the data packet in-transit to be lost, its robustness will ensure that there will be no need to set up a new route [14, 16, 22, 25, 26].

Because GFS adopts hop count as the only deciding factor in the selection process, GFS provides a simple technique for routing in MANET. The choice of GFS to be used in most current position-based routing protocols reflects the acknowledgement of certain mertis. First, GFS does not influence overhead costs involving building, maintaining and distributing the distance vector. Second, GFS does not incur link state routing tables, or the control overhead and latency of route discovery, which reactive topology-based routing protocols often include. Finally, in dense MANET, the GFS algorithm has high delivery rates. On the other hand, it has low delivery rates for sparse MANET. For both, dense and sparse network, the GFS method does not guarantee delivery with nodes in mobile. The situation becomes worst in a high mobility's topology change. These drawbacks prove the low performance of GFS in MANET. This is because GFS was proposed for use in static networks not in mobile enveronment.

2.5 Reasons behind GFS Failure in MANET

Typically, the performance and reliability of nodes' communication are much related to routing protocol efficiency [71, 73]. The power of GFS to route data using only distance metrics comes with several attendant drawbacks in the implementation phase [100]. Because certain factors are needed for the proper operation of this function are not often considered, the GFS selection process is unreliable. These factors mainly include the conditions of the participating nodes, their mobility attributes, and the accuracy of their position information [33, 34, 35, 36]. The ignorance of these factors is the main reason behind GFS failure that leads to decreased its performance in MANET.

2.5.1 The Effects of Participating Nodes' Conditions

Greedy forwarding strategy GFS, based on the shortest path, cannot fairly distribute the routing load among mobile nodes [33, 34]. As shown in Figure 2.4 below, using the shortest path results in a network hot-spots problem [38]. The hot-spot area is more prone to high traffic load than other places in the network.

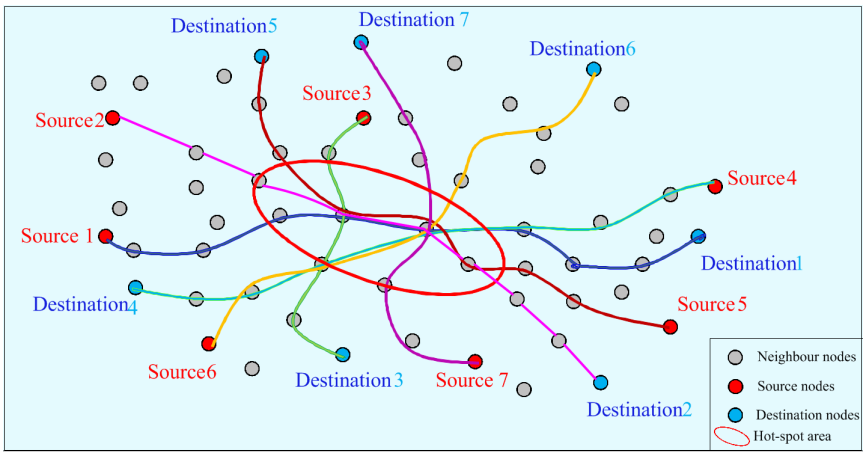


Figure 2.4. A Hot-spot Problem over GFS Criterion in MANET

Hot-spot problem incurs several drawbacks that influance the performance of MANET [33, 34, 35, 36]. First, the packets' collision probability is increased dramatically, that results in more packets to be dropped. Second, the nodes located at the hot-spot area become the most congested nodes of MANET, because those nodes need to forward packets more often than others. Third, the same nodes suffer from more use of battery

power, so their lifetime is decreased very fast. Any destined packet to a node located in the hot-spot area will be lost or at best case will suffer from long delay to be proceeded.

Current GFS do not consider the conditions of the next relay node in their selection process. Consequently, a shortest path sometimes incurs a higher packet loss, higher end-to-end delay, and less packet delivery than some other available routes [34]. The conditions of the next relay node that might incur a hot-spot problem are summarized in the following sub-sections.

2.5.1.1 Congestion Problem

The analysis of MANET using GFS reveals that its traffic load is not evenly distributed [101]. Furthermore, in congested traffic areas (hot-spot areas), the data packets may be dropped due to the fixed length of interface queues and excessive delay caused by the waiting time in the queue. Current design of GFS is not congestion-adaptive [102], and it may allow a congestion to happen. The consequences of these conditions become severe in larg-scale transmission of heavy traffic, where congestion is more probable. Accordingly, this results in degrading of the performance of position-based routing protocol [101, 102]. As an outcome, to increase the performance of GFS scheme in MANET, it is preferable to consider efficiently the neighbours' congestion level as another metric besides the distance metric to make the forwarding decisions.

2.5.1.2 Battery Power Problem

In MANET, each mobile device has limited battery power capacity; hence, its processing power is limited [41, 42]. The objective of deploying MANET is achievable as long as it is considered alive [43]. There are two factors that determine the lifetime of a mobile node in MANET. The first one is the quantity of energy a mobile node consumes over time. The second one is the residual amount of energy that is available for a mobile node use [103].

As a matter of fact, forwarding packets using the shortest route frequently lead to unfair power consumption of the nodes in the centre area of the network, yet other places of MANET are usually distant from any power consumption [103]. Current

design of GFS is a non-energy aware routing protocol, and it may lead to minimizing the whole lifetime of MANET, as well as preventing MANET's deployment goals to be achieved. As an outcome, to increase the performance of GFS in MANET, it is desirable to efficiently consider the neighbour's battery power-level as another metric besides the distance metric to make the forwarding decisions.

2.5.1.3 Next-Relay Node Positive Degree Problem

The node degree of a next-relay node is defined as the number of neighbouring nodes with a direct link to this node [104]. It can also be defined as the number of links a node has in the general direction of the destination. The frequent variation in the node degree occurs for two main reasons. The first one refers to those nodes that come out of battery power as mentioned in the previous section; the neighbours become out of range of each other due to the variation in the motion values of both (speed and direction). The other reason is attributable to the sudden disappearance of some nodes from MANET, while others simply enter. In both cases, the consequence is a variation in the node degree [105].

Current design of GFS does not consider the nodes' neighbours' degree. The situation is that the selected next relay node may or may not have neighbours in the general direction of the destination. The implication of the selected node not having neighbours is that the forwarded packet reaches a dead-end (or void) leading to a higher packet loss, and non-optimal route. It will also attract higher end-to-end delay, and least packet delivery than some other available routes [104, 105]. In conclusion, to increase the performance of GFS in MANET, it is preferable to efficiently employ the neighbours' positive degree as another metric besides the distance metric to make the forwarding decisions.

2.5.1.4 Local Maximum Problem

Local maximum is one well-known problem that limits the performance of GFS [65]. Figure 2.5 below illustrates this case, in which the routed packet ends in a void. In the figure, node *S* has a data packet to be forwarded to node *D*. Node *S* has a group of neighbours, *A*, *B*, and *N*, located in the direction of the destination. Next, node *S* starts

to calculate the distance of each neighbour to select the closest one to the destination. Node *S* executes the estimation based on the latest received beacon packet. In this scenario, both nodes, *A* and *B*, have neighbours in the direction of the destination, but node *N* does not have any neighbour.

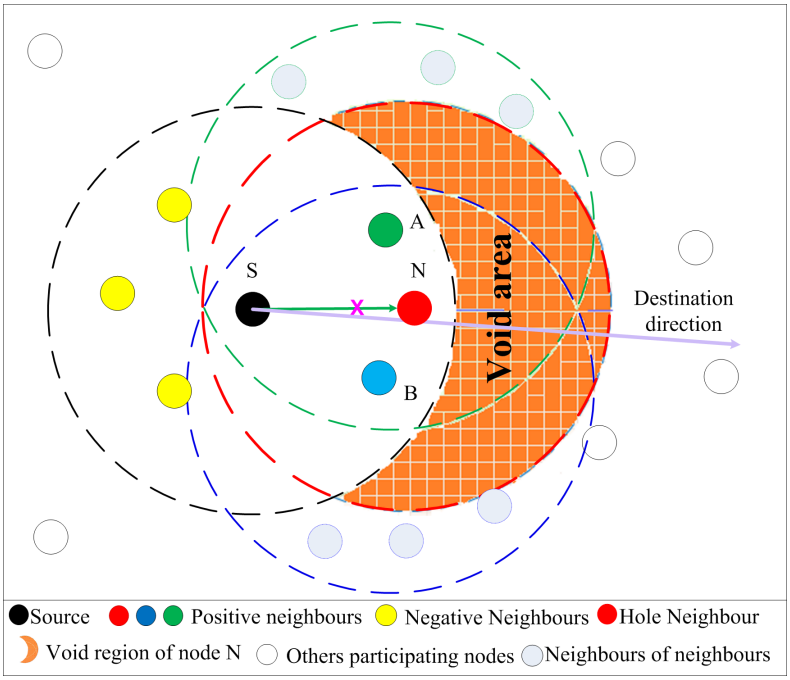


Figure 2.5. Local Maximum Problem

Going back to node *S*, it is found that based on its calculations that node *N* is the appropriate next hop, when, in fact, this is an incorrect choice based on the neighbour's degree. Node *A*, or *B*, at this time are the correct choice. Next, node *S* transmits the data packet to *N*, but node *N* has not positive neighbours in the direction of node *D*, thus, it cannot communicate with node *D* using GFS. In this case, the packet is said to have encountered a communications void with respect to the destination *D* and gets stuck. Node *N* is called a void node while the shaded region, without any nodes inside, is called a void area. This phenomenon is well known as the local maximum problem [28].

To solve the local maximum problem, one of the following possibilities can be used: Some studies on geographic routing as argued in [27], say that node *N* has to buffer (carry the packet and move) until it has new neighbours in the direction of the destination instead of dropping the stuck packet. Otherwise, node *N*, as argued in [28], will apply a recovery mode to forward the packet to the destination *D*. Consequently, oc-

currence of local maxima problems at forwarder node incur a higher end-to-end delay, or at best case, increase the number of hops that the stuck packet has to traverse until it arrives at target (non-optimal path) [65, 66, 67, 68]. The occurrence of the local maximum problem, while performing greedy routing, enhances the value of the solution suggested in the previous section, which is to employ the neighbours' positive degree as another metric besides the distance metric to make the forwarding decisions.

2.5.2 The Effects of Mobility Attributes

Mobility metrics affect positively or negatively the performance of the used routing protocols in MANET [47]. The mobility's topology changes increase the rate of link break, and creation of new links, which increase the volume of control traffic that is required to establish and maintain the new routes. Among these metrics as depicted in Figure 2.6 below are speed, motion direction, link stability, and beacon interval time, as classified in [48]. This section explores the effect of these mobility metrics on the performance of GFS in position-based routing protocols.

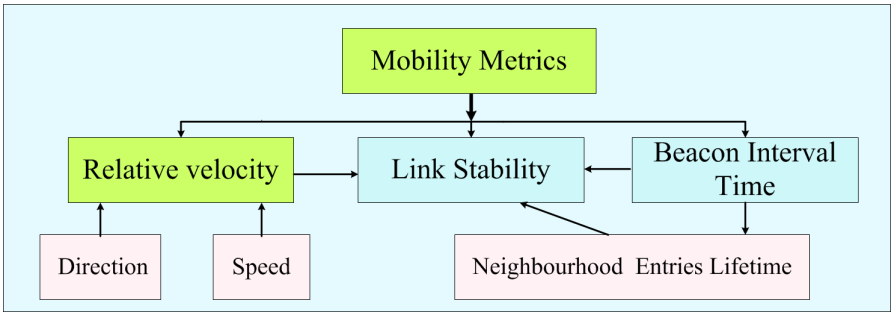


Figure 2.6. Mobility Metrics

2.5.2.1 Movement Direction Problem

The counter movement direction of the next relay node results in a non-optimal path problem [48]. Not considering this factor in the selection process sometimes makes the sender node select the neighbour that has the opposite direction movement toward the destination [54]. Figure 2.7 below illustrates this problem.

In Figure 2.7, source node *S* has data packet to be forwarded to node *D*. Node *S* looks in its neighbours-list, searches for its neighbours located in the direction of the target, and finds that the neighbours *M* and *N*, satisfy the demand. Next, node *S* starts to calculate the distance of each neighbour to select the closest to the destination. Node

S executes this estimation based on the latest received beacons at the time t_1. The neighbours' locations with those beacons are LM_1 and LN_1 respectively.

Node M and N move in a different direction, at time t_2, at which sender node wishes to send a packet, they will be in location LM_2 and LN_2 respectively. Going back to node S, it is found that based on its estimation, node N is the appropriate next hop, when, in fact, this is a wrong choice based on the movement direction of nodes M, and N; at this time, node M is the right choice. This situation leads to transmit the data packet from S to N at location LN_1. This results in packet loss, or in the best case, increases the number of hops that the packet follows until it arrives at target (non-optimal path).

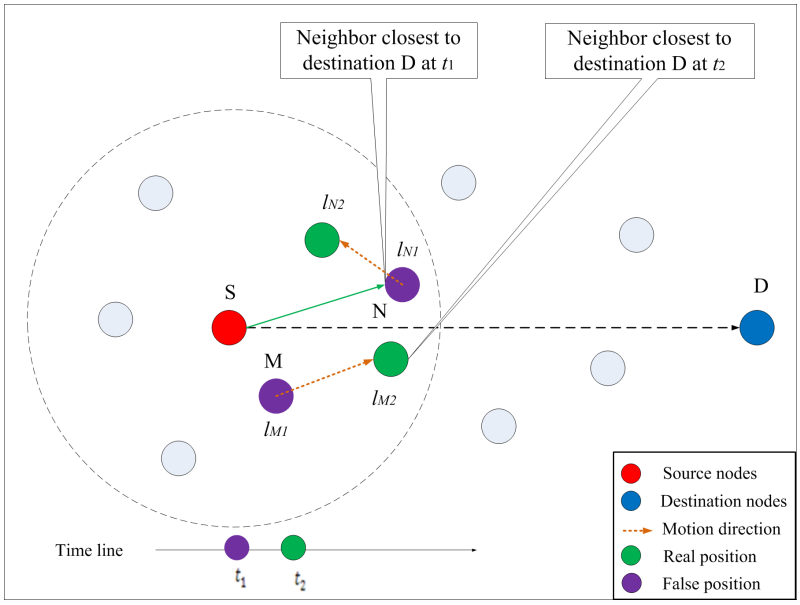


Figure 2.7. Effect of Neighbours' Movement Direction Problem

In the light of this scenario, to increase the performance of GFS, it is preferable to utilize the motion direction of participating nodes as another metric besides the distance metric to make the forwarding decisions.

2.5.2.2 Movement Speed Problem

Nodes in MANET can move in any direction with any speed at any time they wish. This arbitary and unpredictable motion influences the quality and stability of the link between participant nodes [106, 107]. In addition, node mobility results in changes of the position of the node with respect to other nodes. Figure 2.8 below depicts this problem in which selecting a node with inaccurate position information as next relay

node results in the forwarded packet being lost.

In Figure 2.8, source node S has a data packet to be forwarded to node D. Source node S looks in its neighbours-list, searching for its neighbours located in the direction of the target, and finds that the neighbours M and N, satisfy the demand. Next, node S starts to calculate the distance of each neighbour to select the closest one to the destination. Node S executes this estimation based on the latest received beacons at the time t_1. The neighbour's locations with the latest received beacons are LM_1, and LN_1.

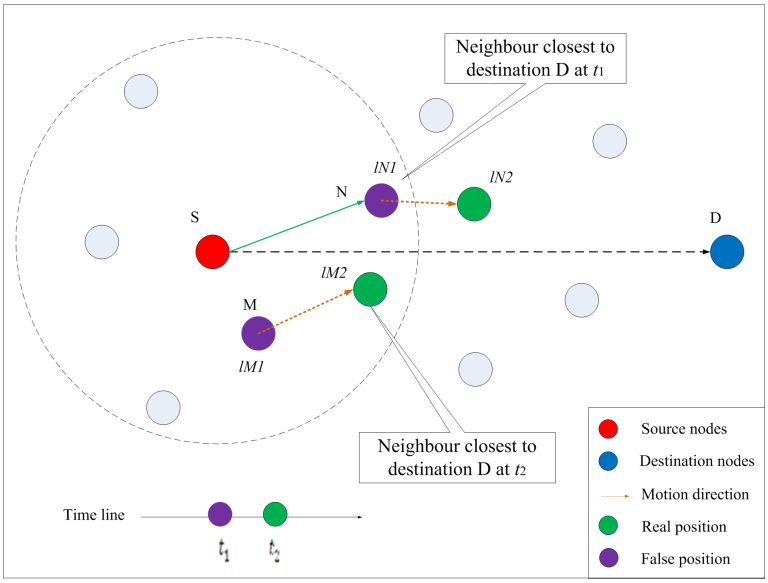


Figure 2.8. Effect of Neighbouring Nodes Speed

The speed of node M and N are different with respect to node S. Later, at time t_2, when node S wishes to send data to the destination, Nodes M and N are supposed to be at location LM_2 and LN_2 up to their speed. Going back to node S, based on its estimation, node S found that node N is the appropriate next hop, when, in fact, this is an incorrect choice based on the speed of node N; it becomes out of the transmission range of node S. Meanwhile, node M is still within its transmission range, so, at this time, node M is the right choice. Therefore, the wrong selection results in packet loss again [54]. Speed of next-relay nodes is not considered with a GFS algorithm. To sum up, to increase the performance of GFS, using nodes speed as another metric besides the distance metric to make the forwarding decisions is desirable.

2.5.2.3 Link Lifetime Problem

The link lifetime is typically reduced with the increment of relative velocity difference between nodes [108]. As shown in Figure 2.9 below,, with GFS, nodes tend to select the long links (node at LN_1), to reach the particular destination in a minimum number of hops.

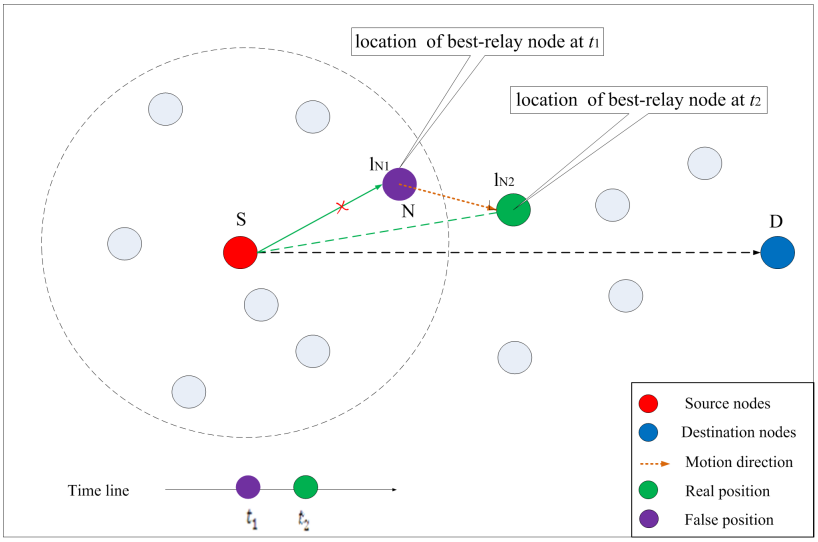


Figure 2.9. Unstable Link Life-time Problem

As argued in [109] a long link is always has the least lifetime, which results in more links break occurring. This problem is called Neighbour Break Link (NBL). NBL usually involves expensive packet loss, more routing overheads and more energy consumption. Thus, the long links are less reliable, which causes MANET's performance degradation.

There are three major reasons behind the NBL problem [107]. The first reason is the using of long beacon interval time to send beacon packet, and the second reason, is the differences in relative speed between a sender node and its neighbours. Third, and by far the more serious issue, is the nature of next hop selection process itself. With a GFS algorithm, a data packet is routed to a neighbour whose distance to the destination node is smaller. Such a neighbour is located mainly at the edge of a sender's transmission range.

As shown in Figure 2.9 above, when node *S* wants to transmit a packet to node *D*, it looks up in its neighbours-list searches for a neighbour that is closer to node *D*, i.e. the

destination. Based on its geometric calculation, node *S* finds that node *N* is the closest the choices. In fact, such a neighbour, i.e., node *N*, may depart from *S*'s transmission range very quickly because it was very close to the edge of node S's transmission range at time t_1. Thus, the chosen next relay-node, in fact, does not exist within the sender's transmission range at time t_2 while it is still listed as a neighbour in *S*'s neighbours-list. This situation is reason behind most NBL problems.

To sum up, link lifetime between the communicating nodes was not considered with GFS. Thus, GFS fail due to the above-mentioned reasons. Consequently, to increase the performance of data routing in MANET, it is preferable to use the link with the highest stability as another metric besides the distance metric to make the forwarding decisions.

2.5.2.4 Beacon Packet Interval Time Problem

Each node in MANET periodically emits a beacon packet (BP) to advertise its presence in the network to its neighbours within its transmission range. The BP message holds the node identity (*ID*) and its present position information in (x, y) coordinates. The node receives the BP message from its neighbour, edits the new entry or refreshes the existing entry of this neighbour in its neighbours-list. The beacon packet interval time (BPIT) determines the maximum time interval between two sequential emitted beacon messages of the participating nodes [49].

To get more accurate location information, BP must be repeated with enough frequency. Solutions to this problem must balance the need to get accurate location information and sending data packets with the cost of carrying out beaconing.

In a MANET environment, task to attempt to determine the optimal beacon interval is challenging. The reason is that, if BPIT is smaller, as compared to the rate at which a node changes its current position then its position information in its neighbours neighbours-list will be more accurate. The increase in the accuracy ratio leads to high communication overheads that will be at the expense of network resources. Moreover, an increment in the beacon-sending rate also leads to increase in the collision probability. which results in more packet loss. Consequently, the overall perfor-

mance of the system will be affected [50].

On the down side, very long BPIT impacts negatively on the accuracy of node position information in the neighbours-list, which affects the packet delivery ratio and throughput [51]. Consequently, with MANET, there is the challenge of optimal performance because none of the suggested beacon interval time schemes mentioned above perform as well as they are supposed to, to be optimally used with MANET [52].

A schematic illustration of BPIT problem in MANET is shown in Figure 2.10 below. In this figure, node S receives the latest beacon from node N at time t_1, and reports its location in LN_1. Later, node S receives a data packet to be forwarded to ultimate destination at time t_2. Node S looks in its neighbours-list, searching for its neighbours located in the direction of the target. Node S, based on the latest received beacon at time t_1, finds that neighbour N at location LN_1 satisfies the demand as a closest neighbour to the target. Node S will forward the packet to N even though N is at location LN_2, and out of S' transmission range, which will cause the packet to be dropped.

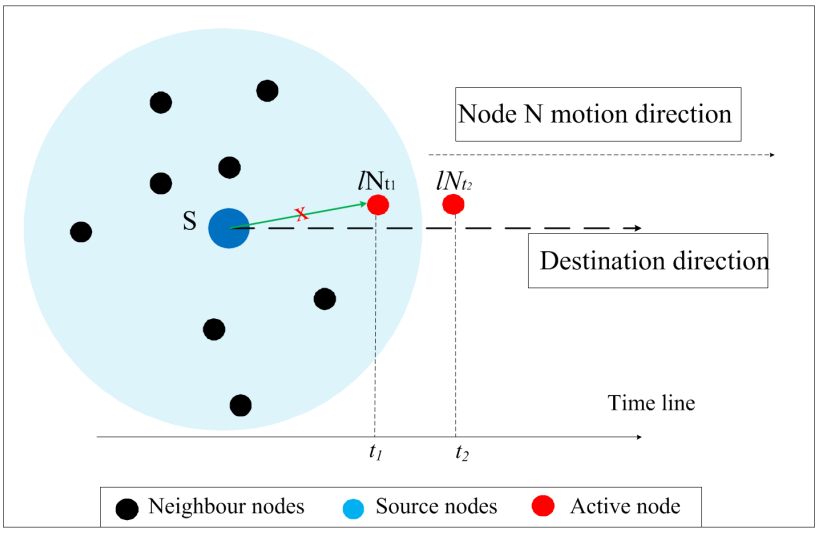


Figure 2.10. NBL and BPIT Problems Combination in GFS

From the above discussion, the conclusion can be made that BPIT interval is inversely proportional to position information accuracy of the neighbourhood in a node's neighbours-list. However, whatever \ way is used to get location information, the belief is that the advantages of using accurate location information outweigh the additional costs [51]. Beacon interval time was not considered with GFS algorithms. Thus, GFS

may fail due to the above-mentioned reasons. Consequently, to increase the performance of GFS routing in MANET, it is desirable to adopt the beacon packet interval time as another metric besides the distance metric to make the forwarding decisions. This PBIT should be able to provide a highest accuracy information of participating nodes with minimal overhead.

2.5.2.5 Neighbourhood Entry Lifetime Problem

In position-based routing protocols, to forward a packet, each node chooses a next relay-node from its neighbours-list. An inaccurate list leads to a wrong selection decision, which can have devastating consequences on MANET's resources. Thus, the freshness of the entries in the node's neighbours-list is in a high demand for position-based routing protocol [50].

A node joining a network should be able to be self-organized. As it joins a MANET, it has to announce its presence by emitting a BP to all of its neighbours within its transmission range. Also, it should start building its own neighbours-list to efficiently communicate with the others. The building of a node's neighbours-list is totally dependent on the frequency of received BP from the neighbourhood. The entries of the neighbours-list are checked periodically by a node to ensure that it does not contain stale entries.

In the literature, as in [44, 45], the Neighbourhood Entry Lifetime (ELT) (i.e., entry timeout) is always fixed to a pre-specified threshold. Such pre-deterministic fixing is unfair, because a node's neighbours-list might always contain out-dated entries [52]. To give more clarity to this claim, let us assume the following two scenarios. In the first one, with a small threshold, if a node transmits its updated beacon to its neighbour and BP could not reach a neighbour's node for any reason (say due to congestion problem), then the neighbour will remove this node's entry from its neighbours-list, while in fact, the node is still in its transmission range.

The second scenario explains the long threshold problem. With a long threshold, the waiting time before removing the entries will be too long. Because the GFS algorithm forwards the packets to nodes close to the destination, consequently, the selected

neighbour is close to the border of the node's transmission range and thus has a higher probability to have left the transmission range. Additionally, for the long threshold, considering the case of high mobility environment, a node may keep a neighbour's entry while waiting for the next BP regarding the long threshold, when in fact, the neighbour has already left the node's transmission range, due to its fast mobility.

Concerning both scenarios stated above, they might severely affect GFS performance, since performing forwarding decision in such circumstances may result in the routed packet to be dropped [51, 52]. This increases the delay, and consumes energy of nodes. ELT was not considered with the current greedy algorithm. Thus, GFS fails to provide accurate information in nodes' neighbours-lists. Consequently, to increase the performance of data routing in MANET, it is desirable to efficiently employ ELT. The frequency of ELT should be tuned and not be considered as a fixed pre-specified time.

2.5.3 The Effects of Position Inaccuracy of Participating Nodes

Recently, several studies as in [110, 111] show nodes' position, rarely, to be accurate in the highly dynamic environment, even when nodes use GPS. Thus, location measurement is often noisy and incurs some error. The next two sub-sections discuss the reasons behind position information inaccuracy.

2.5.3.1 Inaccuracy of Next-relay Node Information (INNI) Problem

There are two main reasons involved behind the inaccuracy of nodes' position information. The first one is due to the nature of GPS information. This is beyond the scope of this present study. The second reason, which revolves around what have been discussed so far that covers the following, the node mobility, BPIT, and the neighbours' entry expiry-time in nodes' neighbours-lists. This was discussed earlier in Subsections 2.5.2.

Because of the nodes' mobility, the topology changes frequently and rarely remains static. Thus, a need exists to update the changes that occur to node position due to the nodes' mobility with respect to all other neighbours [51, 52]. As indicated earlier, using a beacon messages mechanism solves this problem, but still there is a main

drawback in determining the optimal beacon interval time in MANET environment. Adding the validity of the neighbours' entry in a node's neighbours-list is related to the updating mechanism used and to the period of waiting time before a node removes a neighbour's entry from its neighbours-list.

Inaccuracy of the position information induces transmission failure and backward progress [110]. Also, as shown in Figures 2.10 in Sub-section 2.5.2.4, a transmitted packet is prone to be dropped, particularly if the selected next relay-node is out of the transmission range of sender node [111]. Also, As as shown in Figure 2.11 below, if the selected next-relay node, i.e., node *B*, is located behind the ultimate destination node *D*, it will lead to backward progress, which could cause looping [27]. Loops occur when the selected next relay-node, i.e., node *A*, is one of the previous senders in the route. As a consequence, the loop problem, which appears as a result of position inaccuracy, increases the packet loss probability and leads to more end-to-end delay due to using long path, i.e. non-optimal path [62].

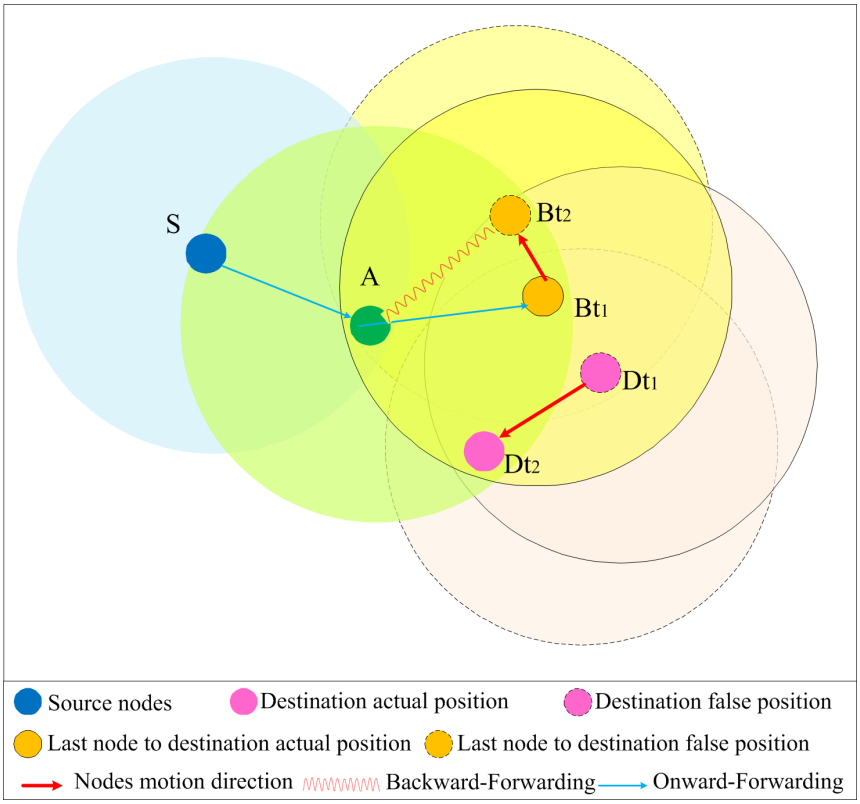


Figure 2.11. The Backward Problem due to Position Inaccuracy

Up to this point, the impact of location error on the GFS from the perspective of sender nodes has been discussed. The next section discusses the impact of location

error on GFS. This is modeling the Inaccuracy in Destination Position Information (IDPI) problem.

2.5.3.2 Inaccuracy in Destination Position Information (IDPI) Problem

As the sender node gets a destination's information (i.e., location coordinates and velocity) from the location server, it next inserts this information into the header of each forwarded packet towards that destination. As soon as the packet is sent, the sender does not care if the destination changes its position during transmission time [110]. As illustrated in Figure 2.12 below, Inaccuracy in Destination Position Information (IDPI) problem represents inaccuracy in distance between the actual and false positions of the destination node during the journey of the data packet from source to destination.

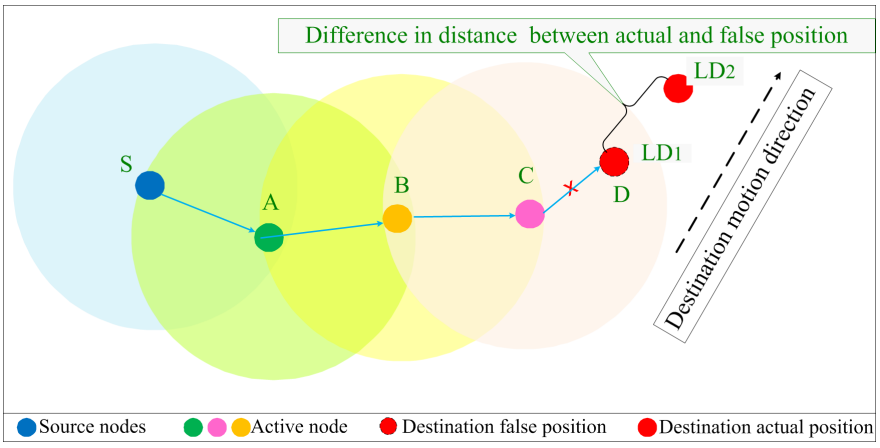


Figure 2.12. Inaccuracy in Destination Position Information (IDPI) Problem in GFS

In Figure 2.12 above, and based on node *S's* request, the location server reports that the location of intended destination is LD_1. Next, node S embeds this information in the header of data packet and transmits it to the next relay node. Every node in the path from source to ultimate destination nodes will select the next relay node and forward the packet using the same destination location information, i.e., LD_1. During transmission time to the target, the ultimate destination D changes its location from LD_1 to LD_2. Back to the last forwarding C in the path, and Based on the address of destination (i.e., node D), node C as the nearest node to node D will forward the packet to the location LD_1. In fact, this is totally wrong behaviour because D has changed its location from LD1 to LD2. This wrong choice will result in the forwarded packet being dropped. As a consequence, this wrong choice results in retransmission, more

end-to-end delay, non-optimal route, and routing loop.

2.5.4 Potential Problems Summary and Consequences

The preceding sections highlighted the problems incurred by using geometric calculation, as the only deciding factor in GFS strategy. This section introduces a quick summary of the potential problems that may occur when using GFS as the forwarding policy in position-based routing protocols. As was earlier discussed in Section 2.5 and shown in the Table 2.1 below, ignorance or lack of awareness of certain nodes' conditions, their mobility attributes, and the accuracy of their position information, while making a forwarding decision, leads to devastating consequences and degrades the performance of GFS. The shortest path as the main objective used with GFS, attracts a higher packet loss, higher end-to-end delay, and least packet delivery ratio compared to some other available routes. Table 2.1 below lists all problems and consequences.

Table 2.1. Greedy Forwarding Strategy (GFS) Problems in MANET

Issue	Problems	Consequences
The effects of participating nodes' conditions	Congestion	1, 2, 3, 4, 5
	Battery power	6,7
	Next relay-node degree	8
	Local maximum	1, 3, 5, 9
The effects of mobility attributes	Movement direction	1, 2, 3, 4, 5,9
	Movement speed	1-5and 10
	Link lifetime	1-5and 10
	Beacon packet interval time	1-5and 10
	Neighbourhood entry lifetime	1-5and 10
The effects of position inaccuracy of participating nodes	Inaccuracy of next relay-node information in sender node	1-5and 11
	Inaccuracy of destination information in sender node	1-5and 11

1. Packet loss 2. Retransmission 3. More end-to-end delay 4. More overhead 5. Increase battery power consumption 6. Minimize the network lifetime 7. Variation in the nodes degree 8. Forwarded packet may reach a dead-end 9. Incurs non-optimal rout 10. Link breakage 11. Routing loop

2.6 Current Literature Efforts to Solve GFS Failure

In a position-based routing protocol, a packet forwarded greedily may face one of these possible ends. It may greedily reach its final target hop-by-hop as was planned for, or it may be dropped due to GFS failure [30, 31]. The reason behind this failure is one of trade-off of several problems mentioned in Section 2.5. This failure appears because GFS was proposed for use in static networks not in MANET. As stated in [27, 30, 31], there are many studies that show how to handle GFS failure. This research classifies broadly the existing efforts into two main categories, each category designed with different approaches, regarding the procedures used to solve GFS failure in MANET. As shown in Figure 2.13, below, these categories are: Recovery Strategies to Handle GFS Failure (RSGF) and Supportive Enhancement for GFS (SEGF), besides using a recovery strategy [27, 30, 31].

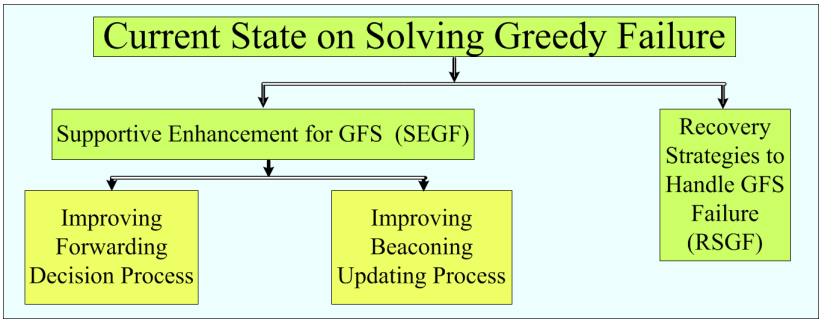


Figure 2.13. Current State-of-the-art on Solving Greedy Problems

2.6.1 Recovery Strategies to Handle GFS Failure (RSGF)

As mentioned earlier in Sub-section 2.5.1.4, as the forwarded packet reaches a void, it cannot be forwarded greedily anymore and gets stuck at the dead-end node [65, 66, 67]. An area at which communications between nodes is denied, for any reason, should be regarded as a void or hole area where a forwarded packet suffers from a dead-end problem [28]. The dead-end problem motivated the appearance of supportive recovery strategies (RSGF) (or supportive void-handling techniques) to handle and to overcome GFS weakness dead-ends [65, 66, 67]. A recovery strategy is defined as the routing mode executed by the void node to face the dead-end problem [28]. In the recovery mode, the stuck packet is forwarded from the geographic hole node towards the destination by using a non-greedy algorithm. Thus, the desired MANET performance might be attained.

Most existing position-based routing protocols that use GFS differ from each other in how they handle communication holes (dead-end problem) [65, 66, 67]. The proposed solutions can be classified as Planar graph-based, Geometric, Flooding-based, Cost-based, Heuristic and Hybrid void-handling techniques [65, 66, 67]. This section covers the most famous recovery strategies in MANET. This section presents brief information about Greedy Perimeter Stateless Routing (GPSR) [44], Position-based Opportunistic Routing (POR) [45], Node Elevation Ad-hoc Routing (NEAR) [28], Greedy Distributed Spanning Tree Routing (GDSTR) [55] and novel potential routing scheme Yet Another Greedy Routing (YAGR) protocol [56].

2.6.1.1 Greedy Perimeter Stateless Routing (GPSR) Protocol

Karp and Kung [44] proposed the Greedy Perimeter Stateless routing (GPSR) protocol. The GPSR apply perimeter routing or face routing to handle the hole problem to guarantee delivery. GPSR is a stateless scheme because it only requires knowledge of local topology information within a single hop neighbourhood. The GPSR routing protocol uses a combination of basic GFS [23] and face routing recovery mode as shown in Figure 2.14 below.

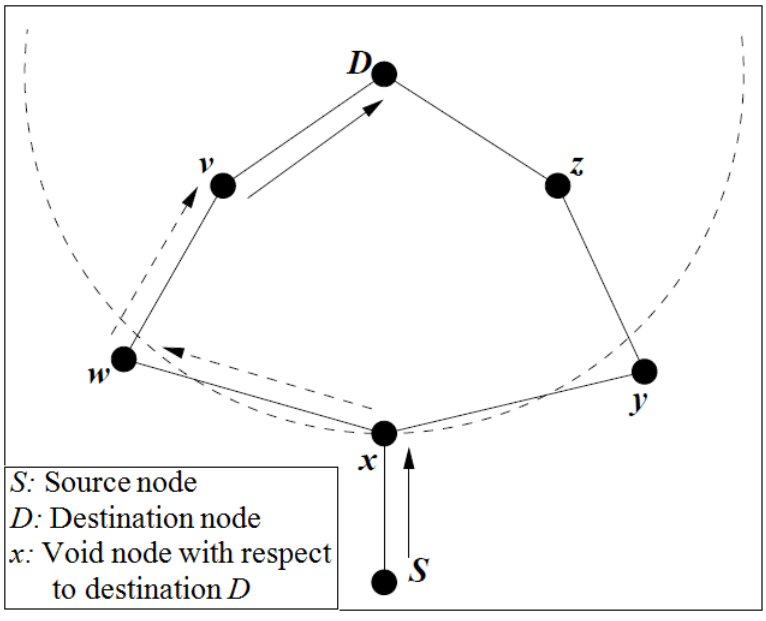


Figure 2.14. Mixed of Primary (Solid Arrows) and Recovery Mode (Dashed Arrows) (Source [44])

As illustrated in Figure 2.14 above, in the primary mode, i.e., the GFS mode, a for-

warder node chooses the next relay node by selecting the neighbour that is geographically closest to the destination. GFS fails when there is no neighbour available that is closer to the destination than the current forwarder node. To address this problem, GPSR algorithm executes its second mode, i.e., recovery mode.

The main techniques in recovery mode are planar graph traversal algorithm and distributed planarization algorithm. The former is used to form a planar sub-graph from the original plane graph of the network. The latter is used to discover a path between communication hole and ultimate destination. Thus, the performance of GPSR scheme totally depends on two related concerns. The first one is concerned about the ability of the planar graph algorithm to construct a connected sub-graph from the original graph of the network in dense as well sparse scenarios. The second one is concerned about the ability of the planarization algorithm to find a reliable rout between communication hole and destination nodes in dense as well sparse network scenarios.

In general, the graph formed by MANET network is not planar. Thus, to form a planar sub-graph, the GPSR scheme uses planarization algorithm such as Relative Neighbourhood Graph (RNG) algorithm [112] or Gabriel Graph (GG) algorithm [113]. Once the planar sub-graph is constructed, then the face routing can be performed. The main idea in a face algorithm is to forward a packet along the interiors of a sequence of adjacent faces that are intersected by the straight line connecting the source and the destination nodes. These adjacent faces provide progress towards the ultimate destination. Face traversal is done in a localized way by applying the well-known right hand rule (or left hand rule): a packet is forwarded along the next edge clockwise counter clockwise from the edge where it arrived.

Figure 2.15 shows the procedures the GPSR algorithm executes in the recovery mode. In Figure 2.15, it is supposed that the mode converts to perimeter at node w for a packet destined to node D. The packet will loop around an interior or an exterior face of the planar graph. The information about the first edge traversed can be used to prevent looping, this information then is used to determine if a packet traverses the first edge on the current face for the second time.

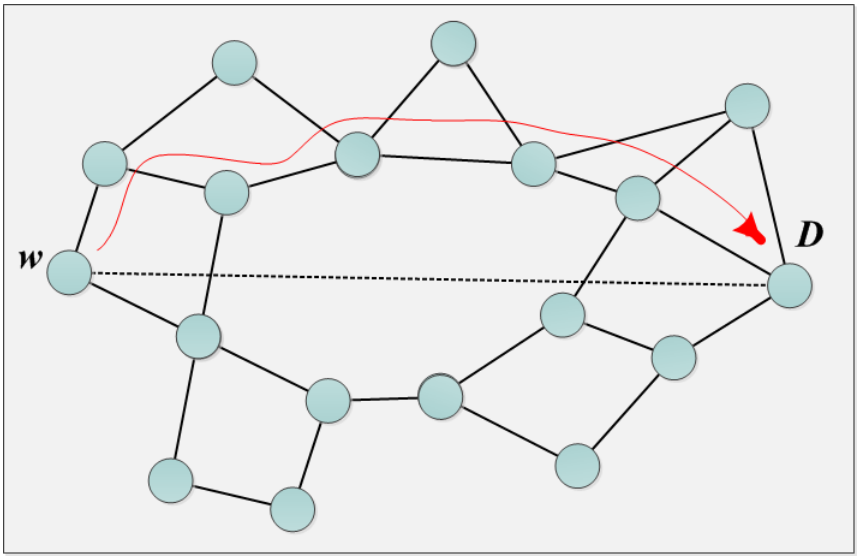


Figure 2.15. A Constructed Route by using planarization and traversal algorithms in GPSR (Source [65])

Karp and Kung in [44] evaluated the performance of the GPSR using network simulator Ns2. The researchers used three different simulation areas which are 1500×300 *m*, 2250×450 *m* and 3000×600 *m* respectively. GPSR is evaluated in two simulation scenarios. The first examines the effectiveness of the GPSR algorithm under various pause times (0, 30, 60, and 120 *s*). The second scenario examines the effectiveness of the GPSR algorithm under various numbers of nodes (50, 112, and 200 *nodes*). To generate traffic, researchers used 30 different CBR flows with packets size of 64 bytes. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The radio range of each node was set at 250 meters. The random waypoint mobility model was utilized in the simulation. The metrics that were used to evaluate the performance of GPSR were packet delivery ratio, average control overhead and routing path length. Based on the simulation results, researchers stated that GPSR outperformed other ad-hoc routing protocols and shortest path scheme in the two proposed scenarios. Also, under mobility's frequent topology changes, the GPSR algorithm can quickly find optimal new routes solely relying on local topology information.

Although planar graph-based routing guarantees packet delivery, the chosen route may not be optimal. Thus, it involves more nodes than GFS, consumes more energy and induces more end-to-end delay [67]. Additionally, in some cases, the GPSR scheme fails if the packet's Time To Live (*TTL*) expires before the packet reaches a correct

neighbour. In the common sparse MANET, the planarization process comes at the cost of three main drawbacks. One, it increases the cost in terms of overheads to maintain the planar sub-graph locally in each participant node. Two, it produces inappropriate routing paths along the boundaries of holes. Three, overloading of nodes on the boundaries of holes exhausts the batteries of those nodes quickly, which further enlarges the holes and eventually connects small holes to big holes or even partitions the network into disconnected pieces [18].

2.6.1.2 Position-based Opportunistic Routing (POR) Protocol

Yang et al. in [45] proposed a Position-based Opportunistic Routing (POR) protocol. POR is a stateless routing protocol that aims to guarantee delivery in the presence of hole problem. The functionality of POR requires knowledge of topology information locally within a single hop neighbourhood. The POR routing protocol uses a combination of basic GFS [23] and Virtual Destination-based Void Handling (VDVH) as a recovery mode. In the primary mode, i.e. GFS mode, a predefined number of candidate nodes cache the packet. If the optimal candidate does not forward the packet (for any reason) in certain time slots, one of the suboptimal candidate nodes will proceed to forward the data packet. In this respect, as at least one candidate node succeeds in forwarding the packet, there will be no chance that the packet will be lost. The primary mode fails if none of the candidate nodes succeeds in forwarding the packet. POR will continue forwarding the stuck packet by activating the second algorithm, i.e., VDVH mode. The VDVH algorithm is executed by the hole node (where the packet is stuck) by sending a message called a "void warning". The "void warning" packet is the same stuck data packet returned from the hole node to the previous forwarder node (trigger node). As soon as the trigger node receives the void warning, it will switch from GFS mode to VDVH mode and reselect a new next relay node. The new selection is based on repositioning of the destination node by giving it a new virtual location. In case the source node itself is a hole node, the packet-forwarding mode will be set as VDVH at the source node itself.

The core idea from VDVH is to select two virtual positions for the original destination. The trigger node has to select to forward either to the left virtual destination position or

to the right. As shown in Figure 2.16 (a) below, virtual destinations are repositioned at the circumference with the trigger node as centre. They are used to guide the direction of packet delivery during VDVH mode. The VDVH has the potential to deal with all kinds of communication voids. Figure 2.16 (b) below shows an example while executing conventional GPSR and GPSR-VDVH. The figure shows that GPSR-VDVH forward the packet by using an optimal path of seven hops while conventional GPSR experiences a much longer route of 15 hops [45].

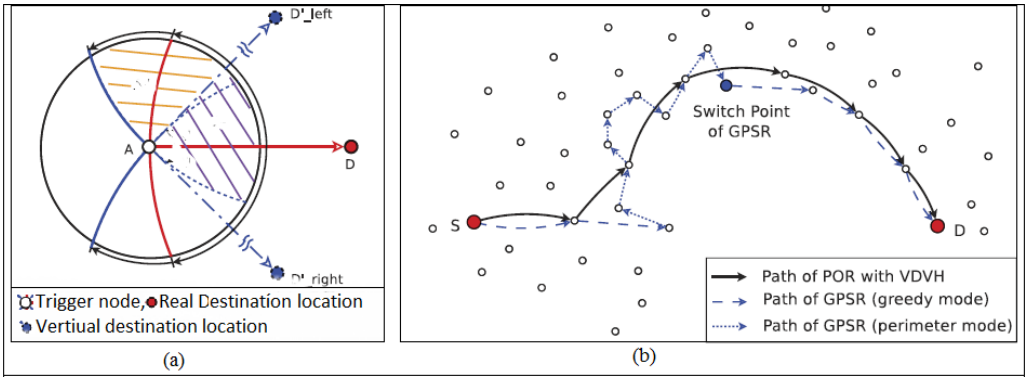


Figure 2.16. (a) Potential Forwarding Area Using Virtual Destination, (b) A Path Exploited by GPSR and GPSR-VDVH Protocols (Source [45])

Yang et al. in [45] evaluated the performance of POR using network simulator Ns2. In the simulation area of 1200 m × 800 m, 80 nodes have been randomly distributed. The simulation time is set to 900 s. POR was evaluated in one simulation scenario. This scenario examines the effectiveness of the POR algorithm under various nodes' speed (5-50 $m\backslash s$). To generate traffic, a CBR flow was used at a rate of 10 packets per second with packet size of 256 bytes. Each flow was started at 170 s and ended at 870 s. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The radio range of each node was set at 250 meters. The random waypoint mobility model was utilized in the simulation. The metrics used to evaluate the performance of POR were packet delivery ratio, average control overhead, routing path length, packet forwarding times per hop and Packet forwarding times per packet.

Yang et al. in [45] argued that simulation results show that POR, under high node mobility, outperforms current position-based routing protocol in terms of end-to-end delay and packet delivery ratio, with acceptable extra overhead. However, in proposed

POR, VDVH results in a longer path that needs more computational procedures to escape from the void. Increased computational procedures lead to more delay and draining of more power. In addition, the selection of the next relay node was just based on distance as the main criterion and did not account for any other important criteria to solve traditional GFS failure. The POR scheme assumes accurate position information to build its virtual path to the destination. Assuming accurate position information under high node mobility is an unrealistic assumption. Additionally, in some cases, the POR scheme will fail during the building of the virtual path if the packet's TTL expires before the packet reaches a correct neighbour.

2.6.1.3 Node Elevation Ad-hoc Routing (NEAR) Protocol

Arad and Shavitt in [28] proposed Node Elevation Ad-hoc Routing (NEAR) protocol. It was a new idea using a virtual repositioning of participating nodes. The purpose of NEAR is to effectively increase GFS efficiency and to minimize the need for using a recovery strategy. To achieve this goal, NEAR algorithm predicts hole nodes in advance and adjust nodes' coordinates to improve routing performance, The NEAR algorithm is comprised of three main algorithms: the repositioning algorithm, the void identification algorithm and the greedy routing algorithm. The brilliant idea in NEAR is the repositioning algorithm which is used to detect so-called 'hole" or "concave" nodes.

The repositioning algorithm calculation is performed locally, based on current node's neighbour positions. In the NEAR algorithm, repositioning process attains three objectives. The first is to mark hole nodes. The second is to increase GFS success rate. The third is to improve the recovery process. Nodes using a NEAR algorithm try to predict dead ends and incrementally adjust routing coordinates based on angle that each node makes with its neighbours. With NEAR, GFS uses virtual coordinates to surround the hols.

Figure 2.17 below explains a simple repositioning process for a node. The figure shows five hole nodes. After applying a NEAR algorithm, the virtual position ($\tilde{O}$) of node (O) is shown above the plane. The projection of node's O virtual coordinates lies between its neighbours E and C. Node's $\tilde{O}$ height is intended to help the NEAR

algorithm direct data packet away from an expected hole node.

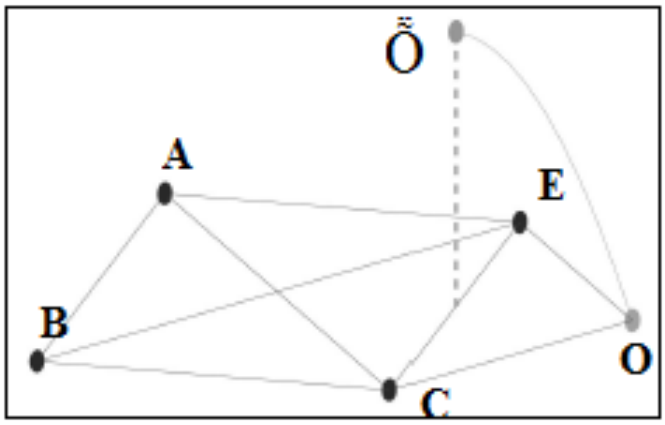


Figure 2.17. A repositioning example (source [28])

Arad and Shavitt in [28] evaluated the performance of NEAR using network simulator Ns2. The simulation area was set to 2000 $m \times 2000\ m$. NEAR was evaluated in two simulation scenarios. The first examined the effectiveness of the NEAR algorithm under a randomly placed hole, whose size was 10% of the network size. The second scenario examined the effectiveness of the NEAR algorithm in which the hole, whose size was 25% of the network size, was placed at the centre of the simulation area. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The radio range of each node was set at 250 *meters*. The random waypoint mobility model was utilized in the simulation. The metrics that were used to evaluate the performance of NEAR were average disconnected nodes, average number of iteration, average hops ratio and average routing success rate. Arad and Shavitt in [28] argued that the simulations results showed the superiority of NEAR and a significant decrease in the number of hole nodes compared to current routing protocols due to the improvement done to GFS.

However, in NEAR, to maintain a height data structure, a NEAR scheme needs various auxiliary algorithms. Executing these algorithms imposes more efforts from the participating nodes that may drain MANET resources. While the NEAR scheme's success rate is high, NEAR may fail to find a route to the destination even when there is a route to the intended destination. Finally, the NEAR scheme is a local algorithm and thereby only can detect local holes and is ineffective in sparse scenarios.

2.6.1.4 Greedy Distributed Spanning Tree Routing (GDSTR) Protocol

Leong et al. [55] proposed a new geographic routing algorithm, Greedy Distributed Spanning Tree Routing (GDSTR). The purpose of the GDSTR algorithm is to the find shortest routes with less maintenance traffic than existing algorithms. In the GDSTR scheme, spanning trees are proposed for use in place of planarization algorithms. The GDSTR algorithm uses two hull trees instead of planarization as a recovery mode. Nodes exchange messages to compute and store a distributed spanning tree. Each node also computes and stores a convex hull of the locations of all of its descendants in the subtree rooted at the node; the resulting tree is called a hull tree. Figure 2.18 below shows the procedure to construct the hull trees by the participating nodes.

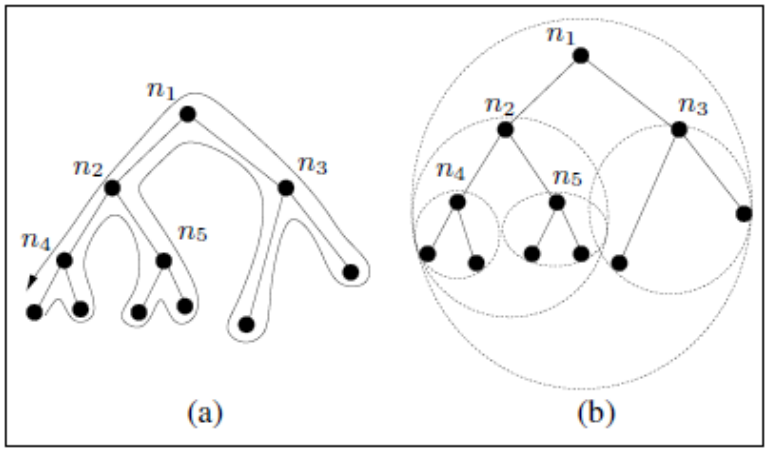


Figure 2.18. (a) A spanning tree and (b) a hull tree construction in GDSTR (source [55])

In the GDSTR algorithm, a data packet is forwarded greedily until it is stuck at a hole. At this point, GDSTR switches from GFS to hull tree-mode then the stuck packet is either sent to the parent node or to child node. In the former method, a packet proceed by parents one by another. By using this method, the forwarded packet either it finally reaches a parent node that has a child that leads to the destination or forwarded packet will be dropped as the destination is unreachable. In the latter method, the forwarded packet sends to a suitable child node directly. This child should has convex hull contains the destination node.

Leong et al. [55] evaluated the performance of GDSTR using network simulator Ns2. GDSTR was evaluated in two simulation scenarios. The first examined the effectiveness of GDSTR algorithm under various numbers of nodes (25-500 nodes). The

second scenario examined the effectiveness of GDSTR algorithm under various nodes' degree (0-16). The other parameters were set as in [44]. The metrics used to evaluate the performance of GDSTR were path length and hop counts. The researchers argued that their simulations results showed that GDSTR uses less bandwidth and achieve lower route and hop stretch than other existing routing protocol that use planarization method [55].

Although the packet delivery is guaranteed by using tree traversal, GDSTR needs to gather the necessary information to build the two hull trees. To execute these processes a large storage capacity of the nodes is needed. This leads to an increase in the routing costs and the end-to-end delays and induce a higher overhead. Another drawback of GDSTR is that the detected routing path is longer than the shortest route, which drains MANET resources.

2.6.1.5 Yet Another Greedy Routing (YAGR) Protocol

Na et al. in [56] proposed Yet Another Greedy Routing (YAGR) protocol. The purpose of YAGR algorithm is to eliminate the local maximum condition and to improve GFS [23] by utilizing the concept of the potential field. Potential field technique is used in robot navigation. This technique is used to control robot movement from one position to another. To create a dynamic potential, the control process is achieved by using the virtual repulsive force in combination with the virtual attractive force of the target field. The same technique is used in YAGR. The YAGR algorithm contrasts the roles of the packet with the robot and routing holes with obstacles and determines the potential field of neighbourhood to choose the next relay-node with the highest potential field [56]. The information of the observed hole node is emitted and maintained to minimize the probability of a packet to fall in a hole node. To achieve this, each forwarded packet's header contains the Dead End Information (DEI) combined with the destination location information.

Na et al. in [56] evaluated the performance of YAGR using network simulator Ns2. The simulation area set to 2000 $m \times 2000\ m$. To show the effectiveness of YAGR, it was examined under various nodes' degree (4-12). To generate traffic, a CBR flow was used. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF)

was used for all the nodes in the simulation. The radio range of each node was set at 250 *meters*. The random waypoint mobility model was utilized in the simulation. The metric used to evaluate the performance of YAGR was the path length. Results were calculated with a confidence interval of 95%.

Na et al. in [56] argued that YAGR preserves the simplicity of traditional greedy routing, and outperforms current planarization routing protocols. The simulation results showed that YAGR protocol outperforms GPSR in terms of path stretch and achieved a similar performance to GPSR in terms of path failure rate. However, by exploiting YAGR algorithm, the hole problem can be partially relaxed or totally resolved only by means of significant routing overheads and considerable computational processes.

2.6.1.6 Summary of Existing Recovery Strategies in Position-based Routing Protocols

The above discussion in Sub-section 2.6.1 presented a selected group of routing protocols belongs to Recovery Strategies to Handle GFS Failure (RSGF) category. These protocols have been developed to improve the GFS success rate and to guarantee packet delivery. As discussed above, These protocols come at the cost of several drawbacks. Table 2.2 below shows the pros and cons of each discussed scheme.

Table 2.2. Prior Works Regarding Recovery Strategies in Position-based Routing Protocols

Author	Protocol	Advantages	Disadvantages
Arad and Shavitt [28]	NEAR	Guarantee packet delivery, improves GFS success rate and decrease the number of hole nodes	Impose more efforts from the participating nodes that drain MANET resources and its algorithm fails to find a route in sparse scenarios
Karp and Kung [44]	GPSR	Guarantee packet delivery	Causes long routing path, drain network resources, and planarization process may disconnect a connected network in sparse scenarios
Yang et al. [45]	POR	Guarantee packet delivery	Long routing path, drain network resources and result in more delay
Leong et al. [55]	GDSTR	Guarantee packet delivery, less bandwidth and lower route and hop stretch	Induce more end-to-end dely, exhaust nodes storage, and may fails in high mobility scenarios
Na et al. [56]	YAGR	Guarantee packet delivery, and decrease path stretch	Causes significant routing overheads

2.6.2 Supportive Enhancement for GFS (SEGF)

The preceding section showed that RSGF category approaches still have several disadvantages to be solved. In this regard, recent years have seen a massive amount of research targeting enhancing GFS itself besides using a recovery approach [27, 30, 31]. In this research, the proposed approaches to overcome the shortcoming with GFS and RSGF approaches are dubbed Supportive Enhancement for GFS (SEGF). The main purposes of the SEGF algorithms category are to minimize the use of the recovery mode and to increase GFS success rate, so that the desired MANET's performance might be attained [27, 30, 31].

In the SEGF category, researchers' works targeting GFS enhancement can be divided

into two classes. The first class concerns improving the forwarding decision process, by adopting another metric(s) besides shortest path to make the forwarding decision. The second class concerns improving the beaconing updating process. In the following sub-sections, the two classes are discussed directly.

2.6.2.1 Extensions on GFS based on Prioritization and Selection Process

To achieve the shortest path objective, GFS selects the next relay-node that is closest to destination as an adopted criterion [23]. As it was extensively discussed in Section 2.5, in GFS, the lowest hops count routing metric is not indicative of the quality of the path. Considering the shortest path as the only routing objective in MANET is insufficient for computing a reliable route. Because adopting the shortest path as the only routing objective can severely compromise MANET's performance on the outstanding ignored objectives. Additionally, it results in a irregular traffic load distribution that leads to a hot-spot problem, which also degrades the whole system's performance [38].

To conclude, simple MANET routing protocols that achieve one objective based on a single metric such as a minimum number of hops or residual battery power, are no longer sufficient [114]. Due to the complex interactions among the multi-objectives, and uncertainty coupled with MANET, achieving multiple objectives is a critical goal to be accomplished. Thus, one crucial issue in MANET is how to find optimal paths that fulfil multi-objectives [114]. Some literature has looked at possible solutions. The following sub-sections provide a brief examination of some prominent work in interest areas such as Multi-criteria Receiver Self-Election (MRSE) scheme [57], Energy-Aware Geographic Routing (EGR) [41], Delay and Reliability-Aware Routing (DR2) [58], Stability and Reliability Aware Routing (SRR) [60] and Predictive Directional Greedy Routing (PDGR) protocol [59].

Multi-criteria Receiver Self-Election (MRSE) Protocol

Khokhar et al. in [57] proposed a Multi-criteria Receiver Self-Election (MRSE) protocol. The purpose behind MRSE is to tackle traffic congestion and high mobility issues and to enable the candidate receivers to make the forwarding decision. MRSE is a distributed routing scheme. To select the optimal next relay-node, MRSE makes the use

of four routing parameters, which are: optimal distance, link lifetime, received power and optimal transmission range. The MRSE scheme suggests suppressing the HELLO scheme and using the four handshaking messages (RTS\CTS\DATA and ACK). In MRSE structure design and based on the four routing parameters, the optimal next relay-node should reply back first with CTS at a short time. To avoid time collisions between nodes' CTS messages, a reasonable waiting time difference has been set with the consideration that this period of time should be too small to prevent extra delays.

Khokhar et al. in [57] evaluated the performance of MRSE using Traffic-Aware Routing Strategy (ReTARS). The simulation area was set to 3968 $m \times 1251$ m. The simulation time was set to 400 s. MRSE was evaluated in two simulation scenarios. The first examined the effectiveness of the MRSE algorithm under various numbers of nodes (150, 250, and 350 *nodes*). The second scenario examined the effectiveness of the MRSE algorithm under various nodes' speed (20-80 *m/h*). To generate traffic, a CBR flow was used with packets size of 1024 *bytes*. The IEEE 802.11b MAC protocol with distributed coordinating function (DCF) is used for all the nodes in the simulation. The STreet RAndom Waypoint (STRAW) mobility model was utilized in the simulation. The beaconing frequency was set to 1s. The metrics that were used to evaluate the performance of MRSE were packet delivery ratio and network lifetime.

The researchers in [57] said that the simulation results showed that the MRSE scheme offered better performance compared to two other routing protocols that used the four handshaking schemes and HELLO scheme. However, MRSE scheme can be only efficient when data is on going, otherwise the position information accuracy of the participating nodes will go down. This results in degrading the performance of the routing protocol and the whole system.

Energy-Aware Geographic Routing (EGR) Protocol

Gang and Guodong in [41] proposed Energy-Aware Geographic Routing (EGR) protocol. EGR is a novel position-based routing protocol that aims to prolong the network lifetime and to ensure high packet delivery ratio. EGR algorithm combines distance to destination and residual energy to make forwarding decisions. In addition, to improve the delivery ratio EGR algorithm uses a prediction scheme to predict a destination's

movement. In EGR, each node should be aware of its own location and residual energy. Also, a node that has data to be sent to a specific target should be aware of its neighbours' location and residual energy information to make the forwarding decision. Thus, every node emits a HELLO packet (beacon). This beacon contains a node ID, location and residual energy value. A sender node gets destination's information by the means of using location server. EGR adopted the beaconing scheme used in DREAM [88], i.e., restricted flooding approach. EGR uses the same recovery approach as in GPSR [44]. In EGR, the lifetime of the network is defined as the moment when the first node reaches its fifth of its initial energy.

Gang and Guodong in [41] evaluated the performance of EGR using network simulator Ns2. The simulation time was set to 300 *sec*. EGR was evaluated in two simulation scenarios. The first examined the effectiveness of the EGR algorithm under various numbers of nodes (100-300 *nodes*). The second scenario examined the effectiveness of the EGR algorithm under various nodes' speed (5-25 *m\s*). To generate traffic, a CBR flow was used with packets size of 64 *bytes*. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The channel's bandwidth was set to 2Mbps. The radio range of each node was set at 250 *meters*. The initial nodes' energy was set to 1000*J*. The metrics that were used to evaluate the performance of the EGR were packet delivery ratio and network lifetime. As a new routing protocol, EGR protocol combines greedy forwarding, energy awareness approach and constrained flooding. Thus, EGR protocol effectively prolongs the MANET lifetime as well as increases packet delivery ratio and decreases end-to-end delay [41]. However, EGR protocol uses flooding approach, which drains the network resources. Also, due to using a recovery strategy EGR may fail in spars scenarios. And finally, the EGR protocol uses only residual energy besides distance routing metric. Thus, it may prolong the network's lifetime at the cost of the other overlooked routing objectives.

Delay and Reliability aware Routing (DR2) Protocol

Kayhan et al. [58] introduced a novel Delay and Reliability Aware Routing (DR2) protocol. The purpose of DR2 is to forward data packets between two communicating

nodes very quickly with high reliability and low latency. As shown in Figure 2.19 below, DR2 protocol is considered to be a cross layer scheme between the MAC and the network layers. Cross layer cooperation facilitates the communication process that leads to less overhead due to the conversion of a little information between the two layers.

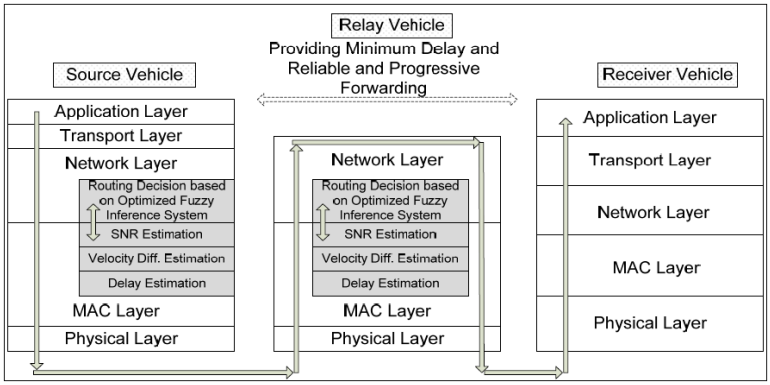


Figure 2.19. The architecture of the proposed DR2 (source [58])

The MAC layer is responsible for gathering information about specific metrics concerning relative velocity and channel condition. These metrics are the Signal to Noise Ratio (SNR), the path delay and the relative velocity. Collected information is then forwarded to the network layer to select the best next-relay node. DR2 algorithm incorporates the concept of Fuzzy Logic Controller (FLC) to calculate the cost of each node's neighbour in terms of the three metrics. The three routing metrics are considered as input to the Fuzzy logic system, and the output is the fuzzy cost function. In DR2 building structure some assumptions have been made to track the routing metrics by MAC layer. It is assumed that nodes are embedded with GPS receiver. GPS is used to provide nodes with current position information [84]. To facilitate communication between nodes, it is also assumed that each node is equipped with wireless radio communication devices.

Kayhan et al. [58] evaluated the performance of DR2 using network simulator Ns2. The simulation area was set to 2400 $m \times 2400$ m. Simulation of Urban Mobility (SUMO) was utilized in the simulation. The simulation time was set to 100 s. To generate traffic, a CBR flow was used with packets size of 1000 *bytes*. DR2 was evaluated in two simulation scenarios. The first examined the effectiveness of DR2 algorithm under various standard deviation (1 to 10). The second scenario examined

the effectiveness of DR2 algorithm under various numbers of nodes (50,100,150 and 200 *nodes*). The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The channel's bandwidth was set to 2Mbps. The radio range of each node was set at 250 *meters*. The metrics that were used to evaluate the performance of DR2 were packet delivery ratio, end-to-end delay and reachability. The researchers argued that (based on the simulation results) DR2 could be used in real time systems in dense and sparse networks environment [58]. Moreover, they added that the DR2 was able to improve end-to-end delay and packet delivery ratio. However, the DR2 algorithm has other important routing metrics that are not considered in the selection process such as buffer occupancy that have bad effects on routing performance in MANET.

Stability and Reliability Aware Routing (SRR) Protocol

Kayhan et al. [60] proposed a novel Stability and Reliability Aware Routing (SRR) protocol. The purpose of SRR is to deliver data packets with a high degree of stability and reliability. The SRR algorithm incorporates the concept of Fuzzy Logic Controller (FLC) in position-based routing protocols to make the forwarding decision. The authors argued that distance between sender and relay nodes, as well nodes mobility attributes, are two critical routing metrics that should be considered to make a forwarding decision. To design their scheme, the authors adopted motion direction and distance as inputs of the fuzzy decision making system. Figure 2.20 below shows how a source node calculates both parameters for its neighbours.

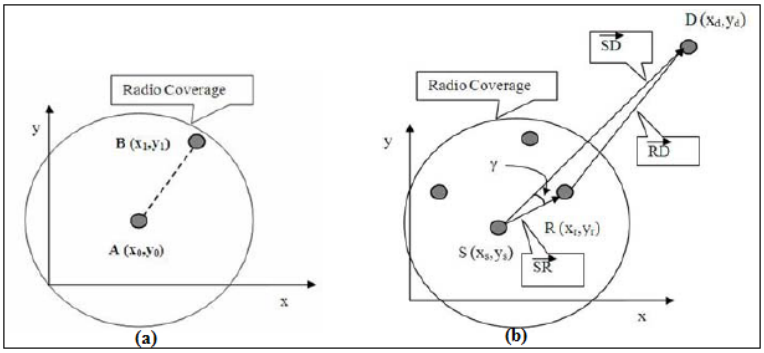


Figure 2.20. (a) Distance calculation between sender node and its neighbours (b) Relative direction calculation between sender node and its neighbours (source [60])

Based on these inputs, the next relay node is selected as it has the highest fuzzy cost.

The highest cost is given to the neighbour node, which is directed more towards the final destination. In the SRR building structure some assumptions have been made to track the routing metrics by the MAC layer. It is assumed that nodes are embedded with a GPS receiver. GPS is used to provide nodes with current position information [84]. To facilitate communication between nodes, it is also assumed that each node is equipped with wireless radio communication devices. In SRR a node can track its neighbours by the means of using periodic beaconing scheme. In case of disconnected network, the sender node will buffer the data packet and move until it finds a neighbour that is closer to the final destination other than itself.

Kayhan et al. [60] implemented SRR using network simulator JiST\SWANS. The simulation area was set to 4000 $m \times 500$ m. The simulation time was set to 160 s. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The channel's bandwidth was set to 2Mbps. The radio range of each node was set at 200 *meters*. The Intelligent Driver Model (IDM) mobility model was utilized in the simulation. SRR was evaluated in two simulation scenarios. The first examined the effectiveness of the SRR algorithm under various traffic flow (5-15 *flows*). The second scenario examined the effectiveness of the SRR algorithm under various numbers of nodes (30 to 250 *nodes*). The metrics that were used to evaluate the performance of SRR were packet delivery ratio, average control overhead, and end-to-end delay. The researchers argued that their proposed scheme outperformed traditional position based routing protocols in terms of packet delivery ratio, and control overhead [60]. However, with the proposed protocol, the authors did not account for the changing in the speed values and just take the mobility direction where both parameters have the same importance in calculating the suitability of the next relay node.

Predictive Directional Greedy Routing (PDGR) Protocol

Gong et al. in [59] proposed Predictive Directional Greedy Routing (PDGR) protocol. The purpose of PDGR is to improve selection process in position-based routing protocols. In the PDGR algorithm, a forwarding decision is made based on the position and mobility direction of nodes. Thus, PDGR combines Position First Forwarding

and Direction First Forwarding. The importance of these two factors is controlled by a weighted function. A weighted score is estimated based on the two routing metrics. The weighted score is calculated not only for a sender but also for its possible future neighbours in very near future. Thus, the PDGR algorithm constructed of two parts. The first algorithm is used to estimate a weighted score for a node's current neighbours. The second algorithm is used to estimate a weighted score for a node's future neighbours in a short interval. A neighbour node with the highest score is selected as the optimal next relay-node. If no next relay-node is available, the PDGR algorithm uses the buffer and forward method. This method saves the forwarded packet from being dropped. The information about routing metrics can be collected using beacon messages. Also, nodes can get their position information by the means of using GPS [84]. A node that has data packets to be forwarded to a destination can get the destination's position information by the means of using a location server that is known in advance.

Gong et al. in [59] evaluated the performance of PDGR using network simulator Ns2. The simulation area was set to 2400 $m \times 2400$ m. To generate traffic, a CBR flow was used with packets size of 512 *bytes*. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The radio range of each node was set at 250 *meters*. The Realistic Mobility Model was utilized in the simulation. The beaconing frequency was set to 0.5 *s*. PDGR is evaluated in two simulation scenarios. The first examined the effectiveness of the PDGR algorithm under various traffic flows (0.1, 0.2, 0.5, 0.8, 1, 2, 3 and 4 *flows*). The second scenario examined the effectiveness of the PDGR algorithm under various nodes' speeds (15-60 *mp\h*). The metrics that were used to evaluate the performance of the PDGR were packet delivery ratio, average control overhead, and end-to-end delay. The PDGR was shown in simulations to outperform GPSR [44] in terms of packet delivery ratio and latency of packet delivery. However, in PDGR the delivery of data packet is not guaranteed to the node at the far edge of forwarding transmission range. This will lead to a low packet delivery ratio and high end-to-end delay.

2.6.2.2 Summary of Prominent Extensions Works for GFS based on Prioritization and Selection Process

The above-discussed routing protocols have been developed to improve the GFS success rate. The proposed schemes come at the cost of several drawbacks. Table 2.3 below shows the pros and cons of each scheme discussed.

Table 2.3. Prior Works Regarding Prioritization and Selection Process improvement for GFS

Author	**Protocol**	**Advantages**	**Disadvantages**
Khokhar et al. [57]	MRSE	Increases packet delivery ratio and decreases the end-to-end delay	Inefficient when there is no data ongoing that decreases nodes' position information accuracy
Gang and Guodong [41]	EGR	Prolongs the MANET lifetime, increases packet delivery ratio and decreases the end-to-end delay	Flooding drains the network resources, due to using a recovery strategy EGR may fails in spars scenarios
Kayhan et al. [58]	DR2	Can be used in real time systems, improves the end-to-end delay and packet delivery ratio	Overlooked some important routing metrics that have very bad effect on the performance of routing in MANET
Kayhan et al. [60]	SRR	Outperforms traditional position based routing protocols in terms of packet delivery ratio and control overhead	Overlooked some important routing metrics that have very bad effect on the performance of routing in MANET such as relative speed
Gong et al. [59]	PDGR	Outperforms traditional position based routing protocols in terms of packet delivery ratio and latency	Packet delivery is not guaranteed to the node at the far edge of forwarder transmission range

2.6.2.3 Extensions on GFS based on Beaconing-Update Scheme

In MANET, arbitrary motions of mobile nodes introduce a frequent and an unpredictable change in network topology [47, 48, 54, 109]. In position-based routing protocols, a mobile node needs to preserve fresh information of its neighbourhood in its neighbours-list to perform efficient forwarding decisions [49, 50]. As a feasible solution to the domain problem in conventional a GFS algorithm in MANET, mobile nodes proactively and periodically exchange HELLO packets (beacons) with their neighbouring nodes. The frequency at which these nodes emit their beacons is typically fixed. In this thesis, this is called the fixed beacon packet interval time (FBPIT). The FBPIT specifies the maximum time interval between the transmissions of beacons among the nodes [51].

Determining a suitable FBPIT time is challenging. A short FBPIT leads to obsolete location information, while a long FBPIT may afford more accurate information, but incurs heavier overhead. The literature suggested that none of the suggested beacon interval time schemes performs as well as they are supposed to be used optimally with MANET [52, 53].

Additionally, with the GFS algorithm, the refreshment frequency of routing entries in neighbourhood list also has a fixed value. Usually, it is set as a multiple beacon frequency. However, this is not enough to adapt well to different requirements in mobility environments [52]. Nevertheless, whatever the way used to get accurate location information, the advantages of using accurate location information outweigh the additional costs [53].

To sum up, in position-based routing protocols, the rule of beaconing is considered as one of the most important processes controlling the determination of node connectivity. However, the specified FBPIT is no less important than the used beaconing mechanism itself [51]. It is critical to decide upon the best FBPIT value to be used by the beaconing method to avoid unnecessary control overhead [53].

The most commonly used beaconing method in MANET is Simple Flooding due to its simplicity. MANET experimental results revealed that Simple Flooding was unreliable and consumes much network resources [51]. In this regard, to solve the Flooding

problem, various adaptive beaconing methods have been proposed. Generally, the new proposed beaconing update can be classified as: time-based, distance-based, mobility-aware-based, and dead reckoning update [115]. The following sub-sections provide a brief look at some prominent work in the interest area such as those presented in [61, 63] Fuzzy Logic Probabilistic (FLoP) Scheme and Fuzzy Logic Probabilistic with Safety (FLoPS), Fuzzy logic-based Dynamic Beaconing (FLDB) [62], several HELLO approaches presented in [49], a Turnover Based Adaptive HELLO Protocol (TAP) [50] and Neighbourhood Lifetime Algorithm (NLA) [64].

FLoP and FLoPS HELLO Schemes

To avoid unnecessary control overhead incurred by HELLO protocols, Liarokapis and Shahrabi [61, 63] proposed two novel broadcast algorithms that incorporate the concept of Fuzzy Logic. The first one is the Fuzzy Logic Probabilistic (FLoP) Scheme, and the second is the Fuzzy Logic Probabilistic with Safety (FLoPS) Scheme. Both algorithms were proposed to alter proportionally and appropriately the HELLO sending frequency based on the Fuzzy Logic concept. In FLoP, two sequential hello intervals are based on how dynamic or static a MANET is. The main idea here is that a node's degree, i.e., its neighbours-number, changes based on the rate of the topology changes. The situation of MANET at a given time is relying on the differentiation in the number of neighbouring nodes (ΔN) in a node's neighbour-list. In case the value of ΔN is very low or 0, then FLoP algorithm increases the hello packet-sending rate to avoid unnecessary overhead and vice versa. In the FLoP algorithm, Fuzzy Logic is utilized to alter adaptively the hello interval time. FLoPS, the second proposed approach, is an enhanced version of FLoP. FLoP achieved low performance in dense scenarios that caused unnecessary overhead. The FLoPS algorithm is proposed to handle this drawback. The FLoPS algorithm incorporates the advantages of Fuzzy Logic as it is used in FLoP, but Fuzzy Logic is utilized to adaptively switch to safety mode when necessary. In FLoPS, after the hello interval reached, a safety check takes place by applying the fuzzy controller. In case a node's neighbourhood is greater than the threshold value (set to 4 nodes), another check process is performed. If the calculated hello interval is lower than its default value, FLoPS triggers the safety algorithm to set the interval to

the default value (set to 3 seconds).

Liarokapis and Shahrabi [61, 63] implemented FLoP and FLoPS using network simulator Ns2.34. The simulation was set to 1000 $m \times 1000$ m. The simulation time was set to 1000 s. To generate traffic, a CBR flow was used with packets size of 512 *bytes*. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The channel's bandwidth was set to 2Mbps. The radio range of each node was set at 250 *meters*. The Random WayPoint mobility model was utilized in the simulation. Both protocols were evaluated in three simulation scenarios. The first examined the effectiveness of both protocols algorithms under various numbers of nodes (30-120 *nodes*). The second scenario examined the effectiveness of both protocols algorithms under various nodes' speeds (1-20 $m\backslash s$). The third scenario examined the effectiveness of both protocols algorithms under various traffic loads (1-20 *flows*). The metrics that were used to evaluate the performance of both protocols were packet delivery ratio, average control overhead and reachability.

As stated by the researchers in [61, 63] simulation results demonstrated the superiority of both FLoP and FLoPS compared to a fixed hello interval in terms of delay and control overhead. However, in both approaches there is no clear idea about the neighbours' time-out interval in a node's neighbours-list, which may increase the inaccuracy information in a node's neighbours-list. Using the FLoP algorithm incurs small hello intervals in dense scenarios that cause unnecessary overhead. Also, the FLoPS algorithm may incur more delay due to applying the FLoP algorithm first then performing the two-check process before switching back to the safety mode.

Fuzzy logic-based Dynamic Beaconing (FLDB)

The purpose of the proposed Fuzzy logic-based Dynamic Beaconing (FLDB) [62] is to overcome the drawbacks of using a fixed beaconing interval in position-based routing protocols for MANET. The FLDB algorithm incorporates the concept of Fuzzy Logic controller in its design. The Fuzzy Logic controller is invoked to adjust the time between the transmissions of beacon packets. Researchers in [62] argued that uncertainty and ambiguity associated with node mobility makes the Fuzzy Logic controller the appropriate and proportional choice to alter adaptively the static hello time interval

to be compatible with MANET conditions. An adaptive hello interval using Fuzzy Logic controller would be able to reflect mobility's topology changes by a relative decrease in the frequency of beacon sending rate when node speeds become low and in turn to relatively increasing the frequency of beacon sending rate in case when nodes speed become high. Researchers in [62] used the speed of participant's nodes as input to the Fuzzy Logic controller and the output is the appropriate frequency of beacon sending rate.

The simulation environment and parameters used to evaluate FLDB scheme were not mentioned in the published paper in [62]. FLDB is evaluated in one simulation scenarios under various nodes' speed (1-30 $m\backslash s$). The metrics used to evaluate the performance of FLDB were packet delivery ratio, average control overhead, end-to-end delay and percentage of optimal routes. Compared to the conventional greedy perimeter stateless routing (GPSR) protocol [44], the authors argued that their simulation results proved the superiority of FLDB in terms of overhead, packet delivery ratio and end-to-end delay. However, in the proposed FLDB it is not clear how to tune the frequency of sending hello packets in semi-static networks. Also, there is no clear idea about the neighbours' time-out interval in a node's neighbours-list that may increases the inaccuracy information in a node's neighbours-list.

Several Hello Protocols for Ad-Hoc Networks

Giruka and Singhal [49] proposed four approaches in order to proportionally and appropriately alter the hello sending frequency in position-based routing protocols. The approaches are periodic hello approach, adaptive hello approach, reactive hello approach and event-based hello approach. In each approach researchers seek to answer two main questions; 1) "How often should a node beacon?" and 2) "When should a node timeout its neighbours?".

In the periodic hello approach, nodes perform the classic periodic HELLO protocol. A node that receives a HELLO packet from a specific neighbour creates a new entry for that neighbour in its neighbour-list if it does not have one or updates the existing entry for that neighbour. If a node does not receive a beacon message from a neighbour for a predefined threshold amount of time, it removes the entry for that neighbour from its

neighbour-list.

In an adaptive hello approach, each node simply sends a HELLO each time it has gone through X meters. Thus, a node moves at higher speed and emits a HELLO packet at higher rates and vice versa. To limit the high frequency rate of beaconing for nodes with very high speeds, a MIN-BEACON-INTERVAL is used. The MIN-BEACON-INTERVAL is set to a predefined threshold, if a node exceeds this limit, then it resets its beaconing interval to the MIN-BEACON-INTERVAL. In an adaptive hello approach, based on neighbours' speed and direction a node calculates the timeout period for its neighbours, a node removes a specific neighbour entry with references to the two parameters. Using this mechanism helps nodes to keep their neighbour-lists up to date.

The third beaconing approach is the reactive approach. In this approach, the idea that nodes should build their neighbour-lists only when needed (on demand) was adopted. Thus, when a node needs to send a data packet, it first sends a request HELLO packet and waits during a time t for an answer. Neighbouring nodes with a node transmission range respond to the request by sending unicasting a HELLO response packet to the requesting node. Upon reception of a request HELLO packet, nodes trigger a timeout before answering, to avoid collisions. If no answer is received, then it repeats the same behaviour up to X times.

In the fourth beaconing approach which is the event-based, nodes perform the first classic periodic HELLO approach, but if they do not receive any HELLO packet and do not need to send data packets during a predefined time period, they stop sending a beacon packet until reception of a HELLO packet from others nodes.

To implement the presented schemes, Giruka and Singhal [49] used network simulator Ns2. In the simulation area of 2000 $m \times 600$ m, 100 nodes were distributed randomly. The simulation time was set to 900 s. To generate traffic, a CBR flow was used at a rate of 4 packets per second with packets size of 128 *bytes*. Each flow was started at 150 s and ended at 750 s. The IEEE 802.11 MAC protocol with distributed coordinating function (DCF) was used for all the nodes in the simulation. The radio range of each node was set at 250 *meters*. The Random WayPoint mobility model was utilized in

the simulation. The proposed schemes were evaluated in three simulation scenarios. The first examined the effectiveness of the proposed schemes algorithms under various numbers of nodes (80, 100, 120 and 160 nodes). The second scenario examined the effectiveness of the proposed schemes algorithms under various nodes' speeds (5, 10, 15, 20, 25 and 30 *m/s*). The third scenario examined the effectiveness of the proposed schemes algorithms under various traffic loads (10, 20 30 and 40 *flows*). The metrics that were used to evaluate the performance of the proposed schemes were packet delivery ratio, average control overhead, end-to-end delay and false positives.

Giruka and Singhal [49] stated that the adaptive HELLO scheme is the best overhead/accuracy trade-off from among all HELLO schemes. The adaptive HELLO scheme achieved slightly lower false positives and packet delivery compared to periodic HELLO scheme. The overhead incurred by reactive HELLO scheme is totally depends on the neighbour's entry lifetime in a node's neighbours-list. However, the periodic or proactive hello approach is performed independently of actual data traffic. Thus, the periodic has several drawbacks such as interference with regular data packet, unnecessary utilization of network resources such as nodes' battery power. In the second approach problems may arise in the case in which nodes move as a group. Only a few changes may happen to the network topology as nodes only marginally change their locations relative to other nodes and hence would not need to emit HELLO packet frequently. The drawback of the third approach is that this scheme minimizes the quantity of HELLO packets, which introduces a high latency before sending data packets, and should not be used in networks with high mobility. The drawback of the event-based approach is that some nodes may never be detected by other mobile nodes.

A Turnover Based Adaptive HELLO Protocol (TAP)

Ingelrest et al. [50] proposed a turnover based adaptive HELLO protocol (TAP). A TAP algorithm presents a useful solution to a beacon's HELLO sending frequency. To ensure that all participating nodes are detected in MANET, a TAP algorithm adjusts dynamically the HELLO frequency f_{HELLO} to match the optimal HELLO frequency (f_{opt}) based on the idea of measuring the turnover (r). The turnover r is equal to the changes in a node's number of new neighbours divided by the current sum number of

its neighbours during Δt. If the calculated turnover r is small this means that there are not enough changes in a node's neighbours-list, thus, the f_{HELLO} should be decreased and vice versa. Based on given assumptions, the researchers stated that they theoretically computed f_{opt}.

To implement TAP, Ingelrest et al. in [50] used a homemade simulation tool, using the Unit Disk Graph model. The simulation area was set to 1000 m × 1000 m. TAP was evaluated in one simulation scenarios under various nodes' speeds (2, 3, 4, 5, 6 and 7 $m\backslash s$). The metrics that were used to evaluate the performance of TAP were delay between two HELLO messages and accuracy of neighbourhood tables. Based on the computed value, Ingelrest et al. in [50] experimentally showed that the TAP algorithm gives a good level of accuracy compared to others. Also, the researchers stated that TAP approach might be deployed in any application of MANET that relies upon HELLO packets to maintain nodes' neighbours-lists. However, the TAP algorithm did not present a clear idea on how to track the mobility's topology changes in the network. Also, the algorithm did not present a clear idea on how to remove neighbours' entries from a node's neighbours-list (entry timeout). Both drawbacks may leads to increase in the inaccuracy level in a node's neighbours-list that leads to an incorrect routing decision.

Neighbourhood Lifetime Algorithm (NLA)

Ahmad and Mitton [64] proposed the Neighbourhood Lifetime Algorithm (NLA). NLA algorithm that adapts dynamically the timeout of nodes' entries in a node's neighbours-list based on the frequency of the HELLO packets and the speed of nodes. A neighbour with a high speed will be removed faster than other neighbours; also a neighbour with a low speed will be removed slower than other neighbours. The NLA algorithm adopted the HELLO packet frequency turnover based adaptive HELLO protocol (TAP) [50]. In the NLA algorithm, three possibilities are considered to determine the lifetime of a neighbour in a node's neighbours-list. One, if a neighbour's sending frequency is fixed, then the neighbour's entry timeout is equal to 3 times the sending frequency period. Two, if a neighbour's sending frequency decreases, then the neighbour's entry timeout is equal to the difference between the latest two frequen-

cies. Three, if a neighbour's sending frequency increases, then the neighbour's entry timeout is equal to the absolute value difference between the latest two frequencies.

To implement NLA Ahmad and Mitton in [64] used network WSNet simulator 2. In the simulation area of 500 m × 500 m, 50 nodes were distributed randomly. The simulation time was set to 200 s. The IEEE 802.11 MAC protocol was used for all the nodes in the simulation. The radio range of each node was set at 100 *meters*. The Random WayPoint mobility model was utilized in the simulation. The proposed NLA was evaluated in one simulation scenario. The scenario examined the effectiveness of the proposed NLA algorithm under various nodes' speeds (0, 2, 3, 4, 5 and 6 $m\backslash s$). The metrics that were used to evaluate the performance of NLA were proportion of actual neighbours of a node and false neighbours.

Ahmad and Mitton in [64] stated that experiments conducted to the implement NLA algorithm showed its superiority to other existing algorithms. The NLA algorithm achieved a good improvement of a node's neighbours-entries information that led to decreasing the control overhead in the network. However, as in the TAP algorithm, the NLA algorithm did not present a clear idea of how to track the mobility's topology changes in the network. More drawbacks may occur in NLA algorithm in the scenarios in which nodes move fast within a bounded area and thus do not really change their locations. Also, the NLA algorithm may be suboptimal in case of group mobility.

2.6.2.4 Summary of Prominent Extensions Works for GFS based on Beaconing-Update Scheme

The above-discussed beaconing update schemes have been developed to improve GFS success rate and to guarantee packet delivery. Most proposed beaconing update schemes come at the cost of several drawbacks. Table 2.4 below shows the pros and cons of each discussed scheme.

Table 2.4. Prior Works Regarding Beaconing-Update improvement for GFS

Author	**Protocol**	**Advantages**	**Disadvantages**
Liarokapis and Shahrabi [61, 63]	FLoP and FLoPS	Decreases control overhead compared to a fixed hello interval	Neighbours' time-out interval is not discussed, that may increases the inaccuracy, incurs more delay due to applying FLoP algorithm first then performing two check process to switch to safety mode
Saqour et al. [62]	FLDB	Decreases beacon overhead end-to-end delay and increases delivery	The frequency of beaconing in semi-static networks is not discussed, and Neighbours' time-out interval is not discussed, that may increases the inaccuracy
Giruka and Singhal [49]	Several schemes	A detailed study on the impact of various hello approaches on MANET	**Periodic hello**: Interferences with regular data packet, drain network resources, **Adaptive hello:** Has problems in the case where nodes move as a group **Reactive hello:** Introduces a high latency **Event-based:** Some nodes may never be detected by mobile nodes
Ingelrest et al. [50]	TAP	Provides a good level of accuracy compared to others	Tracking the mobility's topology changes is not discussed, neighbours' time-out interval is not discussed, both drawbacks lead to increase the information inaccuracy in a node's neighbours-list that leads to incorrect routing decision
Ahmad and Mitton [64]	NLA	Improved nodes' information accuracy	Tracking the mobility's topology changes in the network is not discussed, inefficient in high speed within a bounded area and suboptimal in case of group mobility

2.7 Qualitative Discussion and Directions of the Research

Greedy Forwarding Strategy GFS was adopted for use in position-based routing protocols in MANET. GFS was proposed for use in static networks, and this is why it fails when used in MANET. The reasons behind GFS failure in MANET have been discussed extensively in Section 2.5. GFS failure in MANET motivated researchers to improve GFS to be compatible with MANET features. As mentioned in Section 2.6, in this research, the proposed solutions to alleviate GFS failure in MANET are classified broadly into two main categories. These categories were dubbed as Recovery Strategies to Handle GFS Failure RSGF, and Supportive Enhancement for GFS success SEGF besides using a recovery strategy. Key aspects of the discussion were the effectiveness and the efficiency of the proposed solution. The proposed works in RSGF category aimed at improving GFS through using recovery strategies. With SEGF category, those proposed works tried to enhance GFS while still relying on a recovery strategy. Although room for improvement is clear, some shortcomings and problems remain with the reviewed protocols in both categories. Table 2.5 below summarizes the pros and cons of both categories based on what have been stated in Tables 2.2, 2.3 and 2.4.

The core finding here is that not much work has focused on enhancing GFS routing itself to dispense with the need for recovery approaches. This observation motivated the researcher to seek an extension to GFS that not only preserves the property of GFS but also improves its routing success rate.

This research aimed at extending GFS to be compatible with MANET features, because the conventional GFS is unable to handle the complexity of a mobile network as well as the current solutions. In the intended improvement, this researcher seeks to modify GFS to be standalone routing scheme. To the best of the his knowledge, however, no pure prior work about extending GFS as standalone position-based routing protocol currently exists.

The intended improvement for GFS draws inspiration from three reference points. First is that the proposed routing protocol in this research should satisfy the desired properties mentioned in RFC 2501 [116]. These properties for a new or a modified

routing protocol in MANET are that it should be capable of adapting the MANET features, should have a low routing overhead, should be distributed protocol (i.e., it should not be dependent on a centralized controlling node), should guarantee that the routes supplied are loop-free, should support the use of multiple routes that allow quick establishment of routes between communicating nodes and lastly it should support some sort of quality of service. Second, the desired routing protocol should mitigate the reasons behind GFS failure that discussed in Section 2.5. Third is to consider the shortcomings and problems that have appeared in works in RSGF and SEGF categories as discussed in Section 2.6. These three references points shape the underlying conceptual framework of this research.

Table 2.5. Brief Comparison Between RSGF with SEGF Categories

Category	Protocols	Advantages	Disadvantages
RSGF	NEAR [28] GPSR [44] POR [45] GDSTR [55] YAGR [56]	Solve GFS failure at Hole nodes and guarantee delivery	Incur more overhead, long routing path, more delay, consume more bandwidth and energy. Adding network partition problem may occur with sparse and high dynamic networks
SEGF	**Improving Selection Processes** MRSE [57] EGR [41] DR2 [58] SRR [60] PDGR [59] **Improving Beaconing-Update** FLOP and FLoPS [61, 63] FLDB [62] Several schemes [49] TAP [50] NLA [64]	Guarantee delivery based on the used recovey strategy and achieve extra objective besides shortest path	Dead-end problems still occurs that incured the same drawbacks as in RSGF category, incur non-optimal rout, that induces more delay, and more energy consumption

2.8 Research Application Tools and Concepts

This section provides a brief glimpse at the application tools and concepts that are used in this research flow. The following Sub-sections elaborate the discussion on network evaluation techniques, fuzzy logic system, mobility prediction and mobility model.

2.8.1 Network Evaluation Techniques

The network performance evaluation should be accomplished under controlled conditions [116]. The evaluation of network performance can be conducted by using one of two well-known approaches. The first approach uses measurements while the second approach tests the network behaviour via a model [117].

As stated in [117], the measurement techniques consume time and costs, have limitations in terms of several aspects in MANET such as high mobility scenarios and certain mobility models, and requires greater efforts compared to those of the other approaches. Because of those drawbacks, this research applies a simulative technique to conduct the intended evaluation.

Simulation modeling, as stated in [118, 119], can be defined as a technique used to design a model of the theoretical communication network system under study. The design stage is always followed by executing the model, cropping the results and analyzing them using specific network simulation software. To represent the dynamic behaviours and responses, a simulation uses a computer-generated system. And thus, simulation modeling is a more standardized, mature, and flexible tool for modeling various MANET's protocols [120, 121]. Also as stated in [122, 123], simulation permits the study of system behaviour by varying all its parameters, and considering a large spectrum of MANET scenarios. Compared to the measurement techniques, simulative techniques require less time, have lower costs and offer more practicality for all network sizes with various behaviours [124].

In the light of the above discussion, this research adopts Ns2 [125] version 2.33 to explore the performance of the proposed algorithms. The following sub-section provides brief information about Network Simulator Ns2.

2.8.1.1 Network Simulator 2

Network simulator (Ns) is an open source discrete event simulator that is targeted for network research [126]. This research uses the second version of Ns (Ns2), which was released in 1989. Ns2 was developed as part of the Virtual Internet Testbed project (VINT) at University of California Berkeley. The Ns2 packages can be run on different operating systems such as Linux and Windows using Cygwin. Due to its features and modular nature, NS2 has become the most widely used network simulator in the networking research community.

Ns2 is chosen because of the following reasons. First, it is popular open-source software and can be freely downloaded from the website. Second, it gives the user the ability to modify easily the existing algorithm codes. Third, because of its website, a good opportunity exists to share ideas, and a researcher can easily consult many experts in the area. Forth, it is a useful tool to study the functions and protocols (e.g., Transmission Control Protocol (TCP), and routing algorithms) of wired and wireless networks.

The purposes of using a simulation in this research are: 1) to verify the proposed scheme in a simulation environment; 2) to validate the design assumption; and 3) to exercise scenarios to test and to evaluate the performance of the proposed model. To accomplish this, at the beginning, the C++ code for the improved algorithms was written; next, the linkage to OTcl was established. After constructing the components and the intended topology, the simulation was run to obtain the two output files, which are the trace file, and the Network Animator Trace file. The former describes the events that happen throughout the simulation. The latter is issued by the NAM after it takes the trace file that is generated by Ns2 and displays a visual animation of node movement and the packets' activities during the simulation time.

2.8.2 MATLAB

MATLAB [127] is a programming environment; it is the language of technical computing. The MATLAB environment integrates graphic illustrations with precise numerical calculations, and is a powerful, easy-to-use, and comprehensive tool for per-

forming all kinds of computations and scientific data visualization. It can be used to solve technical computing problems faster than other conventional programming languages. MATLAB is used in a variety of applications. The Communication field is one of the well-known applications that use this tool. With Communications, MATLAB is used to test and measure the efficiency and applicability of any proposed protocol [128]. Recently, MATLAB has been considered to be the first choice for many scientists in industry and academia [128].

This research uses the Fuzzy Logic Toolbox that built on the MATLAB numeric computing environment [128]. The Fuzzy Logic Toolbox for use with MATLAB is a tool for solving problems with fuzzy logic. The Fuzzy Logic Toolbox is a collection of functions built on the MATLAB. It provides a graphical user interface (GUI) tools to help entirely accomplish the intent work from only the command line.

2.8.3 Fuzzy Logic Controller

Due to mobility's arbitrary topology changes of MANET, such network is always combined with great randomness and uncertainty [62]. This uncertainty is associated with the routing parameters, such as speed and motion direction. Therefore, the routing process in MANET is subject to unexpected failures. As stated in [129], to address such randomness and uncertainty problems, some researchers in the area of routing protocols for MANET have involved Intelligent Control Systems such as Fuzzy Logic Theory [130].

The Fuzzy Logic Controller (FLC) concept [130] is a mathematical innovation Lotfi Zadeh invented in 1965. FLC is an effective technique for handling fuzzy uncertainties with well-developed mathematical properties. Also, it provides an excellent way to represent and process linguistic variables. Linguistic variables describe some concepts that usually have vague or fuzzy values. Due to its simplicity, convenience, and efficiency, FLC [130] is regarded to be a promising strategy capable of addressing many issues of MANET complexity [131, 132].

Most well known areas of FLC based routing for MANET are: routes costs estimation, movement attribute-aware, energy-aware routing, and quality of service (QoS)-aware

[129]. By utilizing FLC, the routing protocol parameters can be determined more accurately and dynamically and maximize its chances of success. FLC can be easily tuned by adjustment of its membership functions to meet routing requirements. FLC is also well suited to be implemented on mobile nodes because not only is the decision calculation simple, but also so is forming forwarding tables. In this regard FLC is used easily to make routing decisions relying on specific membership functions and a set of fuzzy rules. This is why this research adopts Fuzzy Logic Controller (FLC) as a controller and decision maker concept.

In terms of components, FLC is composed of four blocks, including fuzzification, fuzzy rules, fuzzy inference, and defuzzification [131, 132]. A block diagram that generalizes the fuzzy system is presented in Figure 2.21 below. The following sub-sections present brief information of each block.

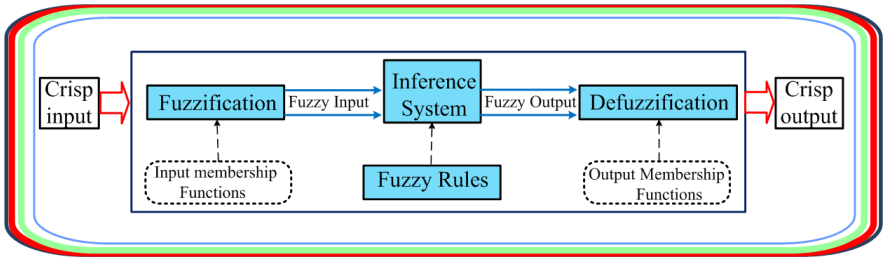


Figure 2.21. The Fuzzy Logic System Architecture

Fuzzification

In fuzzification, the fuzzifier maps each crisp (actual) input value to the corresponding fuzzy sets by using membership functions. Membership function refers to a curve that defines how each value in the crisp input space is mapped to a membership degree between 0 and 1.

Fuzzy Rules

Fuzzy rules consist of logical rules to determine the relationships that exist in the system, between fuzzy sets of its input and output. Usually, fuzzy rules, provided by experts, are based on common sense and are consistent with logic. Formally, the rule-base of FLC can be presented as in Equation 2.1.

$$IF\ (x\ is\ A)\ THEN\ (y\ is\ b) \tag{2.1}$$

where x is the input variable, y is the output variable, A is the fuzzy sets of the input, and B is the fuzzy sets of the output. The IF-part of the "x is A" is called the antecedent, while the THEN-part of the rule "y is B" is called the consequence.

Fuzzy Inference

After the process of fuzzification, the fuzzy inference of the FLC maps input fuzzy sets into output fuzzy sets using the fuzzy rules. Fuzzy inference first evaluates the fuzzy rules and finds out their firing strength. The Mamdani method [133] is one of the most common methods to define the result of a rule. The Mamdani method is the most used method, because state error tends to be zero [134]. With the Mamdani method, the firing strength generated by the antecedent offers the firing strength of a rule. In case the antecedent of a given rule has more than one part (may be either AND or OR operation), the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule.

Defuzzification

The initial state of the defuzzification process is to aggregate all fuzzy output sets for each rule into a single fuzzy set. The aggregation process occurs once, prior to the final step, defuzzification. In defuzzification process, the aggregation output, the single fuzzy set, is considered to be an input and the output is a single crisp output value. The Centroid Weighted Average (CWA) method [135] is considered to be one of the most frequently used methods. In the CWA method, to estimate the crisp value of an output variable, the average of each output of a set of rules is weighed. Equation 2.2 expresses the centroid defuzzification technique [135].

$$crisp_{output} = \Sigma_{j=1}^{n} \mu_j * w_j / \Sigma_{j=1}^{n} \mu_j \tag{2.2}$$

where n is the number of rules, w_j is the centroid weight associated with each rule j, and μ_j is its membership value of the output variable of each rule j.

Fuzzy Logic Membership Functions

Because the membership function defines the fuzzy set, the most key issue in fuzzy sets is the way used to determine fuzzy membership functions [134]. This research uses the Z-shaped, triangular, and S-shaped shapes membership functions. The main features of these functions are their smoothness and that they can be defined using a minimal amount of information due to their simple formulae and computational efficiency [136]. Data relating to the corner points of a function are thereby sufficient to define the functions.

2.8.4 Mobility Prediction Using Dead-reckoning Model

The dead-reckoning technique [137] is used to predict a node's present and future position, by utilizing the direction and the speed of a known past position information. Originally, the idea behind dead-reckoning was to posit that a navigator is able to determine its present position by projecting its past steered direction and speed over current from a known past position, e.g., a port. Dead-reckoning has the ability to determine its future position by projecting an ordered direction and advance speed from a known present position. As a matter of fact, dead-reckoning technique is imbued with the capacity to determine sunrise and sunset as well as predict landfall. It can also evaluate the accuracy of electronic positioning information.

In Figure 2.22 below, it is assumed that the node's last known position at time t_1 is at point A with (x_1, y_1) position coordinates, and the node navigates with direction D and speed S.

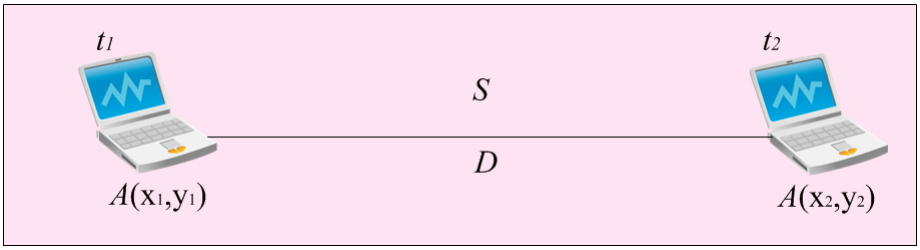


Figure 2.22. Dead-reckoning Position Estimation

By applying dead-reckoning concepts (the laws of sins and cosines), the node can estimate its future position at time t_2 at point A at position (x_2, y_2) coordinates by applying the following equations [137]:

$$A_{x_2} = A_{x_1} + S \cdot \cos D \cdot (t_2 - t_1) \tag{2.3}$$

$$A_{y_2} = A_{y_1} + S \cdot \sin D \cdot (t_2 - t_1) \tag{2.4}$$

where, A_{x_2} represents the estimating position of the node at point A along x coordinate and A_{y_2} represents the estimating position of the node at point B along y coordinate.

By using Mobility prediction dead-reckoning-based concepts, the participating nodes in MANET have the ability to estimate its present or future position, based on known direction and speed of past or present position.

2.8.5 Mobility Model

Mobility model mimics the movement of the mobile nodes within the network [138]. The choice of a mobility model can affect significantly the outcome of a simulation study [139]. It has been noticed in [140] that the selected mobility pattern is considered as a selective issue. It should accurately characterize the planned scenario in which the proposed protocol is likely to be used.

Generally, most simulated MANET's routing protocols rely on randomized mobility models [141], especially on the Random WayPoint mobility model (RWP) [77]. The RWP model was first used by Johnson and Maltz in the evaluation of Dynamic Source Routing (DSR) Protocol [77]. As stated in [139], in this model, the motion of a node is characterized by two parameters: the pause time and the speed. These two parameters determine the movement of a node. when the speed is low and the pause time is long, the network topology becomes stable, and vice versa.

As with other mobility models, RWP is used to identify the way in which nodes move in the simulation area. As stated in [139], in this model, each node moves with speed randomly selected from a minimum and maximum speed interval, i.e., $[v_{min}, v_{max}]$, towards its destination. The random selected speed and direction values of a node's motion in the current movement bear no relation to their past values in the previous movement [140]. A node's destination is also selected randomly by the node itself. After the node reaches its destination it pauses for random time before it proceed to

another destination using the same procedures.

A simulation study in [142] proved that nodes moving according to the RWP model have a tendency to concentrate in the centre of the simulated area that results in hot-spot problem. Also, due to the random distributions of the nodes and the variation in their speed during the simulation progresses, the performance of the protocol under study will be varied as well. In particular, the initial collected performance results may differ from the performance results collected later in the simulation. The most used method to solve these problems has been to be assigned a steady-state period in which the results are discarded [81]. As stated in [143] this method has two main drawbacks. First, this method requires the discarding of some simulation results thus, it is inefficient. Second, it is difficult to define how long a safely steady-state to be assigned.

To address the problems accompany with the traditional RWP, Navidi and Camp [143] proposed an improved RWP model without pausing to model nodes' mobility. The simulation results in [143], proved the effectiveness of the improved RWP model with zero pause time and zero steady-state initialization. This solution for the steady-state and pause time problems of the RWP is reasonable and has been used widely in literature [45, 144, 145].

To sum up, this research uses the improved RWP model without pausing [143] to model nodes' mobility. To achieve variant topology changes, this research apply different range of nodes' speed. Finally, as the same as in [45, 144, 145], this research collects the simulation results without using the steady-state approach (i.e., steady-state is set to zero).

2.9 Summary of Chapter

Thus far this chapter has provided a comprehensive overview of the position-based routing protocols in MANET. Existing forwarding strategies used by position-based routing protocols, were also highlighted. The advantages of the Greedy Forwarding Strategy (GFS) over other forwarding strategies were pointed out.

Problems that results in GFS failure when using in MANET were also described and reviewed in-depth. In addition, existing research efforts in solving GFS failure was also highlighted. The need for an improvement in the basic GFS, which could potentially improve the efficiency of the overall position-based routing protocol, was also demonstrated. Finally, the chapter also provided brief knowledge of the application tools and concepts used in this research flow.

To sum up, this chapter is concerned with all factors required to improve current GFS in MANET. To receive full benefit from this survey, this researcher determined most desired features that should be available to modify GFS as standalone routing protocol. This present effort aims to improve GFS success rate with low overhead, decreased end-to-end delay, and increased packet delivery ratio. As far as this researcher knows, not much work has focused on developing GFS itself. Next chapter will present the design and implementation of the first mechanism that shapes the first objective of the proposed routing protocol.

CHAPTER THREE

DESIGN AND IMPLEMENTATION OF THE DYNAMIC BEACONING UPDATE MECHANISM

3.1 Overview

This chapter elaborates the design and implementation of the first objective. The proposed solution is termed the Dynamic Beaconing Update Mechanism (DBUM). The rest of this chapter is arranged in the following order: Section 3.2 shows DBUM design and objectives. Sections 3.3 and 3.4 introduce the compulsory update techniques (CUT) and the neighbourhood matrix entries management (NMEM) detailed design respectively. Sections 3.5 introduce DBUM's functionality followed by the deduction of the workflow. Section 3.6 introduces the implementation of DBUM in a simulated environment. Section 3.7 introduces the optimal tolerance deviation distance and fuzzy logic-delay calculation. Finally Section 3.8 summarizes the chapter.

3.2 Dynamic Beaconing-Update Mechanism Detailed Design and Objectives

The overall purpose of the Dynamic Beaconing Update Mechanism (DBUM) is to dynamically ensure and maintain up-to-date neighbours' status information in the network. By achieving this goal, the performance of routing protocols under study can be thoroughly improved. Retransmission and rerouting that are required for the case of inaccuracy of position information, which increases the packet loss probability, either can be prevented totally or decreased to the lowest rate.

DBUM scheme integrates two coherent techniques, namely, the Compulsory Update Technique (CUT), and Neighbourhood Matrix Entries Management (NMEM) technique. CUT is designed to react fast enough to the frequent mobility's topology changes to emit an urgent beacon message. The CUT technique cooperates with the second part of DBUM which is NMEM. NMEM is proposed to ensure the refreshment of the entries in nodes' neighbourhood's location-matrix.

Each of these techniques has its own structure, and special usage objectives through the

routing process that will be discussed separately in this chapter. The detailed design of DBUM is shown in the model's structure in Figure 3.1 below.

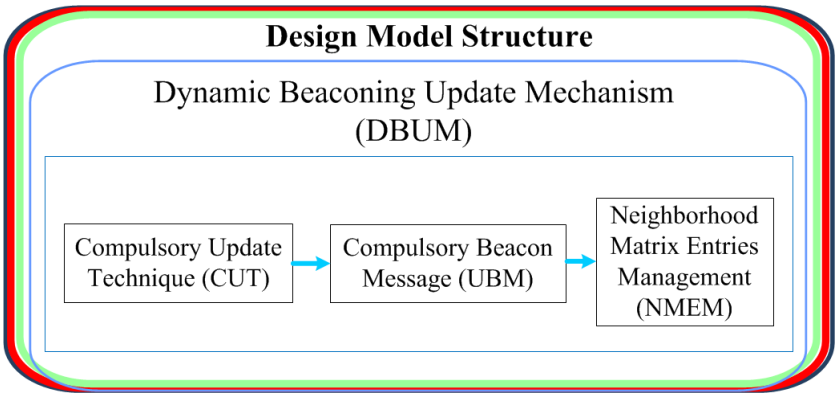


Figure 3.1. DBUM Model Structure Details

To make it easier for a node to distinguish between the varieties of received packets, each sent packet holds a sole byte flag called the Packet Distinction Number (PD_n). The PD_n is a unique number assigned to each given packet.

3.3 Compulsory-update Technique Detailed Design

As noted in Chapter 2, Section 2.5, fixed beaconing-update and unpredictability of nodes' mobility are two main factors decreasing nodes' location accuracy in the network. This resulting in greedy forwarding strategy (GFS) failure when used in MANET. The compulsory update technique (CUT) is designed to mitigate the GFS failure by increasing the location accuracy of the participating nodes in the network. The CUT accounts for the location inaccuracy problem induced by arbitrary motion of participating nodes in MANET. It is performed even in scenarios in which no data packet is communicated between the nodes. All participating nodes execute CUT to generate an Urgent Beacon Message (UBM). UBM is generated by the node that makes considerable changes in its motion characteristics that are different from those reported in the latest update packet.

3.3.1 Urgent Beacon Message Structure

Conventional Beacon Packet (BP) consists of three fields; node identity (*ID*), position information as (x, y) coordinates, and sending time [49, 50]. In the current techniques,

the used fixed beacon packet interval time (FBPIT) to send BP fluctuates is between $1\,s$ and $10\,s$. This value reflects the mobility status of the network. If the mobility's topology change is very high, beaconing will be very short and vice versa [62]. In this research, the researcher made some alterations to conventional BP. It is constructed using eight cells divided into four main fields. Table 3.1 below illustrates the proposed urgent beacon message (UBM) sent by node i at time t_s.

Table 3.1. Urgent Beacon Message (UBM) Structure, the Shaded Columns are the New Proposal and the Rest as the Old Packet

1	2				3	4
ID_i	x	y	*vel.*	*dir.*	t_s	PD_n
	x^i_{ts}	y^i_{ts}	v^i_{ts}	θ^i_{ts}		1
4Bs	4Bs	4Bs	1B	1B	1B	1B

The proposed UBM size (BP*size*) is 16 bytes. Compared with BP, the increase in UBM packet is just 3 bytes. As shown in the table, FBM holds the following fields: node identity (ID_i) e.g., IP address, which are unique properties of a node, the geographical position of the node as (x, y) coordinates, the node's velocity (v), (θ) is the node's motion direction with respect to x-axis, t_s is the beacon sending time, and packet distinction number (hold #1). The Bs symbol refers to bytes.

3.3.2 Urgent Beacon Message Design Goal

The Compulsory update technique (CUT) would be able to adaptively track network mobility by adjusting the UBM sending frequency based on nodes' mobility rate. A node that makes considerable changes in its motion characteristic since its most recent update generates the UBM. UBM is considered to be an urgent message for the local topology changes of the nodes in the network. Thus, UBM maintains more accurate location information in a high mobility environment, in which important data forwarding activities are on-going. In this regards, using CUT to transmit UBM attains better results in improving MANET's reliability while keeping the control traffic low.

3.3.3 Scheduling Time to Emit Urgent Beacon Message

In the proposed CUT, the need to send UBM is related totally to a node's deviation distance, occurring over time, which has considerable negative influence on connectivity and link breaks. A deviation distance is reached faster, as a node is moving fast, thus localization process and updating decision are performed more often and vice versa. The CUT adaptively optimizes the maximum time period that can transpire before the node broadcasts a new UBM. Thus, CUT achieves trade-off sending updating messages carefully to represent the real needs for updating.

The CUT is built based on the correlation between the speed of nodes and their deviation distance. To design a reliable CUT, issues that need to be clarified are: 1) when should a node execute the check process to find out its deviation distance; 2) how can the node relate the check time with the level of its mobility value; 3) how to calculate a node's deviation distance over time; and 4) what is the acceptable deviation distance that, when a node exceeds that distance, it should broadcast UBM. The answers to these questions are discussed in the following sub-section.

3.3.4 Check Time Identification by Using Fuzzy Logic

In this section, the proposed beaconing technique is improved using Fuzzy Logic Controller (FLC). As stated in [129], routing protocol parameters in MANET can be optimized adaptively using the FLC [130]. FLC assists in determining those parameters more accurately and dynamically [60, 62, 136].

The Compulsory update technique (CUT) is adapted by utilizing the maximum period of time that can run out before the next check operation. The time period utilization is built upon the reciprocal relationship between check time (CHT) and the node speed (NS). Mobile nodes with high mobility value achieve deviation distance at high rates. This should increase the number of check processes by reducing CHT. Similarly, nodes moving at slow mobility achieve deviation distance at low rates, thereby decreasing the number of check processes by increasing CHT. This adaption could achieve a good balance between acceptable error tolerance, and position information accuracy in a nodes' neighbourhood's location-matrix (NLM).

The CUT is improved by using the Dynamic Fuzzy Logic Controller Check-time (DFLCH). In DFLCH, NS and CHT are used as input parameter and output parameter respectively. Thus, DFLCH approach as FLC, selects the best CHT based on NS mobility metric. Figure 3.2 below shows the fuzzy logic design of the DFLCH model.

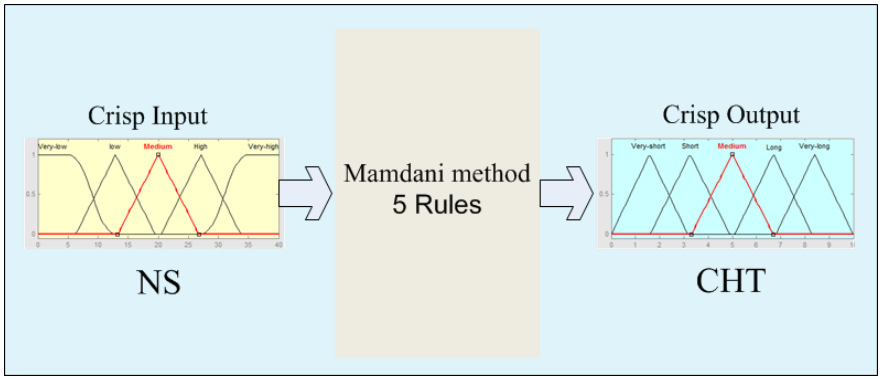


Figure 3.2. Generalized Fuzzy System FLC for DFLCH

3.3.4.1 Fuzzify Node Speed Input

The used membership functions for the DFLCH model are the Z-shaped, triangular, and S-shaped shapes. As stated in [60, 62, 136], the main features of these functions are their smoothness and that they can be defined using a minimal amount of information. Adjacent fuzzy sets are overlapped between 10% and 50% [131]. As the overlapping ratio increases, the mapping function becomes smoother. Fuzzy sets, which have too much overlapping that is exceeding 50%, or do not have overlapping at all, blur the distinction in the control action [131]. Fuzzy sets can be equal or unequal and based on this, the overlapping ratio between them also could be equal or unequal [134].

The crisp input values of NS are fuzzified between NS-*min* = zero *m*/*s* and NS-*max* = 40 *m*/*s*. As stated in [146], this speed range is the participants' common range in MANET. The linguistic values of NS are: very slow (*vsl*), slow (*sl*), medium (*m*), high (*h*), and very high (*vh*). Table 3.2 and Figure 3.3 below show the assignment of range and degree of membership functions for input NS linguistic variable respectively.

Table 3.2. Linguistic variable: Node Speed (NS)

NS numerical range *m/s*	**Linguistic value**	Notation
Z-curve parameters (a, b, c) = (0, 6.6, 13.2)	*very slow*	*vsl*
Tria.-curve parameters (a, b, c) = (6.7, 13.3, 19.9)	*slow*	*sl*
Tria.-curve parameters (a, b, c) = (13.4, 20, 26.6)	*medium*	*m*
Tria.-curve parameters (a, b, c) = (20.1 , 26.7, 33.3)	*high*	*h*
S-curve parameters (a, b, c) = (26.8, 33.4, 40)	*very high*	*vh*

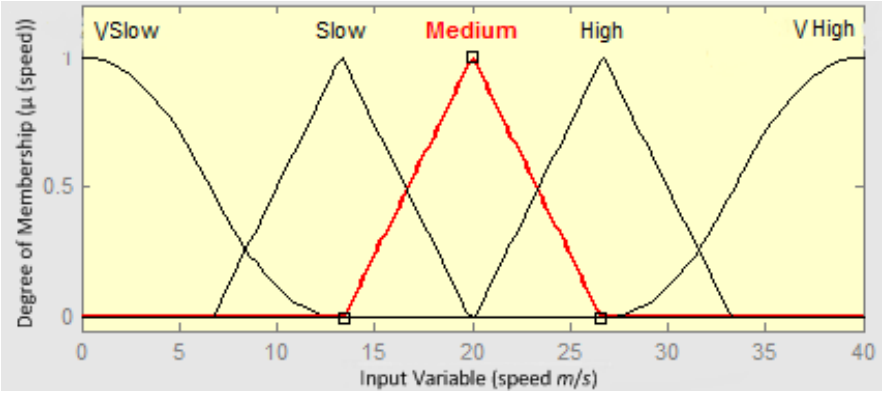


Figure 3.3. Membership Functions of NS Input Variable

In Table 3.2 above, NS input values can be interpreted as: "very slow" as "a speed below about 6.6 *m/s*", "slow" as "a speed close to 13.3 *m/s*", "medium" as "a speed close to 20 *m/s*", "high" as "a speed close to 26.7 *m/s*" and "very high" as "a speed above about 33.4 *m/s*". Otherwise, the speed can located in two ranges, for example if the node speed is 9 *m/s*, then node speed has a very low membership value of 0.20248 and it has a slow membership value of 0.34848. The example in Section 3.3.5 clarifies these calculation processes.

Based on the above discussion and the allowed overlapping ratio [131], the researcher uses 49.24% overlapping ratio between adjacent fuzzy sets of NS. The following equations gives the explicit formulas for the selected membership functions of NS, i.e., $\mu(NS)$, based on [130, 131, 132].

$$\mu(NS_{vsl}, 0, 6.6, 13.2) = \begin{cases} 1, & NS \leq 0 \\ 1 - 2\left(\frac{NS-0}{13.2-0}\right)^2, & 0 \leq NS \leq 6.6 \\ 2\left(\frac{NS-13.2}{13.2-0}\right)^2, & 6.6 \leq NS \leq 13.2 \\ 0, & NS \geq 13.2 \end{cases} \tag{3.1}$$

$$\mu(NS_s, 6.7, 13.3, 19.9) = \begin{cases} \left(\frac{NS-6.7}{13.3-6.7}\right), & 6.7 \leq NS \leq 13.3 \\ \left(\frac{19.9-NS}{19.9-13.3}\right), & 13.3 \leq NS \leq 19.9 \\ 0, & otherwise \end{cases} \tag{3.2}$$

$$\mu(NS_m, 13.4, 20, 26.6) = \begin{cases} \left(\frac{NS-13.4}{20-13.4}\right), & 13.4 \leq NS \leq 20 \\ \left(\frac{26.6-NS}{26.6-20}\right), & 20 \leq NS \leq 26.6 \\ 0, & otherwise \end{cases} \tag{3.3}$$

$$\mu(NS_h, 20.1, 26.7, 33.3) = \begin{cases} \left(\frac{NS-20.1}{26.7-20.1}\right), & 20.1 \leq NS \leq 26.7 \\ \left(\frac{33.3-NS}{33.3-26.7}\right), & 26.7 \leq NS \leq 33.3 \\ 0, & otherwise \end{cases} \tag{3.4}$$

$$\mu(NS_{vh}, 26.8, 33.4, 40) = \begin{cases} 0, & NS \leq 26.8 \\ 2\left(\frac{NS-26.8}{40-26.8}\right)^2, & 26.8 \leq NS \leq 33.4 \\ 1 - 2\left(\frac{NS-40}{40-26.8}\right)^2, & 33.4 \leq NS \leq 40 \\ 1, & NS \geq 40 \end{cases} \tag{3.5}$$

3.3.4.2 Fuzzify Check Time Output

Fuzzy sets for the CHT output variable have the following names: very short (*vs*), short (*s*), medium (*m*), long (*l*), and very long (*vl*). Table 3.3 and Figure 3.4 below show the assignment of range and membership functions for CHT output variable respectively. In conventional beaconing-update schemes, based on the nodes' mobility degree, FBPIT fluctuates between 1*s* and 10*s* to send a beacon [62]. In the proposed beaconing-update CUT this interval is adopted to figure out the CHT intervals. Hence, CHT is fuzzified between *CHT-min* = *zero* and *CHT-max* = 10*s*.

Table 3.3. Linguistic variable: Check Time (CHT)

CHT **numerical range** *s*	**Linguistic value**	Notation
Tria.-curve parameters (a, b, c) = (0 1.6 3.2)	*very short*	*vs*
Tria.-curve parameters (a, b, c) = (1.7 3.3 4.9)	*short*	*s*
Tria.-curve parameters (a, b, c) = (3.4 5 6.6)	*medium*	*m*
Tria.-curve parameters (a, b, c) = (5.1 6.7 8.3)	*long*	*l*
Tria.-curve parameters (a, b, c) = (6.8 8.4 10)	*very long*	*vl*

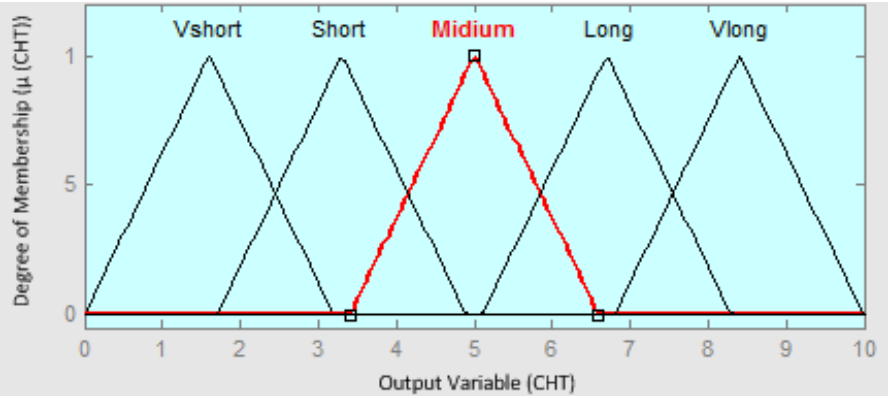


Figure 3.4. Membership Functions for CHT Output Variable

The CHT output values can be interpreted as: "very short" as "a check time close to 1.6 *s*", "short" as "a check time close to 3.3 *s*", "medium" as "a check time close to 5 *s*", "long" as "a check time close to 6.7 *s*" and "very long" as "a check time close to 8.4 *s*". Otherwise, the CHT can located in two range, for example if the CHT is 3 *s*, then CHT has short membership value of 0.8125 and it has very short membership value of 0.125. The example in Section 3.3.5 clarifies these calculation processes.

Based on the above discussion and the allowed overlapping ratio [131], this research uses 46.875% overlapping ratio between adjacent fuzzy sets of CHT. The following equations gives the explicit formulas for the selected membership functions of CHT, i.e., $\mu(CHT)$, based on [130, 131, 132].

$$\mu(CHT_{vs}, 0, 1.6, 3.2) = \begin{cases} \left(\frac{CHT-0.0}{1.6-0.0}\right), & 0.0 \leq CHT \leq 1.6 \\ \left(\frac{3.2-CHT}{3.2-1.6}\right), & 1.6 \leq CHT \leq 3.2 \\ 0, & otherwise \end{cases} \tag{3.6}$$

$$\mu(CHT_s, 1.7, 3.3, 4.9) = \begin{cases} \left(\frac{CHT-1.7}{3.3-1.7}\right), & 1.7 \leq CHT \leq 3.3 \\ \left(\frac{4.9-CHT}{4.9-3.3}\right), & 3.3 \leq CHT \leq 4.9 \\ 0, & otherwise \end{cases} \tag{3.7}$$

$$\mu(CHT_m, 3.4, 5, 6.6) = \begin{cases} \left(\frac{CHT-3.4}{5-3.4}\right), & 3.4 \leq CHT \leq 5 \\ \left(\frac{6.6-CHT}{6.6-5}\right), & 5 \leq CHT \leq 6.6 \\ 0, & otherwise \end{cases} \tag{3.8}$$

$$\mu(CHT_l, 5.1, 6.7, 8.3) = \begin{cases} \left(\frac{CHT-5.1}{6.7-5.1}\right), & 5.1 \leq CHT \leq 6.7 \\ \left(\frac{8.3-CHT}{8.3-6.7}\right), & 6.7 \leq CHT \leq 8.3 \\ 0, & otherwise \end{cases} \tag{3.9}$$

$$\mu(CHT_{vl}, 6.8, 8.4, 10) = \begin{cases} \left(\frac{CHT-6.8}{8.4-6.8}\right), & 6.8 \leq CHT \leq 8.4 \\ \left(\frac{10-CHT}{10-8.4}\right), & 8.4 \leq CHT \leq 10 \\ 0, & otherwise \end{cases} \tag{3.10}$$

3.3.4.3 Fuzzy Rules and Fuzzy Inference

The following are the proposed fuzzy rules that can be used by fuzzy inference to map the fuzzy NS input sets: *very slow*, *slow*, *medium*, *high*, and *very high* into fuzzy CHT output sets: *very long*, *long*, *medium*, *short*, and *very short*.

RULE **1**: IF NS is *very slow* THEN CHT is *very long*

RULE **2**: IF NS is *slow* THEN CHT is *long*

RULE **3**: IF NS is *medium* THEN CHT is *medium*

RULE **4**: IF NS is *high* THEN CHT is *short*

RULE **5**: IF NS is *very high* THEN CHT is *very short*

The five rules (RULE 1 to RULE 5) can be combined together to control CHT adaptively based on NS.

3.3.5 An Illustrative Example for Dynamic Fuzzy Logic Controller Check-time

This section presents an example to explain the operations of FLC used for the DFLCH model. This example is built using the proposed five fuzzy rules in Sub-section 3.3.4.3. Also, NS input variable range fuzzifies between *NS-min* = *zero* and *NS-max* = 40 *m/s* to the fuzzy sets shown in Table 3.2 and CHT output variable range fuzzifies between *CHT-min* = *zero* and *CHT-max* = 10*s* to the fuzzy sets shown in Table 3.3 above. As an illustrative example, suppose that the current node NS=25 *m/s*, is the input value.

Step 1. Fuzzify the inputs: As shown in Figure 3.5 below, the first step is to take the crisp input NS=25.0 *m/s* and determine the degree to which it belong to each of the appropriate fuzzy sets via membership functions (*very slow, slow, medium, high,* and *very high*). The 25 is a numerical value limited to the universe of discourse of the NS input variable (in this example the interval between 0 and 40) and the output is a fuzzy degree of membership (always the interval between 0 and 1). So fuzzification really doesn't amount to anything more than table lookup or function evaluation. In this example, the rating of speed = 25 m/s corresponds to two membership functions: *medium* and *high* with value 0.2424 and 0.7424, respectively. These fuzzy inputs (i.e., 0.2424, 0.7424) are then passed on to the next step, Rule Evaluation.

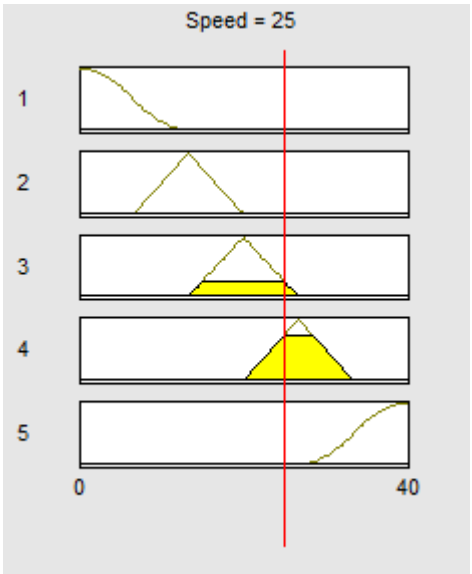


Figure 3.5. Fuzzification of the NS=25 *m/s*

Step 2. Apply fuzzy inference: Once the NS = 25 have been fuzzified, the degree to which each part of the antecedent has been satisfied for each rule becomes clear.

Hence, in the second step, the fuzzy inference takes the fuzzified inputs, $\mu_{(NS=medium)} = 0.2424$, and $\mu_{(NS=high)} = 0.7424$ and apply them to the antecedents of the fuzzy rules. The fuzzy inference evaluates all the five fuzzy rules (**RULE 1** to **RULE 5**). In this example, because the NS input is 25.0 m/s, two rules will be fired (**Rule 3** and **Rule 4**).

RULE **3**: IF NS is *medium* (0.2424) THEN CHT is *medium* (0.2424)

RULE **4**: IF NS is *high* (0.7424) THEN CHT is *short* (0.7424)

Step 3. Aggregation of the rule outputs: In the aggregation process, the membership functions of all fired rule consequents are combined into a single fuzzy set. Thus, the input of the aggregation process is the list of consequent membership functions, and the output is one fuzzy set for each output variable. Figure 3.6 below shows how the output of each rule is aggregated into a single fuzzy set for the overall fuzzy output.

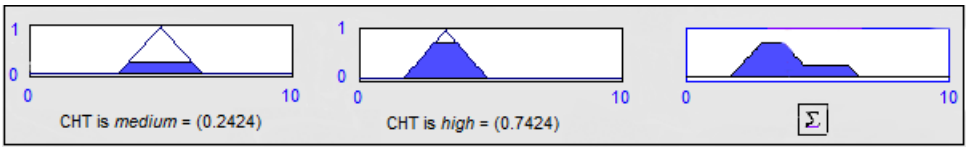


Figure 3.6. Aggregation of the Rule Consequents

Step 4. Defuzzify the outputs: Defuzzification is the last step in the fuzzy inference process. To defuzzify the aggregate fuzzy set, this research uses the Centroid Weighted Average (CWA) method [135], which is the most frequently used method in fuzzy applications. The input for the defuzzification process is the aggregate output fuzzy set from step 3 and the output is a single crisp number, i.e., CHT.

The CWA method finds the point where a vertical line would slice each component of the aggregate set into two equal masses, i.e., the centroid of each set. The final CHT found be calculation and the toolbox is as shown in Figure 3.7 and Equation 3.11 below.

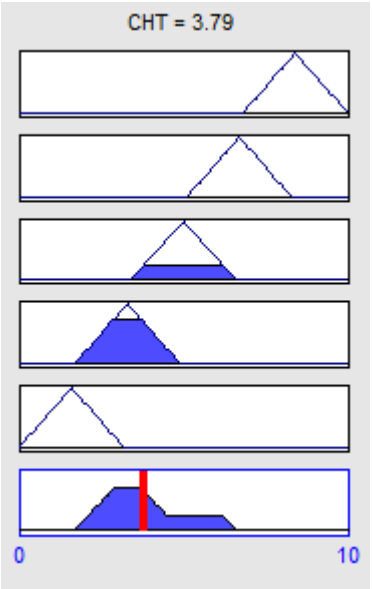


Figure 3.7. Defuzzification of CHT Output

$$CHT = \Sigma_{j=1}^{n} \mu_j * w_j / \Sigma_{j=1}^{n} \mu_j =$$

$$\frac{((0.0*8.4)+(0.0*6.7)+(0.2424*5)+(0.7424*3.3)+(0.0*1.6))}{(0.0+0.0+0.2592+0.6818+0.0)} = 3.72\,s \tag{3.11}$$

where n is the number of rules, w_j is the centroid weight associated with each rule j, and μ_j is its membership value of the output variable of each rule j.

3.3.6 Deviation Distance Identification

A node could scarcely keep consistent movement characteristics in MANET because it can move with random speed and direction [109]. Therefore, error distances, say (ω), mostly accompany each node compared to its latest emitted beacon update. From a node's perspective, the ω represents the difference between its current (actual) and the expected (false) locations, depending upon its latest sent beacon update after a specific period of time. The time that assigned for a node to make these calculation is the check time. A schematic illustration to estimate a node's deviation distance ω is introduced in Figure 3.8 below.

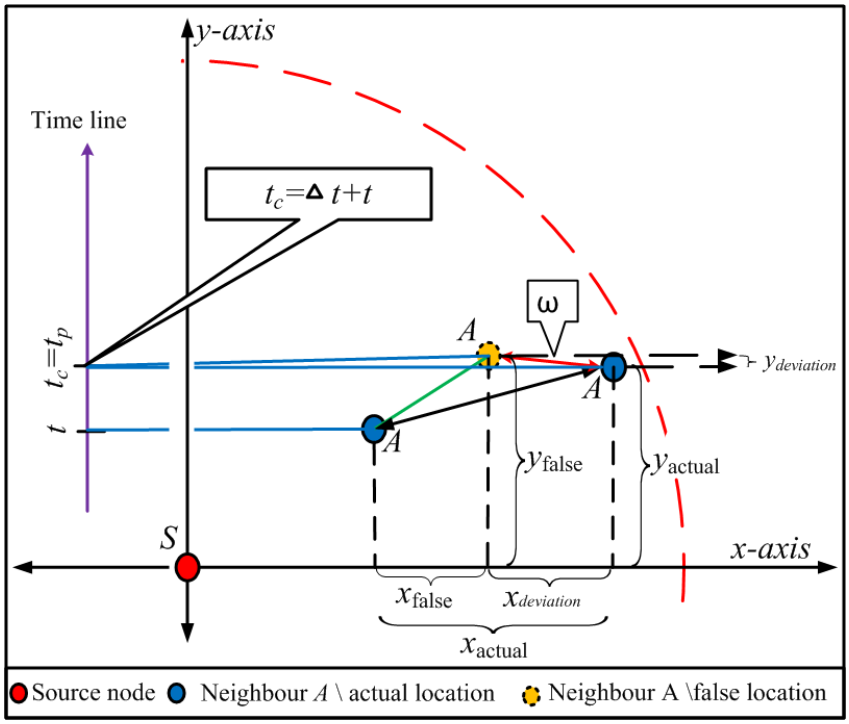


Figure 3.8. The Actual Deviation Distance ω Estimation Process

In Figure 3.8 above, it is assumed that node A sent the latest beacon update to its neighbours at time t with its current status $(ID_A, x_t^A, y_t^A, v_t^A, \theta_t^A)$. Also, it is assumed that the periodic CHT for A is scheduled at time t_c at which node A carries out localization and gets its current (actual location) with the following information, $(x_{t_c}^A, y_{t_c}^A, v_{t_c}^A, \theta_{t_c}^A)$. At the same time, node A predicts its position (false location) regarding its information sent in the latest sent beacon update $(x_{t_p}^A, y_{t_p}^A)$ where, $t_p = \Delta t + t = t_c$.

At time t_c, node A predicts its expected location at the same time of CHT, by using the self-location prediction scheme (SLPS). SLPS formula is inspired from [109] based on the dead-reckoning approach [137]. In both works, the authors used a liner speed in the prediction process. In the proposed SLPS, this research takes into account non-linear node motion with changes in the node's direction and speed (acceleration). The alterations are stated below in Equations 3.12, 3.13 and 3.14.

At the beginning, node A calculates its acceleration over the elapsed time, i.e., t_p and t.

$$a_{t_p-t}^A = \frac{v_{t_p}^A - v_t^A}{t_p - t} \tag{3.12}$$

Now, node A calculates its future location using the following equations.

$$x_{t_p}^A = x_t^A + v_{x_t}^A \cdot (t_p - t) + \frac{1}{2} a_{x_t}^A \cdot (t_p - t)^2 \tag{3.13}$$

$$y_{t_p}^A = y_t^A + v_{y_t}^A \cdot (t_p - t) + \frac{1}{2} a_{y_t}^A \cdot (t_p - t)^2 \tag{3.14}$$

where the expected position at time t_p of node A at *x-axis* is $x_{t_p}^A$ and at *y-axis* is $y_{t_p}^A$. Also, $v_{x_t}^A$ and $v_{y_t}^A$ are the speed values of node A along the *(x* and *y) axis* respectively, where $v_{x_t}^A = v_t^A \cdot \cos\theta$ and $v_{y_t}^A = v_t^A \cdot \sin\theta$, $a_{x_t}^A$ and $a_{y_t}^A$ are the acceleration values of node A along the *x-axis* and *y-axis* respectively, where, $a_{x_t}^A = a_t^A \cdot \cos\theta$ and $a_{y_t}^A = a_t^A \cdot \sin\theta$. Further, where v_t^A is the speed and a_t^A is the acceleration values of node A reported in latest beacon update message at time t.

Based on the information about the three locations of A, early co-ordinates (x_t^A,y_t^A) reported in latest beacon update message, expected node's A position ($x_{t_p}^A$,$y_{t_p}^A$) (false position), and the latest (actual) position obtained from location service provider ($x_{t_c}^A$,$y_{t_c}^A$), node A now can calculate its ω deviation distance by using the Pythagorean Theorem formula [147] as in Equation 3.15 below. The deviation distance ω represents the distance between actual position and the predicted position.

$$\omega = \sqrt{\left(x_{t_c}^A - x_{t_p}^A\right)^2 + \left(y_{t_c}^A - y_{t_p}^A\right)^2} \tag{3.15}$$

3.3.7 Tolerance Deviation Distance and Beaconing Decision Making

The tolerance deviation distance (TOD) limits the deviation distance to acceptable value. Thus, TOD acts as a trigger for a node to emit UBM and has considrable influence on the performance of the CUT. To figure out the most appropriate TOD, the researcher conducted several experiments, the results are stated in Section 3.7.1. The simulation results in Figures 3.17, 3.18 and 3.19 showed that when TOD value is closed to 12 meters, it presents a reasonable overhead and high consistency of a nodes' neighbourhood's location-matrix. Hence, this research consider TOD with 12 meters value that can achieves a good balance that regulates the beaconing frequency

and nodes's status information accuracy in the network.

Based on the calculated deviation distance ω and TOD, a checker node can execute the intended comparison to decide whether it should initiate UBM or not. As shown in Figure 3.8, node *A,* named, the checker node, should transmit a UBM to the broadcast MAC address, if and only if, the condition in Equation 3.16 below is satisfied.

$$\omega > TOD \tag{3.16}$$

In a case in which a node maintains constant motion speed and direction during the estimated check time CHT, it will take a long time to exceed the tolerance deviation distance TOD that is required to emit an UBM. This will result in increasing its information inaccuracy in its neighbours' neighbourhood's location-matrix. To handle this problem, in a CUT approach, if a node did not send a beacon during the last $10\,s$, it should immediately emit a new UBM. A sender node that transmits UBM packet again performs a DFLCH algorithm to calculate the next CHT to reset its timer. As the timer becomes zero a node follows the same previous mentioned procedures to transmit the next new UBM beacon.

3.4 Neighbourhood Matrix Entries Management Detailed Design

With traditional beaconing-update, if a node does not get a beacon from its neighbour, within specific threshold time, it will remove that neighbour from its NLM. The threshold time is a multiple of fixed beacon intervals. The accuracy of location information in a nodes' NLM is proportional to the threshold period. However, as mentioned in Chapter 2, Section 2.5 such an approach still has several drawbacks. The listed information in a node's NLM might be inaccurate due to several reasons . To solve this problem, the frequency at which an entry is considered to be stale, should be tuned and not be considered to be a fixed pre-specified time.

This section presents Neighbourhood Matrix Entries Management (NMEM). NMEM collaborates with CUT scheme to ensure the refreshment of a node's NLM. NMEM aims to adapt dynamically the lifetime of entries (ELT) in a node's NLM with regards to node-neighbours link lifetime. Thus, NMEM allows reliable and timely removing of neighbours' entries from a node's NLM.

In this research, the link lifetime between nodes is dubbed as the Residual Links Lifetime (RLT). With NMEM, a node estimates RLT for each neighbour according to the relative velocity between them. The ELT for a neighbour is proportional to its RLT with a node. The ELT timer helps in determining the neighbour's existence in a node's transmission range. By accomplishing this, the nodes' NLM can be consistent and more efficient.

3.4.1 Residual Link Lifetime Identification

In this research, it is assumed that the transmission range (R) of all nodes is the same. In MANET, a pair of mobile nodes (say node i and node j) can directly communicate if the maximum distance between them is less than R. Residual link lifetime RLT between mobile nodes i and j can be defined as the maximum time of connectivity between the two nodes, before one of them leaves the transmission range of the other one [106, 107, 108]. Figure 3.9 below presents a clarification to show how to calculate RLT between two neighbours nodes.

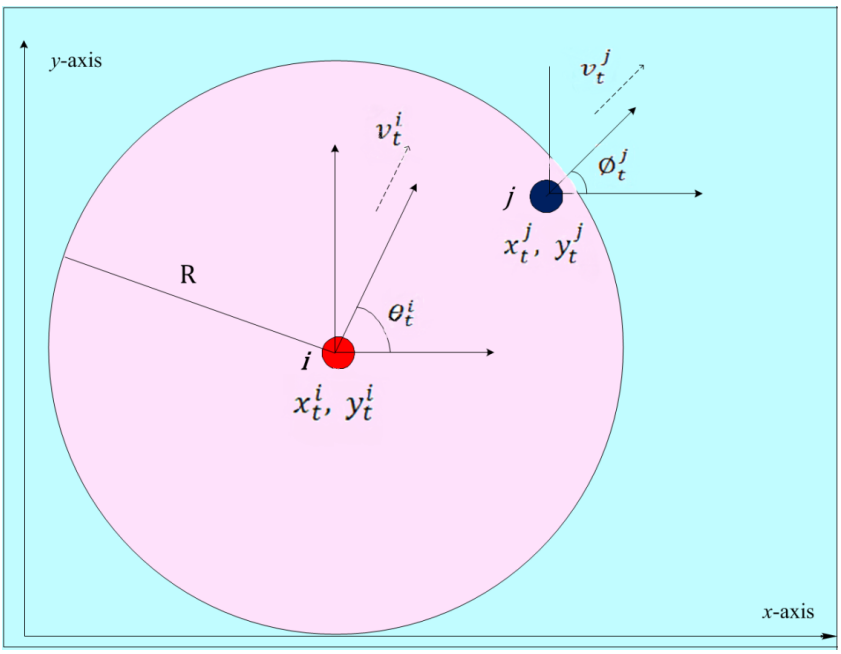


Figure 3.9. Communication Relation and RLT of A Pair of Nodes

As illustrated in Figure 3.9 above, it is assumed that nodes i and j are neighbours. Also, it is assumed that the current information of node j, as reported in its latest update for node i is (ID_j, x_t^j, y_t^j, v_t^j, θ_t^j) at time t. At the same time, node i's information is (ID_i, x_t^i, y_t^i, v_t^i, θ_t^i).

To estimate RLT between two nodes, work presented by Bai et al. in [148] is adopted as shown in Equation 3.17 below.

$$RLT_j^i(t) = \left(R \mp des_j^i(t) / RV_j^i(t)\right) \tag{3.17}$$

where, $RLT_j^i(t)$ is the residual lifetime of the link between node i and j at time t, R is the transmission range of the nodes, $des_j^i(t)$ is the current distance between node i and j at time t, and $RV_j^i(t)$ is the magnitude of the relative velocity (speed and direction) between node i and j at time t. In the equation, R is added to des_j^i if node j is located behind node i. Also, R is subtracted from des_j^i if node j is located in front of node i.

The distance between node i and j at time t can be calculated as in Equation 3.18 below [147].

$$des_j^i(t) = \sqrt{\left(x_t^i - x_t^j\right)^2 + \left(y_t^i - y_t^j\right)^2} \tag{3.18}$$

The relative velocity between the two nodes i and j can be calculated as in Equation 3.19 below.

$$\overrightarrow{RV_j^i} = \overrightarrow{V_i} + \overrightarrow{V_j} \tag{3.19}$$

The magnitude of the relative velocity is,

$$RV_j^i = \sqrt{\left(RV_j^i(x)\right)^2 + \left(RV_j^i(y)\right)^2} \tag{3.20}$$

where,

$$RV_j^i(x) = \left(v_i \cos\theta - v_j \cos\phi\right) \tag{3.21}$$

$$RV_j^i(y) = \left(v_i \sin\theta - v_j \sin\phi\right) \tag{3.22}$$

where, $RV_j^i(x)$ is the relative velocity of node j in the x-direction for node i, and, $RV_j^i(y)$ is the relative velocity of node j in the y-direction for node i, v_i and v_j are the velocity

of nodes i and j respectively, $\theta$and ϕ are the motion direction of nodes i and j with respect to x-axis, respectively.

Algebraic Rearranging of Equation 3.18, leads to the result;

$$RLT_j^i(t) = \left(R \mp \sqrt{\left(x_t^i - x_t^j\right)^2 + \left(y_t^i - y_t^j\right)^2} / \sqrt{\left(v_i \cos\theta - v_j \cos\phi\right)^2 + \left(v_i \sin\theta - v_j \sin\phi\right)^2} \right) \tag{3.23}$$

3.5 Dynamic Beaconing-Update Mechanism Functionality

This section presents the functionality of Dynamic Beaconing-Update Mechanism (DBUM) from the perspective of participating nodes. Also, it prsents the workflow of the proposed DBUM.

3.5.1 Participating Nodes Procedures

In the DBUM approach, participating nodes dynamically adjust their beacon intervals based on their deviation distance. By using DBUM, a node executes compulasory update technique (CUT) to emit urgent beacon message (UBM). The routing information in a node's NLM can be well managed using the residual link lifetime with its neighbours.

3.5.1.1 Sender nodes

To maintain up-to-date status information, each node in the network should perform CUT functionality. In CUT, each node emits a UBM packet to the broadcast MAC address. Transmitting UBM is specified by three related conditions. First, a node, i.e., a checker node, should estimate CHT by using DFLCH. Second, a checker node finds that it has deviated the tolerance deviation distance. Third, in case it did not deviate the tolerance deviation distance during $10\,s$, due to constant movement characteristics, it should also emit a UBM. As a node transmits a UBM, it directly resets its timer to determine the next check time CHT.

3.5.1.2 Receiver Nodes

Each node maintains and updates its NLM to store recent information of its neighbours within its transmission range. When a node receives a UBM packet, it processes this packet according to the following steps. A receiver node first checks whether or not the sender's identity exists within its NLM. If it does, the associated entries taken from the same node (formerly *ID* values, e.g. *IP* address) are compared. The packet that has a higher *ID* sequence number is recorded as a new entry in the NLM instead of old entry. Otherwise, a new entry is created for a new neighbour. In addition, a receiver node uses the time when the neighbour node sent UBM packet t_s, also it adds the time it receives the UBM packet t_r in NLM.

The NLM contains neighbour's information extracted from the header of the UBM packet. Based on the information extracted from the UBM header, a node calculates three values and adds them to NLM. First, it calculates the RLT with the sender of the UBM. Second, it calculates ELT of the neighbour's entry (timeout) based on the calculated RLT. Third, it calculates the sender acceleration based on the difference between latest and current velocity values over time that are reported in the latest and current received UBM packets. This matrix for i-nodes network looks like the following in Equation 3.24.

$$NLM = \begin{bmatrix} ID_1^n & x_1 & y_1 & v_1 & a_1 & \theta_1 & RLT_1 & ELT_1 & t_s & t_r \\ ID_2^n & x_2 & y_2 & v_2 & a_2 & \theta_2 & RLT_2 & ELT_2 & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ ID_i^n & x_i & y_i & v_i & a_i & \theta_i & RLT_i & ELT_i & t_s & t_r \end{bmatrix} \tag{3.24}$$

where for any neighbour node i, ID_i^n is the identity of the node with the sequence number n, (x_i, y_i) represents i*'s* current position information in x and y *axis*, v_i is the velocity, a_i is the acceleration (a neighbour's acceleration is calculated by using Equation 3.12), θ_i is the motion direction with respect to x-axis, RLT is the residual link lifetime of the neighbour with the node, ELT is the neighbour's entry lifetime, t_s is the transmitting time of the latest received UBM packet from neighbour i, t_r is the receiving time of the latest received UBM packet from neighbour i.

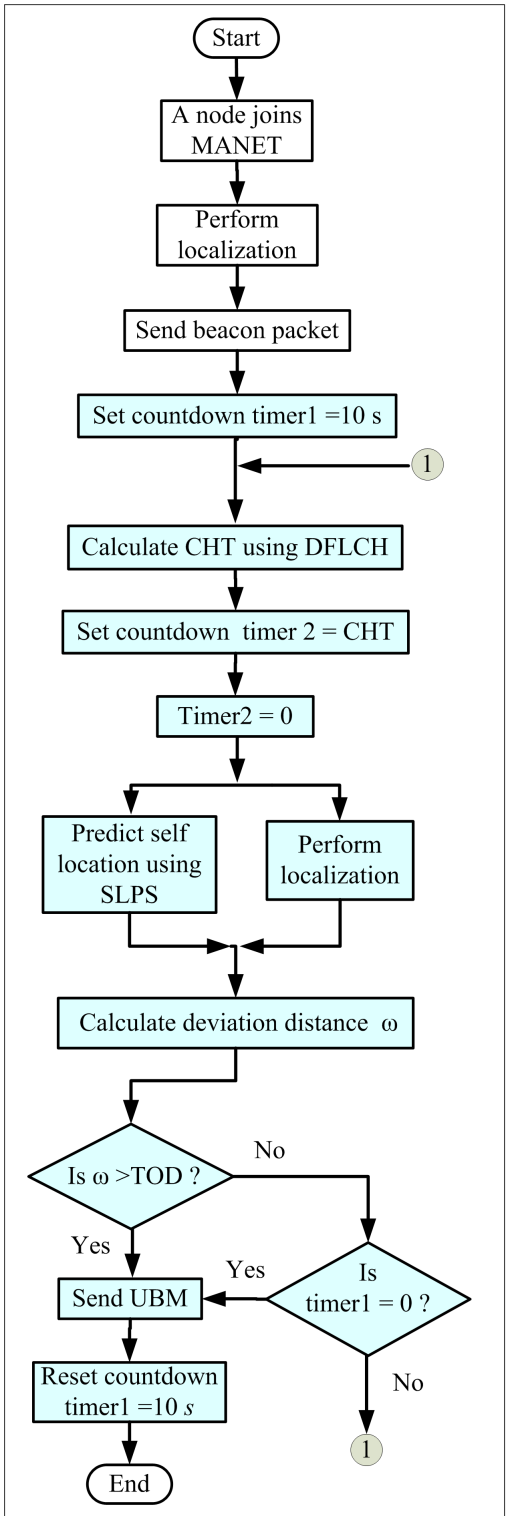
The node's NLM is checked periodically by a node to update it. In this research, the node scans NLM every 0.2 seconds. As mentioned, the node performs the NMEM technique to ensure the refreshment of routing entries in neighbourhood matrix. The proposed method to add\remove neighbours' entries from NLM is ELT period time which is equal to RLT as demonstrated in Section 3.4.1.

3.5.2 Workflow of the Dynamic Beaconing-Update Mechanism

This sub-section presents the proposed flowcharts that visualize a series of the workflow's. There are two separate flowcharts. The first one shows the procedures from the perspective of the sender node. The second one shows the procedures from the perspective of the receiver node.

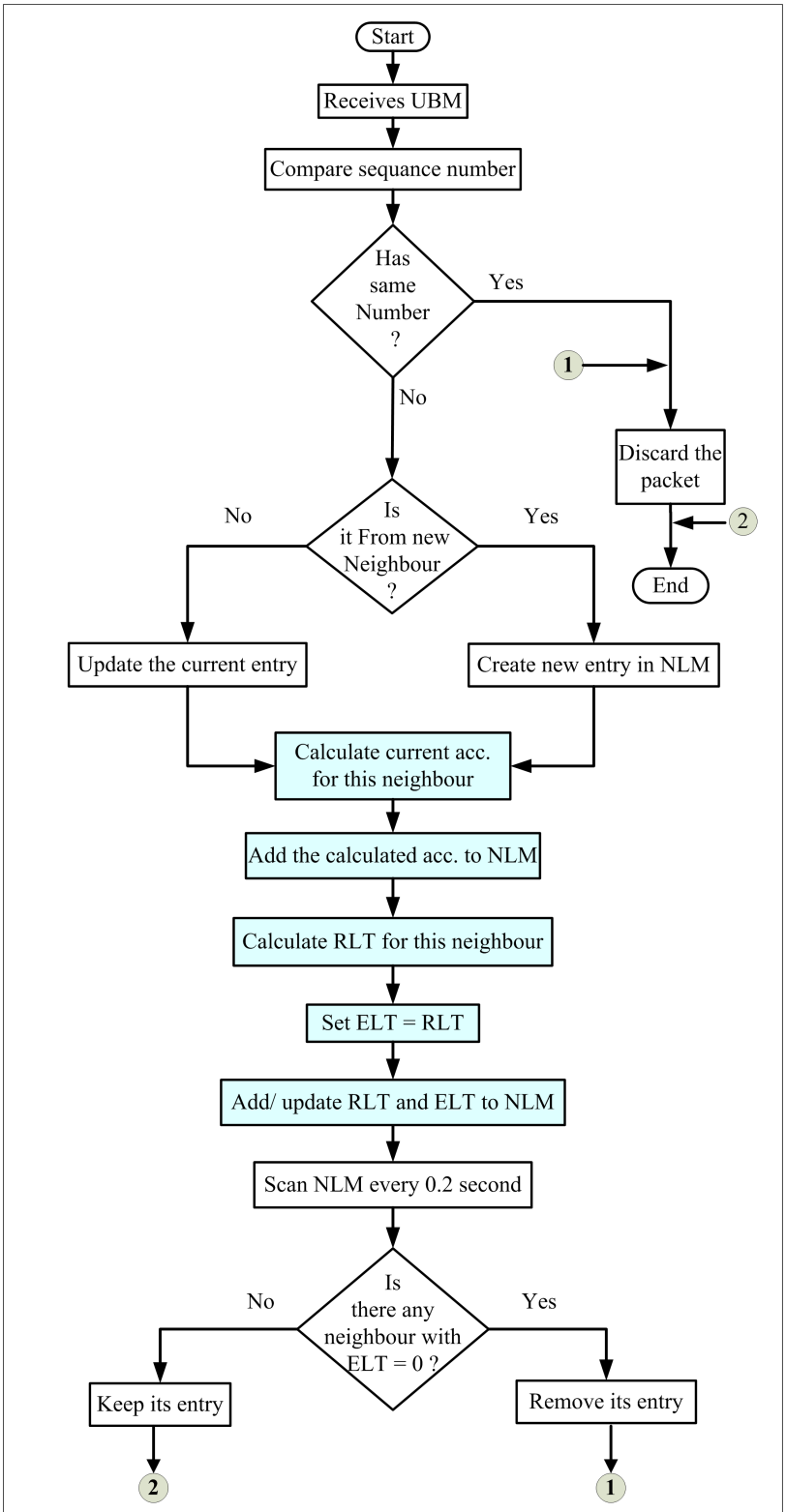
The first flowchart in Figure 3.10 below represents the process to initiate UBM by all nodes. The initial state in this flowchart starts as a node joins the network by performing localization process and sending beacon packet. The second main state is to calculate the next check time based on a node speed by using the proposed fuzzy logic model DFLCH to perform localization process. This state is followed by calculating the expected location of the node (the checker node) by using the dead-reckoning approach. The pre-final state of this flowchart is to find the deviation distance ω and to compare it with the tolerance deviation distance TOD. The final state of this flowchart is to make the decision to send an urgent beacon packet if ω is greater than TOD.

The second flowchart in Figure 3.11 below represents the procedures handled by a node when it receives a beacon packet. The initial state in this flowchart receives the UBM packet and looks at its header information. The second main state is to make the decision to update an existing entry or to create a new one. Based on the information at the header of the UBM, a receiver node calculates the acceleration, the residual link lifetime RLT and the entry lifetime ELT. This state is followed by maintaining or updating an exist entry. The pre-final state of this flowchart is to scan the nodes' neighbourhood's location-matrix NLM to find a stale entry. The final state of this flowchart is to make the decision to keep an entry or to remove it base on its ELT value.



CHT: Check Time, DFLCH: Dynamic fuzzy logic check time, SLPS: Self Location Prediction Scheme, ω :deviation distance, TOD: Tolerance deviation distance, UBM: Urgent Beacon Message

Figure 3.10. Flowchart of DBUM Executed by Sender Node, the Shaded Blocks are the Contribution of the Proposed Protocol



UBM: urgent Beacon Message, NLM: neighbours' Information Matrix, RLT: Residual Link Lifetime, ELT: Entry Life Time, acc: Acceleration

Figure 3.11. Flowchart of DBUM Executed by Receiver Node, the Shaded Blocks are the Contribution of the Proposed Protocol

3.6 Implementation of Dynamic Beaconing-Update Mechanism

This section focuses on the implementation of Dynamic Beaconing-Update Mechanism (DBUM) in a simulated environment. In this research, Ns2 simulation is considered in the implementation to validate the research assumptions and propositions. The implementation of the proposal in Ns2 goes through three main steps. The first step of the implementation is concerned with the build of the simulation environment. The second step is concerned with the validation of the simulation model, which will be presented in Sub-section 3.6.2. The third step is concerned with the validation of the proposed DBUM, which will be presented in Section 3.6.3.

3.6.1 Simulation Environment

This section presents the simulation environment for implementing the proposed DBUM in Ns2. The simulation setups include the traffic pattern generation, mobility model, and simulation input parameters.

3.6.1.1 Traffic Generation

As stated in [149], existing routing protocol implemented Transmission Control Protocol (TCP), Constant Bit Rate (CBR) or Variable Bit Rate (VBR) as network traffic model. As stated in [149], CBR pattern outperforms other patterns in MANET in terms of packet loss and delay. Hence, in this research, the simulated network traffic is based on a script consisting of CBR data connections in the all proposed scenarios .

The traffic generator in Ns2 is called "Connection Pattern File Generator". The traffic generator is located in the directory *"cmu-scen-gen/cbrgen.tcl"*. The file defines the packet type, packet size and the number of traffic flows. Figure 3.12 below shoes an example of the UDP agent used in the program. In the example, node 0 connected to node 11, the CBR traffic generator is started at time 0.42613981311496 and stopped at time 900.

```
#
# 0 connecting to 11 at time 0.42613981311496
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp_(0)
set recv_(0) [new Agent/recvAgent]
$ns_ attach-agent $node_(11) $recv_(0)
$recv_(0) dst-ip [$node_(11) address?]
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 256
$cbr_(0) set interval_ 0.5
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $recv_(0)
$ns_ at 0.42613981311496 "$cbr_(0) start"
$ns_ at 900 "$cbr_(0) stop"
#
```

Figure 3.12. An Example Shows the cbr Traffic Generation Between Two Nodes

3.6.1.2 Movement Model

As mentioned earlier in Chapter 2, Section 2.8.5 this research focuses on Random Waypoint mobility model (RWP) [143]. To generate the movement file the mobility scenario file generator is used. The file defines the simulation area and the mobility model of a set of mobile nodes over the simulation time period. The generator is located in the *Setdest* sub-directory of Ns2 independent utility directory and it is under "*$ns2_home/indep-utils/cmu-scen-gen/setdest*". The following command is utilized to generate the RWP model.

"*./setdest*[*-n num_of_nodes*][*-p pausetime*][*-s maxspeed*][*-t simtime*][*-x maxx*][*-y maxy*] > [*scenario_output_file*]"

3.6.1.3 Simulation Input Parameters

As mentioned in [150], the suitability and the performance of any protocol in MANET are greatly influenced by its operational parameters. Also, as stated by Perkins et al. in [151], the behaviour of participating nodes in MANET is complex to be modelled, especially in a high mobility environment; thus the studies on this kind of environment would be a challenge. Moreover, selected parameters in a high mobility envi-

ronment would be more challenging and should be planned carefully and thoroughly understood in order to produce meaningful results [152, 153]. In addition, up to this date, there is no standard lower or upper value for these simulation parameters in the MANET community [154]. Also, there is an uncertainty about the significant factors involved in a MANET's routing protocol, and there is a lack of tools available to enable a researcher to conduct credible simulation-based studies [151].

In this research, the GPSR simulation model is used as a simulation environment to study the feasibility of the proposed algorithms. Hence, unless necessary otherwise, all simulation experiments in this research use the simulation parameters identical to a subset of those used by Karp [44]. Those parameters are recorded in Table 3.4. Additionaly, because this research uses the Position-based Opportunistic Routing (POR) protocol [45] in the evalution stage, the simulation parameter sused by Yang in [45] are also presented in Table 3.5. Also, Table 3.6 shows the additional proposed simulation parameters with justification.

Table 3.4. The Ns2 Simulation Model of GPSR

Description	**Value**	**Unit**
Simulation time	900	*s*
Simulation area	1500 × 300 , 2250 × 450 and 3000 × 600	m^2
Mobility model	RWP	-
No. of nodes	50,112 and 200	*nodes*
Node queue size	50	*pkt*
Interface Queue Type	FIFO	-
No. of data packets	10000	*pkt*
Data packet generating ratio	2	*Kbps*
DPsize	64	*byte*
No. of sources	30	*flows*
TTL	30 or 10	*s* or *hops*
Node transmission range	250	*m*
BPIT	3	*s*
ELT	3 * BPIT	*s*
BPsize	13	*byte*
MAC protocol	IEEE 802.11 DCF	-
Propagation Model	TwoRayGround	-

s: Second, RWP: Random WayPoint model, No.: Number, pkt: Packet, FIFO: : First In First Out, *Kbps:* Kilobit Per Second, DPsize: Data packet size, *m*: meter TTL: Time To Live, BPIT: Beacon Packet Interval Time, ELT: Neighbour's Entry LifeTime, BPsize: Beacon Packet size, DCF: Distributed Coordination Function

Table 3.5. The Ns2 Simulation Model of POR

Description	**Value**	**Unit**
Simulation area	1200 × 600	m^2
No. of nodes	80	*nodes*
Data packet generating ratio	4	*Kbps*
DPsize	256	*byte*
No. of sources	10	*flows*
The other parameters are set as the same as in Table 3.4		

Table 3.6. Proposed Simulated Topology Characteristics

Description	Value	Unit	Justification
Pause time	0	*s*	To solve the problems accompany with RWP, hot-spot problem and steady-state period problem as stated in [142, 145], this research adopted the solution presented in [143] that suggested to use zero pause time
Node speed	[5 , 40]	*m\s*	These values are selected to give a node diversity mobility range to correctly evaluate the proposed algorithms under high and low mobility as suggested by [41, 45, 57, 150, 151]
No. of sources	5, 10, 15, 20 and 25	*flows*	These values are selected to evaluate the proposed algorithms under various traffic loads as suggested by [59, 60, 61, 63, 150, 151]
DPsize	256	*byte*	This value is chosen to evaluate the proposed algorithms under heavy traffic loads as used in [45]
Simulation area	600 × 300 800 × 450 and 2000 × 450	m^2	These values are chosen to give node density of 1 node / 9000 m^2 as suggested by [44, 45, 100, 104, 105]. This value is a reasonable value that provides a connected network
Steady-state	0	*s*	The simulation results are collected directly as the simulation started to function as suggested by [143], the argument to use zero steady-state value is to not losing any results that may change the performance judgment of a protocol
The other parameters are set as the same as in Table 3.4			

3.6.2 Implementation and Verification of GPSR and POR

Testing a simulation model with different simulation environment is a very important technique used for the verification of a simulation model [122, 123]. The greedy perimeter stateless routing (GPSR) simulation model [44] is used as a simulation environment to implement the proposed GSAR-DBUM in order to study its feasibility. Thus, it is very important to ensure the correctness of GPSR model in the simulation's environment before adding the proposed alteration to the original simulation code of GPSR. The GPSR simulation model was built based on the Ns2 model components. In order to ensure that the GPSR simulation model and its results are sufficiently correct, validation processes is in need to be performed. To achieve this, the simulator is run by using the same parameters as used in the original protocol GPSR as mentioned in Table 3.4 above. The purpose of using the same setting is to check to see that the simulation output is realistic compared to simulation results reported in [44]. Appendix *A*.1 shows that the collected simulation results are comparable to those published in [44].

This research uses Position-based Opportunistic Routing (POR) protocol [45] to evaluate the performance of the proposed protocol. Therefore, the researcher performs the same procedures executed with GPSR to verify POR. The researcher builds the simulation scenarios for POR based on the same parameters as used in the original protocol POR as mentioned in Table 3.5 above.

3.6.3 Dynamic Beaconing-Update Mechanism Validation

The purpose of this section is to ensure the correctness of the proposed mechanism based on its intended design objectives. The following two sub-sections present the procedure to validate the DBUM in two different methods. The first method is adopted from [44] in which GPSR-DBUM is validated in static environment. The second method is adopted from [122, 155]: in this method the validation process is executed according to Sergent's recommendation as stated in his paper "...Comparison to Other Models: Various results (e.g., outputs) of the simulation model being validated are compared to results of other (valid) models. For example, (1) ***simple cases of a simulation model are compared to known results of analytic models***, and (2) the

simulation model is compared to other simulation models that have been validated".

3.6.3.1 DBUM Validation in Static Environment

In this section, as suggested by Karp in [100], the GPSR-DBUM implementation is validated by running it on ten experiments of static scenario. The purpose of these experiments is to achieve 100% packet delivery success to demonstrate that the GPSR-DBUM algorithm makes correct forwarding decisions. The experiments are performed over an ideal MAC layer (the Null MAC [156]), a 2 Mbps and contention-free network. After achieving the 100% delivery on the Null MAC, the GPSR-DBUM implementation is again validated on the same static scenario atop the Ns2 802.11 MAC DCF layer (traditional version). This is done to verify GPSR-DBUM's response to MAC transmit failure callbacks.

In both scenarios the network area is set to $2000\,m \times 450\,m$ and 25, 50, 75, 100, 150, 175 and 200 participants, meanwhile other parameters of the network are the same as described in Table 3.6 above except the nodes speeds is set to zero. Figure 3.13 below shows the simulation results of the two scenarios. The figure shows that after running the GPSR-DBUM several times over the Null MAC, a 100% delivery is acheived. This means that the GPSR-DBUM code can make correct forwarding decisions. The same figure shows that because of running the GPSR-DBUM atop the Ns2 802.11 MAC DCF layer, when the number of nodes is 200 nodes (the highest number of nodes in this experiment), the packet delivery rate decreases to 89.3% as the number of participants increases. The decrement in the delivery rate is attributed to the contention at the interfaces of the nodes to access the radio channel. Hence, based on the simulation results, it can be concluded that the GPSR-DBUM can make the correct forwarding decisions.

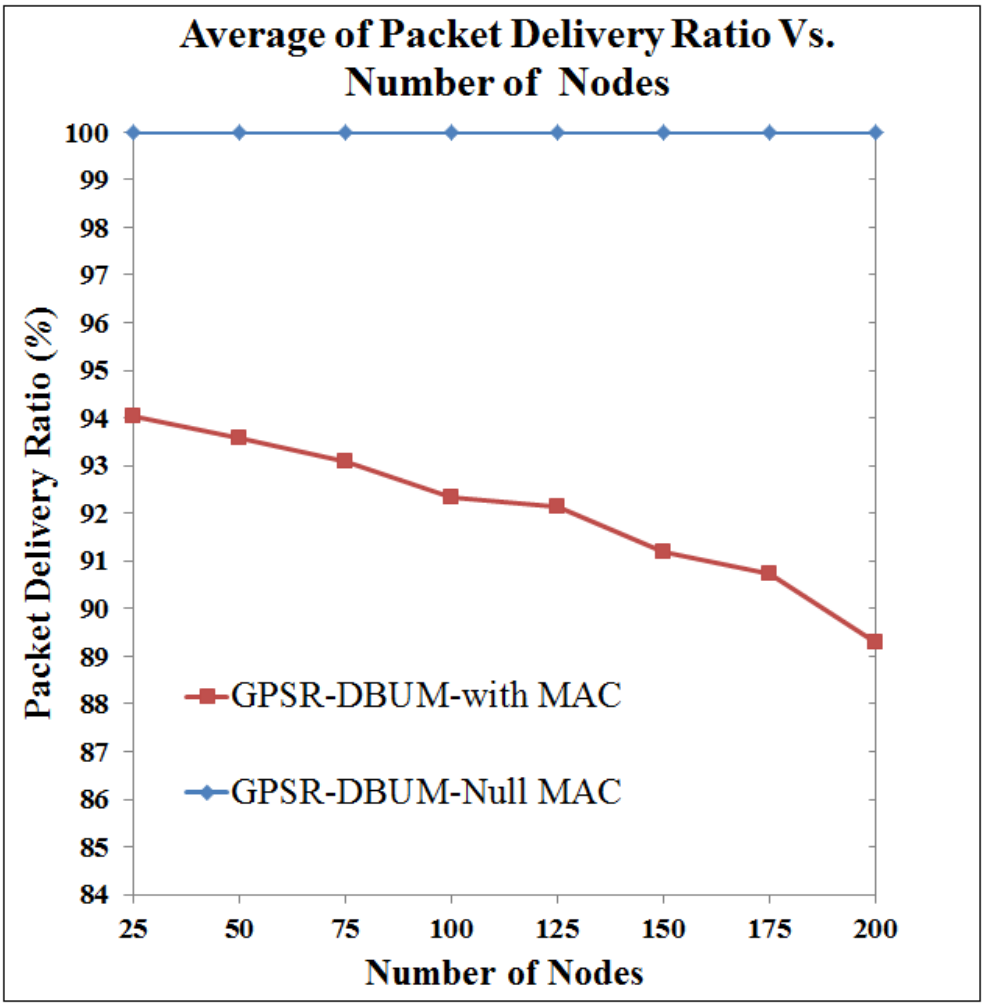


Figure 3.13. Average Packet Delivery Ratio via Number of Nodes

3.6.3.2 DBUM Validation in Mobile Environment (Simple Case)

To validate the correctness of the proposed DBUM, this section presents a mathematical analysis and an experimental results comparison. The researcher monitors the ability of DBUM algorithm to track the freshness of the information in nodes' NLMs of a simple network topology created in Ns2 as suggested by [122, 155]. In the proposed scenario, the mathematical result of calculating the neighbours' entries lifetime ELTs, based on Equation 3.23, are compared with the simulation output result.

The proposed scenario is shown in Figure 3.14 in which 20 nodes are configured in a rectangular simulation area of $600\,m \times 300\,m$. The argument to use this simulation dimensions is attributed to the number of nodes needed to maintain the ratio of 1 node / 9000 m^2 as suggested by [44, 45]. The other parameters of the network are same as described in Table 3.6 except that the simulation time is set to 50 s because the steady

state is disabled in this research. On other words, the researcher started to collect the results as the simulation started.

In the proposed scenario, source node S has the neighbours A, B, C, E, F, G, H, I, J, K and L. At the beginning of the simulation the nodes are inserted with the following specification of (ID_i, x, y, v, θ); source node $S = (1, 0, 0, 10, 0°)$, node A with $(2, 190, 80, 8, 120°)$, node B with $(3, 170, -60, 12, 90°)$, node C with $(4, -200, 15, 9, 135°)$, , node E with $(5, 60, 160, 11, 270°)$, node F with $(6, -40, -100, 14, 315°)$, node G with $(7, -170, -60, 25, 225°)$, node H with $(8, 150, -120, 17, 135°)$, node I with $(9, 50, -50, 15, 300°)$, node J with $(10, -120, 70, 22, 45°)$, node K with $(11, 90, 90, 30, 44°)$, and finally node L with $(12, -115, -155, 36.5, 90°)$.

where, (ID_i): is the node identity, (x, y): the geographical position of a node, (v): is the node's velocity, (θ) is the node's motion direction with respect to x-axis.

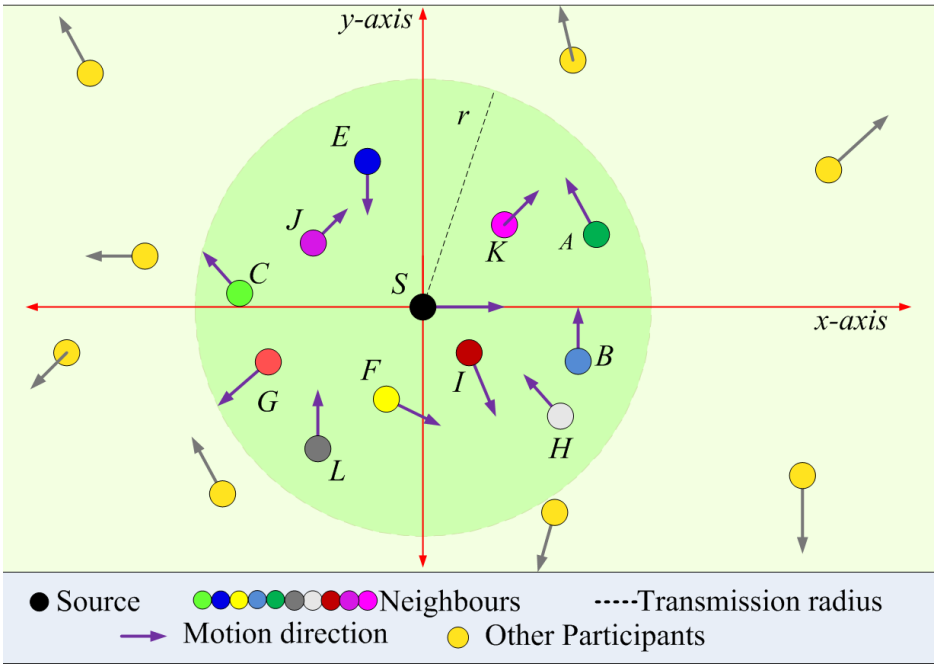


Figure 3.14. Network Model for Performance Validation of DBUM

Based on these specifications, the Equation 3.23 in Section 3.4.1 is applied to estimate ELT, i.e. RLT of nodes A, B, C, E, F, G, H, I, J, K and L in node S's NLM as shown in Table 3.7 below.

Table 3.7. RLT and ELT Values of Neighbours Nodes in Node S's NLM

Node	$des_i^s\ m$	$RV_i^s x$	$RV_i^s y$	$RV_i^s m\backslash s$	$RLT_i^s = \frac{R \mp des_i^s}{RV_i^S} s$	$ELT_i^s s$
A	206.15	6.00	6.9	9.16	4.79	4.79
B	180.27	10.00	12	15.62	4.46	4.46
C	200.56	3.63	6.36	7.32	6.75	6.75
E	170.88	9.99	-11	14.86	5.32	5.32
F	107.70	19.89	-9.89	22.22	6.4	6.4
G	180.28	-7.67	-17.67	19.27	3.61	3.61
H	192.09	-2.02	12.02	12.18	4.75	4.75
I	70.71	17.50	-12.99	21.79	8.22	8.22
J	138.92	25.55	15.55	29.18	3.80	3.80
K	127.27	31.21	21.21	37.84	3.25	3.25
L	193.00	10.00	36.5	37.84	1.56	1.56

Table 3.7 above shows that, although the neighbours *B* and *G* have the same distance from the source node *S*, each of them has different ELT due to the different in the relative speed values of them. Also, it can be concluded from the table that, although the neighbour *K* has the highest relative speed, the neighbour *L*'s entry will be removed faster than the neighbour *K*. This is because, with respect to node *S*, the neighbour *L* is located farther from the neighbour *K*. As aconcequanc, the neighbour *L* will depart the transmission range of nodes *S* faster than the neighbour K.

To monitor the add\delete neighbours' entries from the node *S*'s NLM, the researcher performed another test by running the simulation based on the network setting mentioned at the beginning of this section. In the simulation, the state of the simulated nodes, i.e., the execution of the deletion of a neighbour's entry events is displayed just after each deletion event occurs (in node *S*). Because the state maintained by node *S* changes constantly over time due to nodes mobility, determining the exact knowledge of its NLM's entries changes occur during the simulation run is very difficult. Thus, the simulation results in this experiment are discussed for the first 8.5 *sec*. The determination of only 8.5 *sec* is based on the mathematical calculation in Table 3.7 above.

Figure 3.15 below shows a comparison between the calculated and experimental number of neighbours' entries in node *S*'s NLM. The figure shows that the GPSR-DBUM node *S* stores state of different number of neighbours as the simulation time increases. At time 1 *sec* from the life-time of the simulation, it is found that node *S* stores the state

for 11 neighbours. At time 1.6 *sec* from the life-time of the simulation, it is found that node *S* stores the state for 10 neighbours because it already removed node *L*'s entry. At time 3.2, it is found that node *S* stores the state for 9 neighbours because it already removed node *K*'s entry. Also, it is found at time 3.62, 3.8, 4.46, 4.8, 4.85, 5.32, 6.5, 6.75 and 8.3 *sec*, from the life-time of the simulation, that node *S* stores the state for 8, 7, 6, 5, 4, 3, 2, 1 and 0 neighbours respectively.

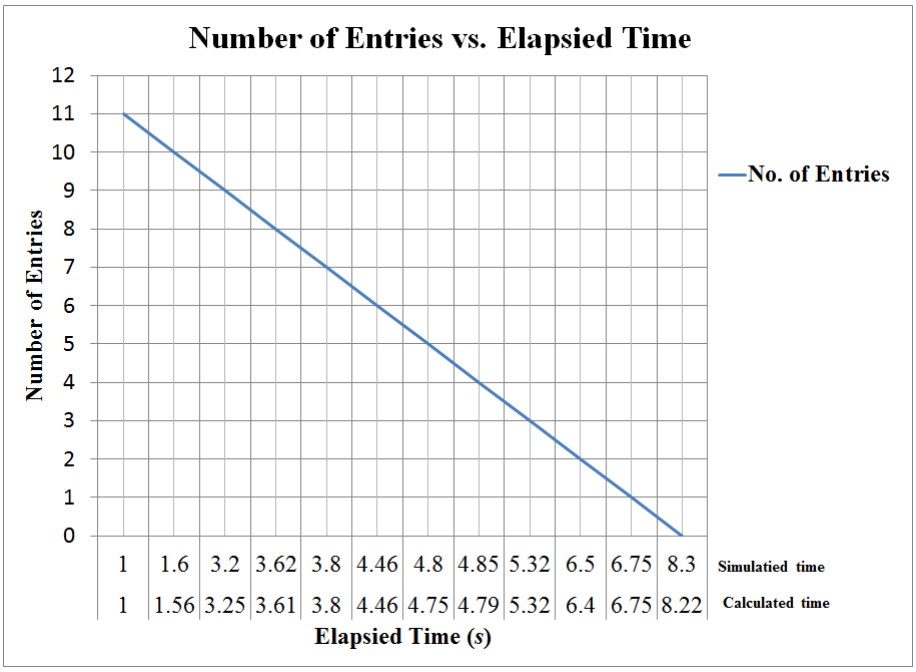


Figure 3.15. Comparison of Calculated and Simulated Number of Neighbours' Entries in Node S's NLM Via Elapsed Time

Based on both mathematical, i.e., calculated relative speed and simulation results, i.e., neighbours entries removal time, Figure 3.16 below shows that neighbours with same distance from a node (i.e., the neighbours *B* and *G*), the neighbour with highest relative speed is removed faster, i.e., the neighbours *G*. This is because, as the relative speed increases a neighbour's entry in a node's NLM decreases. In the case of different distances from the source node and same relative speed (i.e., the neighbours *K* and *L*), the neighbour with lowest distance is removed faster, i.e., the neighbours *L*. In the case of different distances from the source node and different relative speed, the neighbours' entries lifetimes are different as well.

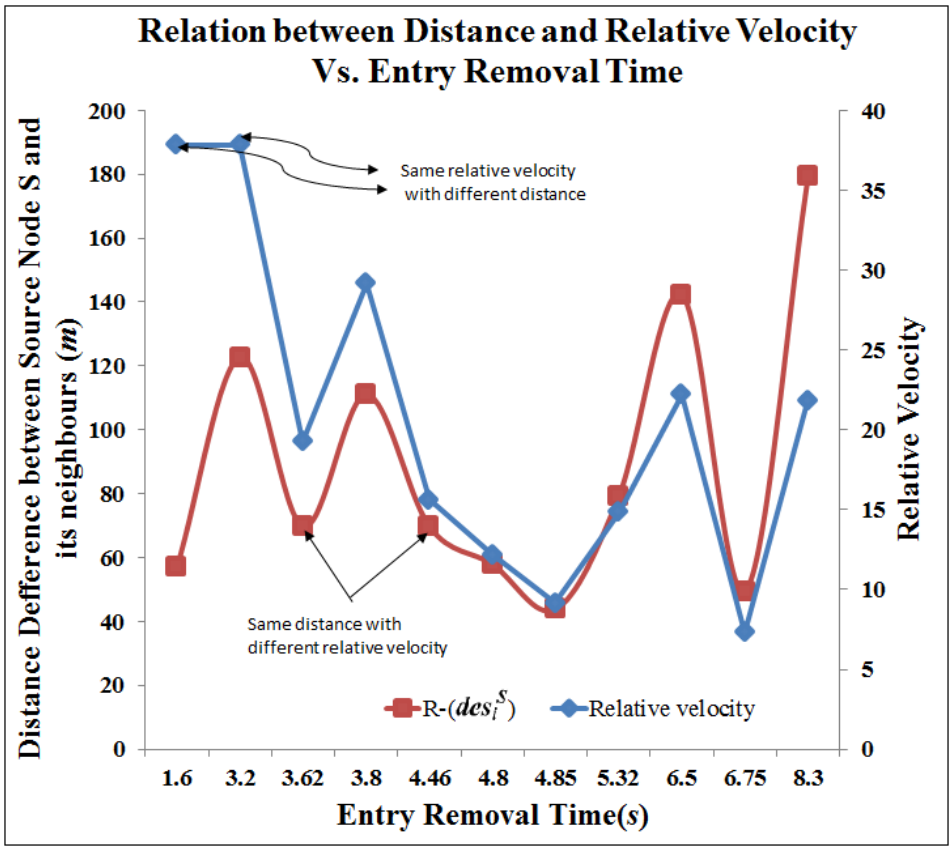


Figure 3.16. Neighbours Entries Lifetime in Node S's NLM Via Relative Velocity

The simulation results in Figures 3.15 and 3.16 are in very good agreement with the analytical view in Table 3.7. Hence, because the mathematical calculation and the simulation have comparatively the same results, the model of DBUM is valid because it can achieve its design objective.

3.7 Implementation of DBUM to Assign the Tolerance Deviation Distance and the Experience Fuzzy Delay in CUT

Next two sub-section presnt how does this research determine the optimal tolerance deviation distance (TOD) and the fuzzy-delay experinced by a node performing GPSR-DBUM routing protocol. The TOD measurement is critical issue to the performance of the compulsory beaconing-update CUT. The fuzzy-delay is incurred by executing fuzzy logic in the Dynamic Fuzzy Logic Controller Check-time (DFLCH) model that is might be a critical issue to the performance of the GPSR- DBUM routing protocol.

3.7.1 Optimal Tolerance Deviation Distance Calculation

The purpose of this section is to determine the optimal tolerance deviation distance TOD to be used in the proposed compulsory beaconing-update CUT. To determine the optimal TOD, the GPSR-DBUM is implemented as the routing protocol. To facilitate communication between the nodes within their transmission range, the simulations scenario is set with no obstacles. Additionally, in this simulation experiments, the source-destination pairs are chosen randomly amongst the participating nodes.

The researcher conducts three simulation experiments. In the proposed scenarios, 100 nodes are configured in the simulation area of $2000\,m \times 450\,m$. The argument to use this simulation dimensions is attributed to the number of nodes needed to maintain the ratio of 1 node / 9000 m^2 as suggested by [44, 45]. The other parameters of the network are the same as described in Table 3.6 except the nodes' speed is set to a 5, 10, 20, 30, and 40 $m\backslash s$ for the first simulation experiment and a 20 $m\backslash s$ for the second and third simulation experiments. Also, the researcher varies the deviation distance to 3, 6, 9, 12, 15, 18, and 21 *meters*.

The first simulation experiment that is stated in Sub-section 3.7.1.1 investigates the relation between the nodes' speed, time and the deviation distance ω. In the second and third simulation experiments that are stated in Sub-sections 3.7.1.2 and 3.7.1.3 respectively, the researcher uses two performance metrics to determine the optimal TOD. The first performance metrics is the Control OverHead (COH), and the second performance metrics is the Inconsistency of Neighbourhood Matrix (INM). Both selected performance metrics can show the effectiveness of the proposed GPSR-DBUM, because the purpose behind DBUM is to increase the accuracy of nodes' information in the network with very low overhead. Further, both selected performance metrics have a huge influence on other performance metrics like End-to-end delay (EED), and Packet delivery ratio (PDR) [116].

As stated in [116, 157, 158], COH represents the average control packet sent by participant during the simulation time period. The lower value of COH means the better performance of a protocol. The INM metric represents the difference between the actual a node's neighbours' number to the actual neighbours' number listed in a node's

neighbours-list. The lower value of INM means the better performance of the protocol.

3.7.1.1 The Relation between Deviation Distance and Nodes' Speed

This sub-section investigates the relation between the nodes' speed, time and the deviation distance ω. The researcher conduct this investigation to show the effect of nodes' speed on the time required to reach each proposed deviation distance.

Figure 3.17 below shows the time required to reach a deviation distance in GPSR-DBUM protocol as a function of nodes' speed. The simulation results revealed that as the nodes' speed increases the required time to reach the assigned deviation distance decreases. This is because when a node moves faster, the error in its position information compared to the position information in the latest sent beacon update will increase. The error increment in a node position information decreases the time to reach a specific deviation distance.

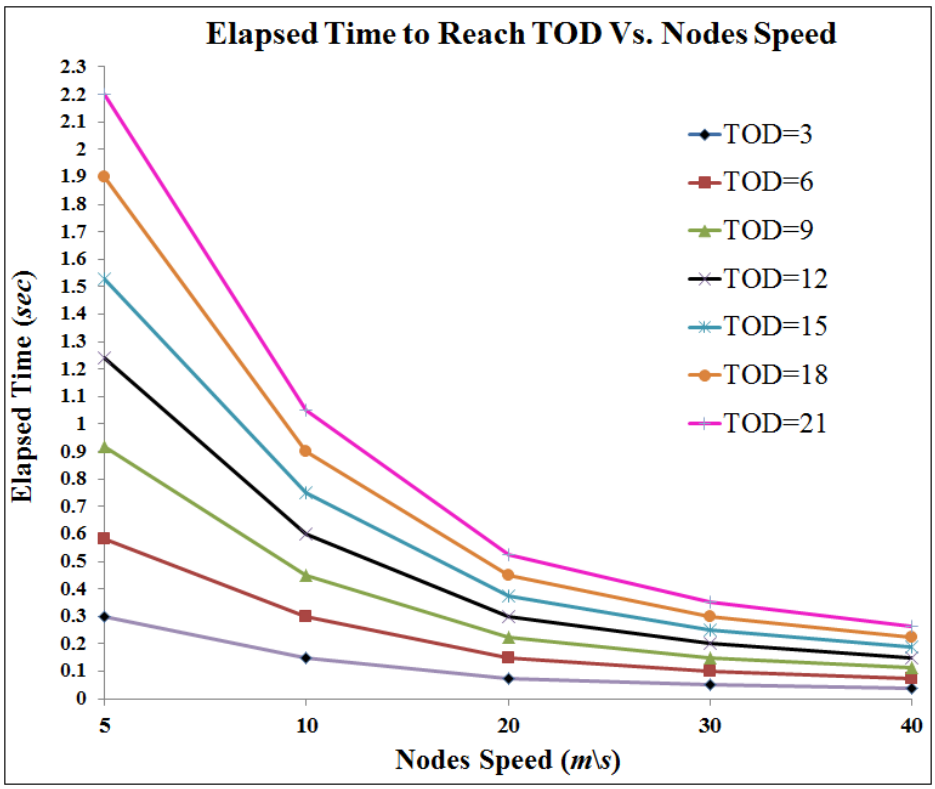


Figure 3.17. Required Time A node Need to Reach TOD Via Node Speed

3.7.1.2 The Relation between Deviation Distance and Inconsistency of Neighbourhood Matrix

Figure 3.18 shows the inconsistency of neighbourhood matrix ratio in GPSR-DBUM protocol as a function of tolerance deviation distance TOD. As mentioned earlier in the design specification in Sub-section 3.3.7, the final decision to send the urgent beacon message UBM is related to the TOD value.

Figure 3.18 below shows that when the TOD value increases, the inconsistency of neighbourhood matrix INM increases as well. The reason is when the TOD value increases the UBM sending frequency decreases, i.e., the interval time between two consecutive UBM's is long. Long sending frequency does not show the exact mobility's topology changes. This results in increasing the outdated entries' ratio in nodes' NLMs. The values of inconsistency of neighbourhood matrix INM for GPSR-DBUM protocol in all scenarios as pairs of (TOD, INM percentage) are $(3, 2\%)$, $(6, 6\%)$, $(9, 9\%)$, $(12, 11\%)$, $(15, 17\%)$, $(18, 29\%)$ and $(21, 42\%)$.

As Figure 3.18 below demonstrates, the inconsistency of the nodes entries slightly increases and relaxes as TOD gets close to 12 *meters* and dramatically increases as TOD exceeds 12 *meters*. This experiment concludes that the GPSR-DBUM protocol presents reasonable accurate entries in nodes' NLM at TOD close to 12 *meters.*

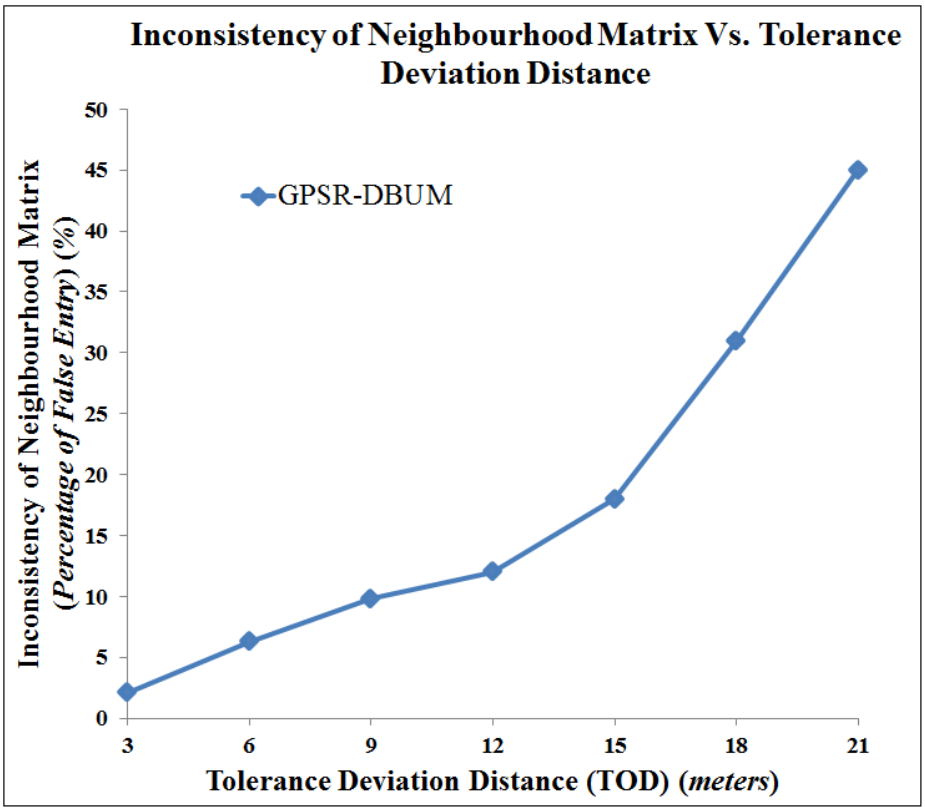


Figure 3.18. Inconsistency of Neighbourhood Matrix Via Tolerance Deviation Distance

3.7.1.3 The Relation between Deviation Distance and Control Overhead

Figure 3.19 below depicts the average control overhead exchanged by GPSR-DBUM as a function of tolerance deviation distance TOD. Figure 3.19 below indicates that when the tolerance deviation distance TOD increases, the average control overhead exchanged by GPSR-DBUM decreases. The reason behind the decrement in the control overhead is atributed to the increment in the time that a node needs to exceed the assigned deviation distance, i.e., TOD, as shown in Figure 3.17. This results in decreasing the UBM sending frequency. On one hand, a low sending frequency leads to a lower overhead as shown in Figure 3.19 and higher obsolete location information in a node's neighbourhood location information NLM as shown in Figure Figure 3.18; on the other hand, a high sending frequency may afford more accurate information as shown in Figure 3.18, but incurs heavier overhead as shown in Figure 3.19.

Figure 3.19 below shows that the values of emitted control overhead for GPSR-DBUM protocol in all scenarios as pair of (TOD, COH) are (3, 26500), (6, 25500), (9, 24000),

(12, 19000), (15, 10000), (18, 5000) and (21, 2500). Also the same figure shows that UBM sending frequency slightly decreases and relaxes as TOD is close to 12 *meters* and dramatically decreases as TOD exceeds 12 *meters*. This experiment concludes that the GPSR-DBUM protocol presents reasonable control overhead at TOD close to 12 *meters.*

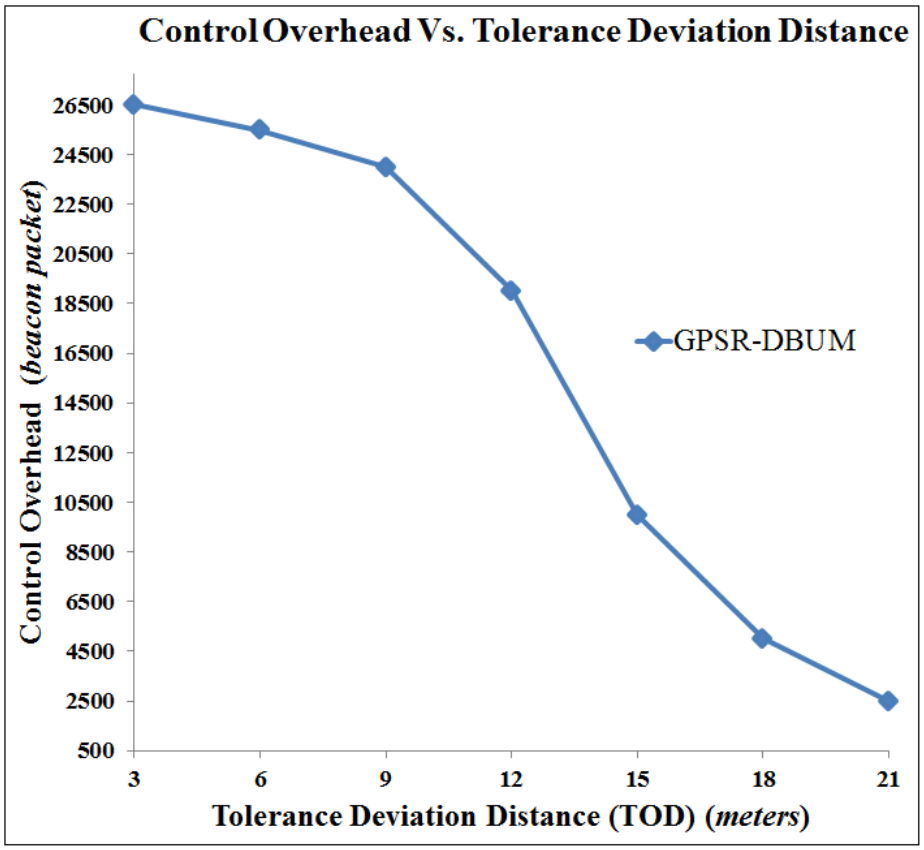


Figure 3.19. Control Overhead Via Tolerance Deviation Distance

3.7.2 Fuzzy Processing Delay

This research uses fuzzy logic to propose DFLCH model that is used to find the check time CHT to perform the localization process. To figure out if the fuzzy-delay is harmful to the performance of the routing protocol, the researcher conduct a comparative study. In this comparison, the processing delay experienced to execute the beaconing-update process is measured and compared for both DFLCH in the dynamic update mechanism DBUM, i.e., GPSR-DBUM and the conventional static beaconing-update in GPSR. The researcher adopts the method suggested in [159, 160] to measure the processing delay of both GPSR-DBUM and GPSR protocols.

The researcher coducts the simulation experiments through varying the node speed. There are two reasons behind selecting this scenario. First, in the proposed GPSR-DBUM, executing the DFLCH algorithm to figure out the CHT values increases as a function of speed. Second, in the simulation results presented in [52, 53], researchers showed that the end-to-end delay increases as a function of speed. In the proposed scenarios, a 100 nodes are configured in the simulation area of $2000\,m \times 450\,m$. The other simulation parameters are built based on those parameters mentioned in Table 3.6.

As every mobile node in the simulation area has the same settings and therefore the experience internal processing delay to execut beaconing algorithm by these nodes will be the same. To make sure of this assumption, 5 % of the participants are selected randomly in each experiment to measure their average total experienced beaconing delay over the simulation time. The fuzzy-delay and traditional beaconing delay, are incremented at each time a node performs the process.

To measure the fuzzy-delay, a node performs the following actions: a node starts a counter when it is about to perform a fuzzy algorithm to determine a new CHT. After finishing executing the fuzzy algorithm, it stops the counter and records the counter result in a timestamp, say FuzzyMonitor. This FuzzyMonitor contains an incremented per-process fuzzy-delay counters. To estimate the total fuzzy-delay, every time a node needs to perform DFLCH algorith it will follow the same procedures executed in the first step and adds the estimated time to the FuzzyMonitor.

Figure 3.20 shows the delay in both GPSR and GPSR-DBUM as a function of node speed. The figure indicates that the proposed GPSR-DBUM method has a slightly higher average delay compared with the traditional beaconing process. The proposed GPSR-DBUM method needs more processing delay to calculate the CHT using DFLCH model compared to traditional method used in GPSR. The traditional beaconing method that is used in GPSR, on the other hand, is executed at a fixed time. This is why the beaconing method achieved a lower delay than the GPSR-DBUM method.

The results revealed that as the speed of nodes increases the execution of the DFLCH algorithm in the GPSR-DBUM increases too. This is because a node exceeds the

Tolerance Deviation Distance (TOD) faster as its speed increases as discussed in Subsection 3.17. This requires more effort from a node to perform fuzzy algorithm to find the next Check Time (CHT) to execute beaconing-update process that increases the fuzzy processing delay. The simulation results show that the increment in the fuzzy-delay for GPSR-DBUM protocol compared to traditional beaconing in all scenarios as pair of (speed, increment in fuzzy-delay) are: $(5\,m\backslash s, 0.4\,\mu s)$, $(10\,m\backslash s, 0.5\,\mu s)$, $(15\,m\backslash s, 0.6\,\mu s)$, $(20\,m\backslash s, 0.8\,\mu s)$, $(25\,m\backslash s, 1.0\,\mu s)$, $(30\,m\backslash s, 1.4\,\mu s)$, $(35\,m\backslash s, 1.9\,\mu s)$ and $(40\,m\backslash s, 2.6\,\mu s)$.

This experiment concludes that the increment of the fuzzy-delay in the GPSR-DBUM method is too small (very few microsecond) compared to the delay experienced in traditional method in GPSR. To conclude, the fuzzy-delay in a MANET is negligible for the DFLCH functionality and has no practical relevance in the beaconing-update design.

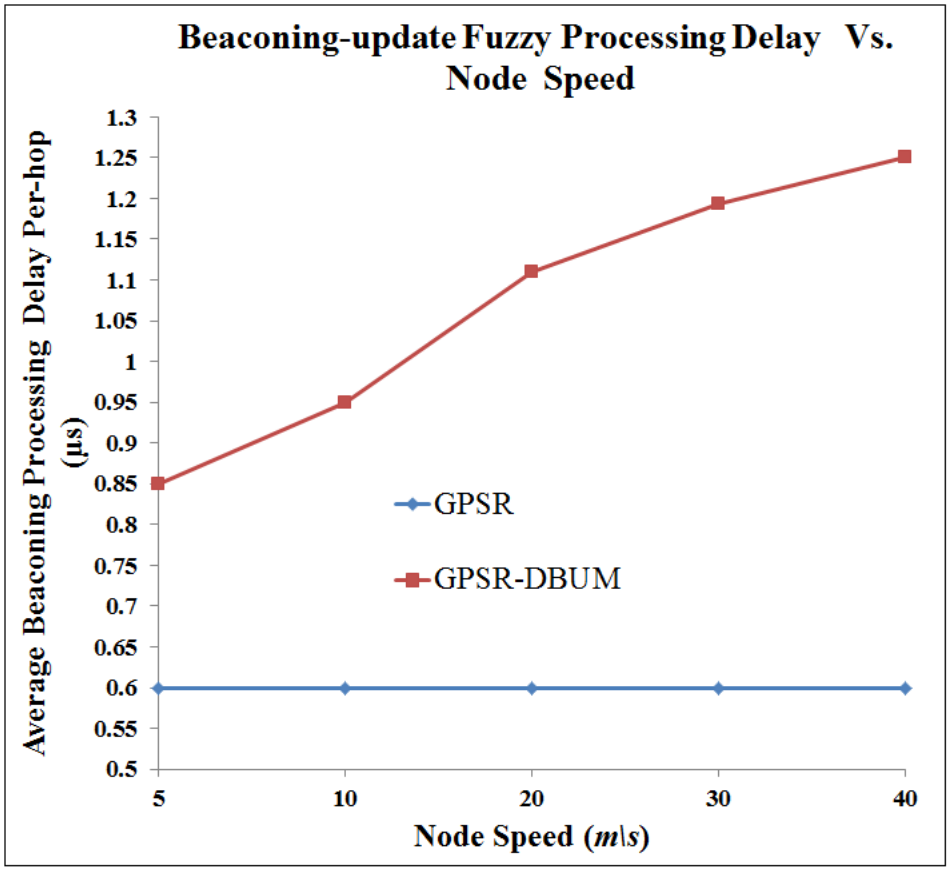


Figure 3.20. The Fuzzy Processing Delay Compared to Traditional Beaconing Processing Delay

3.8 Summary of Chapter

This chapter presents the model of the first proposed mechanism. The design of the beaconing-update algorithm termed as DBUM is presented in this chapter. The goal of the design is to develop a DBUM that is primarily responsible for providing a node with fresh information about its neighbours. In order to achieve the above goal, DBUM encompasses two main parts:

1. The design of the Compulsory update CUT; and
2. The design of the NMEM.

The functionality from the perspective of both sender and receiver nodes is explained in detail. To ensure the smooth transition from modeling to implementation, the flowchart stages are shown. Further, validation of the proposed mechanism is extensively presented. The proposed mechanism is validated inside Ns2 by observing the correctness of nodes' NLM during simulation time. The validation is accomplished by examining the proposed mechanism in a proposed scenario. The results of the simulation proved the correctness and efficiency of the proposed mechanism. The next chapter establishes the analytical model, the design and the implementation of the second proposed mechanism.

CHAPTER FOUR

DESIGN AND IMPLEMENTATION OF DYNAMIC AND REACTIVE RELIABILITY ESTIMATION WITH SELECTIVE METRICS MECHANISM

4.1 Overview

This chapter elaborates the design and implementation of the second objective. The proposed solution is termed the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM). The rest of this chapter is arranged in the following order: Section 4.2 presents the DRESM objectives and design details. Section 4.3 presents the Information Distribution and Outgoing Traffic Control Management (IDOTM) technique, followed by the Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE) technique in Section 4.4. Section 4.5 presents DRESM functionality with an illustrative example followed by the deduction of the workflow. Section 4.6 presents the DRESM implementation in a simulated environment. Section 4.7 presents the experience fuzzy processing delay incurred by DRESM. Finally Section 4.8 summarizes the chapter.

4.2 Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism Detailed Design and Objectives

The overall purpose of the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM) is to alleviate the hot-spot problem caused by using the Greedy forwarding Strategy (GFS) in MANET. To achieve this, the traffic load should be evenly distributed in the network by using a multi-routing scheme. The proposed solution DRESM comprises two coherent techniques namely Status Information Distribution and Outgoing Traffic Control Management (IDOTM) and Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE) technique. The details of DRESM model structure are depicted in Figure 4.1 below.

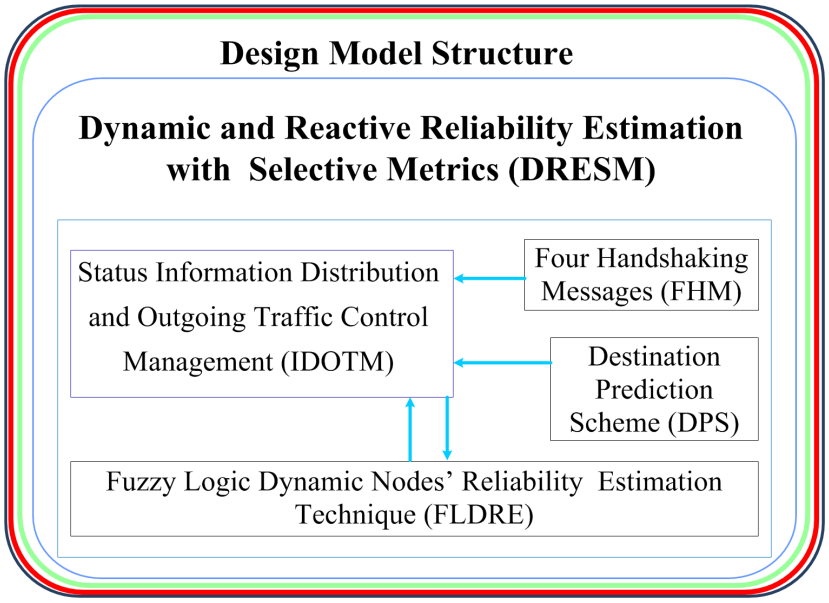


Figure 4.1. DRESM Model Structure Details

The FLDRE is designed to select the optimal next relay-node among the available candidate nodes based on their reliability index. The selection process of the optimal next relay-node requires the implementation of a coordination process among the neighbour nodes and the sender node. Thus, the FLDRE scheme is required for another explicit scheme to exchange the participants' status information and to control the outgoing traffic between them. To fulfil this demand, the IDOTM technique is proposed as the second part of the DRESM mechanism.

The IDOTM technique is designed to reactively allocate and distribute the participating nodes' status information. Also, through using the IDOTM technique, a sender node predicts the destination location by using a destination prediction scheme DPS before sending a packet to the next relay-node. As a consequence, every node in the path can reactively get complete and accurate information about its neighbours and destination nodes. These collected data are then forwarded to the network layer to be used by FLDRE technique.

To sum up, the proposed DRESM algorithm is responsible for constructing the most reliable (optimal) route between communicating nodes in MANET based on multiple objectives. It is responsible for controlling the outgoing traffic and the forwarding decision process using multiple metrics. By deploying DRESM, the GFS algorithm is refined as a localized routing scheme, in the sense that it does not need information

about the entire MANET, but only needs to keep track of neighbours within direct communication range. This adds a level of resilience to dynamic network behaviour. Moreover, the forwarding decision is made at each node based on its own information and a reliability index of its neighbours. Thus, the process avoids the need for costly resource reservations.

4.3 Status Information Distribution and Outgoing Traffic Control Management Detailed Design and Objectives

The status information distribution and outgoing traffic control management IDOTM technique is the key engine to distribute nodes' status information and to control the outgoing traffic in the network. The IDOTM algorithm is only executed when required, i.e., on-demand, when there is DATA packet to be forwarded. One objective of executing IDOTM technique is to discover the candidate nodes, namely, the best next relay-nodes in terms of their reliability index in the sender's transmission range. Another objective is to provide the sender with fresh information about its neighbours and destination node to make routing decisions. The last objective is to control the outgoing traffic.

To satisfy these objectives, small parts of the used Medium Access Control (MAC) are modified in addition to adopting a destination prediction scheme. The destination prediction scheme (DPS) will be presented in Section 4.3.6. The modification for MAC is accomplished in five stages.

i. The address filtering modification;
ii. The four handshaking messages re-construction modification;
iii. The Network Allocation Vector (NAV) timing modification;
iv. The handshaking packets as beacon packet modification; and
v. The forwarding candidate number restriction.

4.3.1 The IEEE.802.11 DCF Address Filtering Modification

In this thesis, the IEEE.802.11 Distributed Coordination Function (DCF) is used [161]. In the IEEE 802.11 DCF, the contention problem between participating nodes for ac-

cessing the shared wireless channel is resolved by utilizing a Carrier Sense Multiple Access mechanism with Collision Avoidance (CSMA/CA) [162, 163]. A full study of IEEE 802.11 DCF can be found in [161].

Based on the objective of the proposed IDOTM design, nodes should have the ability to use the broadcast and the unicast simultaneously. In this research, the Medium Access Control (MAC) is supported based in the works presented in [44, 45, 77]. As suggested in [44, 45, 77], the address filtering on the IEEE 802.11 is disabled (in the software). By doing this, the neighbours of a sender are enabled to receive a copy of the sent packet by using MAC interception. In the improved DCF, the DATA packet is sent as unicast in network layer; simultaneously, the neighbours in the vicinity of a sender node will have a higher probability of delivering the packet to the upper layer due to medium reservation. Thus, IDOTM benefits both broadcast and unicast modes simultaneously.

4.3.2 The Four Handshaking Packets Re-construction Modification

In IDOTM, since IEEE.802.11 MAC DCF is adopted, the Four Handshaking packets are used which are.

i. The Request To Forward message (RTF). RTF message is originated by the node that has DATA packet to be forwarded;
ii. The Clear To Forward message (CTF). CTF message is generated by candidate nodes as a response to RTF message;
iii. The DATA packet (DATA) sent by the sender node; and
iv. The acknowledgement (ACK) packet that is generated by optimal node (or one of the two sub-optimal nodes) and after it receives DATA packet.

The exchange of RTF and CTF packets prior to DATA packet is a sign of the need for medium reservation. Thus, in each of RTF and CTF packets, the duration field should be set to a period of time to reserve the medium, enough to transmit DATA packet and to receive back ACK packet. The goal and re-structure of the four handshaking packets are stated in the following sub-sections.

4.3.2.1 Request To Forward Packet Design

The Request To Forward (RTF) packet is generated by nodes that have DATA packets to be forwarded. As illustrated in Table 4.1 below, the message packet size (MP*size*) of the proposed RTF is 27 bytes. The RTF packet contains the Control Frame (CF), the Duration for MAC usage (Du) in the first and second fields respectively. The source and destination information are listed in fields 3 and 4 respectively. Each field consists of five cells which are: (ID) node identity, e.g., the *IP* address; (x,y) coordinates (latitude and longitude); velocity $(vel.)$; and motion direction $(dir.)$ with respect to x-axis, respectively. Field 5 indicates the time when the source node gets (or predicts) the destination's location. Field 6 indicates RTF sending time. Finally, field 7 indicates the distinction number (hold #2).

Table 4.1. Request To Forward (RTF) Packet Structure, the Shaded Columns are the New Proposal and the Rest as the Old Packet

1	2	3	4	5	6	7
CF	*Du*	*Sender*	*Destination*	t_{des}	t_s	PD_n
						2
2*Bs*	2*Bs*	10*Bs*	10*Bs*	1*B*	1*B*	1*B*

3				
ID_i^s	x	y	$vel.$	$dir.$
4*Bs*	x_{ts}^S	y_{ts}^S	v_{ts}^S	θ_{ts}^S
	2*Bs*	2*Bs*	1*B*	*1B*

4				
ID_i^s	x	y	$vel.$	$dir.$
4*Bs*	x_{ts}^D	y_{ts}^D	v_{ts}^D	θ_{ts}^D
	2*Bs*	2*Bs*	1*B*	*1B*

The network allocation vector (NAV) offers the virtual carrier-sensing function. NAV is piggybacked in MAC's frames header, that is the RTS and beacon packet. In the proposed IDOTM, NAV protection is desired for only one DATA packet in each transmission.

With the scenario in which a node has one positive neighbour, one candidate neighbour, the duration value NAV_{RTF} is set to the time, in microseconds, required to transmit DATA packet, plus one CTF packet, plus one ACK packet, plus 3 Short Interframe Space (SIFS) intervals times.

4.3.2.2 Clear To Forward Packet Design

Candidate neighbours transmit the Clear To Forward (CTF) Packet on the condition that it receives RTF packet and that it has at least one neighbour in the general direction of the final target. Hence, a neighbour benefits source node and can achieve positive progress towards the desired destination.

As illustrated in Table 4.2, the MP*size* of proposed CTF is 19 bytes. CTF contains 8 fields related to candidate node. The first and second fields show the Control Frame CF, the Duration for MAC usage Du. Field 3 holds 5 cells which are; (ID) node identity e.g., the *IP* address; (x,y) coordinates; velocity $(vel.)$; and motion direction $(dir.)$ of candidate node with respect to x-axis, respectively. Fields 4-8 hold the node's buffer free size (φ_C); the reminder battery power as a percentage (ϑ_C); the number of candidate node's positive degree $(\varkappa_C)$; the time that the CTF packet is sent (t_S); and the distinction number PD_n respectively.

Table 4.2. Clear To Forward (CTF) Packet Structure, the Shaded Columns are the New Proposal and the Rest as the Old Packet

1	2	3					4	5	6	7	8
CF	Du	ID_i^C	x	y	$vel.$	$dir.$	φ_C	ϑ_C	$\varkappa_C$	t_S	PD_n
			x_{ts}^C	y_{ts}^C	v_{ts}^C	θ_{ts}^C					3
$2Bs$	$2Bs$	$4Bs$	$2Bs$	$2Bs$	$1B$	$1B$	$1B$	$1B$	$1B$	$1B$	$1B$

With the scenario in which a node has one positive neighbour and one candidate neighbour, the duration value NAV_{CTF} is set to the time, in microseconds, required to transmit the DATA packet, plus one ACK packet, plus two *SIFS* (Short Interframe Space) intervals times.

4.3.2.3 DATA Packet Design

As a node receives CTF packets from its candidate neighbours, it sends a DATA packet for the optimal next relay-node. The distinction number PD_n for a DATA packet is 4. As shown in Table 4.3 below, the data packet header size (if there is just one optimal node) is 34 bytes, and will increase 9 bytes for each extra sub-optimal node. With the

proposed enhancement, the maximum sub-optimal nodes are set to 2 neighbours, with extra load equivalent of 18 bytes.

Table 4.3. DATA Packet Header Fields, the Shaded rows are the New Proposal and the Rest as the Old Packet's Header

Field	Function	Bytes number
CF	Control Frame	2
Du	Duration for MAC usage	2
DL	Destination Location	4
DA	Destination Address	4
HC	Hop Counter	1
DM	Destination mobility attribute	3
PA	Previous Hop Address	4
PL	Previous Hop Location	4
ONA	Optimal Next Relay-node Address	4
ONL	Optimal Next Relay-node Location	4
RSN_0	Reliability sequence number-optimal node	1
SNA_1	Sub-optimal node Address (1)	4
SNL_1	Sub-optimal node Location(1)	4
RSN_1	Reliability sequence number-Sub-optimal node1	1
SNA_2	Sub-optimal node Address (2)	4
SNL_2	Sub-optimal node Location(2)	4
RSN_2	Reliability sequence number-Sub-optimal node2	1
PD_n	Packet distinction number (hold #4)	1

4.3.2.4 Acknowledgement Packets Design

The acknowledged (ACK) packet is emitted by the optimal node, or in the worst case by one of the sub-optimal nodes. In ACK packet, the receiver's address is copied as the same address from the latest received DATA packet. With the proposed IDOTM, the More Fragment bits field in the Frame Control field of the instantly previous sent DATA is to zero because only one DATA packet is being sent in each transmission. As illustrated in Table 4.4 below, the ACK message contains 4 fields related to the sender of the latest received DATA packet.

Table 4.4. Acknowledgment (ACK) Packet Structure

<table>
<tr><td>1</td><td>2</td><td colspan="3">3</td><td>4</td></tr>
<tr><td rowspan="2">CF</td><td rowspan="2">Du</td><td rowspan="2">ID_i</td><td>x</td><td>y</td><td>PD_n</td></tr>
<tr><td>x^i_{ts}</td><td>y^i_{ts}</td><td>5</td></tr>
<tr><td>$2Bs$</td><td>$2Bs$</td><td>$4Bs$</td><td>$2Bs$</td><td>$2Bs$</td><td>$1B$</td></tr>
</table>

The first and second fields show the Control Frame *CF* and Duration for MAC usage *Du* respectively. The third field is extracted from the latest received DATA packet. The DATA duration will be covered in RTF packet. The distinction number *PDn* is 5. The size of the of ACK packet, MPsize, is 13 bytes.

4.3.3 The Network Allocation Vector (NAV) Modification

In IEEE 802.11 DCF, a node that wants to transmit DATA packet must be free for a Distributed Inter Frame Space (*DIFS*). *DIFS* is used for the asynchronous DATA service purpose. *DIFS* is equivalent to *SIFS-Time* plus double time of *Slot-Time*. Typically, *SIFS-Time* and *Slot-Time* are fixed, these intervals set to $10\mu s$ and $20\mu s$ respectively, and thus, *DIFS* was set as $50\mu s$.

As discussed earlier in Sub-sections 4.3.2.2, 4.3.2.3, and 4.3.2.4, NAV time duration is carried in the packet headers of RTF, CTF, DATA and ACK packets. The specified NAV_{RTF}, NAV_{CTF}, and NAV_{ACK} durations are designed for only one DATA packet in each transmission. Moreover, all are designed with the scenario where a sender node has one positive neighbour, that is, one candidate neighbour.

To generalise those rules for a number of positive and candidate neighbours, as the sender node gets the number of positive neighbours, it calculates the required waiting time to receive the CTF packet from up to maximum three of them (candidate neighbours). In this thesis, each CTF packet is jittered by 50% of the *SIFS* interval. Thus, the waiting time in *microseconds* to receive CTF packet from candidate neighbours is equivalent to the time needed to send CTF multiplied by the number of candidate neighbours plus the number of candidate neighbours multiplied by 1.5 *SIFS*. The time for each candidate neighbour to send a CTF packet is uniformly distributed in [0.5 *SIFS* multiplied by the number of candidate neighbours, 1.5 *SIFS* multiplied by the number of candidate neighbours]. The waiting time in *microseconds*, to receive ACK packet from all candidate neighbours is equivalent to the time to send one ACK packet plus the number of candidate neighbours multiplied by *SIFS*. The time for each candidate neighbour to send an ACK packet is calculated incrementally (that is, the optimal node needs 1 *SIFS*; 1^{st} sub-optimal node needs 2 *SIFS* and 2^{ed} sub-optimal node needs 3 *SIFS*).

Based on the above dissection, in this thesis work, the *NAV* duration time and other related periods time can be generalized as follows:

$$NAV_{ARTF} = T_{DATA} + T_{ACTF} + T_{AACK}\ \mu s \tag{4.1}$$

In case a sender node has more than one candidate neighbour, the following equation is used.

$$T_{ACTF} = [(T_{CTF} * N_{cn}) + (N_{cn} * 1.5 * SIFS)]\ \mu s \tag{4.2}$$

In case a sender node has one candidate neighbour, or the next relay-node is the destination, the following equation is used.

$$T_{ACTF} = T_{CTF} + SIFS\ \mu s \tag{4.3}$$

The following equations is used to figure out T_{AACK}.

$$T_{AACK} = [(T_{ACK}) + (N_{cn} * SIFS)]\ \mu s \tag{4.4}$$

In case the next relay-node is the destination, or the sender node has one candidate node, N_{cn} will be set to 1 in Equations 4.4.

where, NAV_{ARTF} is total channel's reservation time required to send a data packet until receiving an ACK packet from one of the candidate nodes, T_{ACTF} is the total time required to receive all CTF from all candidate nodes, T_{AACK} is the total time required to receive one ACK from one of the candidate nodes, N_{cn} is the number of candidate neighbours, T_{DATA} is the time needed to send DATA packet, T_{AACK} is the total time required to receive one ACK from one of the candidate nodes.

4.3.4 RTF/CTF/DATA Beacon Packet Like

To reduce the cost of updating, sender node information is piggybacked on the head of all DATA and other control packets. To facilitate this process, in the IDOTM technique all nodes operate in the promiscuous mode. In the promiscuous mode, participating nodes disable the MAC interface's address filtering. Thus, every node in the range of a sender node can eavesdrop on the sent RTF, CTF and DATA packet and can use it too.

In IDOTM, each proposed RTF, CTF and DATA packet carries beacon-equivalent to the urgent beacon packet (UBM). Thus, a sender node that sends RTF, CTF or DATA packet should reset its clock for the next check time (CHT) after it sent one of those packets.

A node that receives the RTF, CTF, or DATA packet can analyze the piggybacked information in the packet's header and use this information to maintain and update its neighbours-matrix at zero cost. This improvement in utilizing RTF/CTF/DATA packet significantly decreases the updating overhead, decreases the collision probability and effectively improves the wireless channel utilization.

4.3.5 Forwarding Candidate Number Restriction

As a sender node sends RTF packet, all positive neighbours in the direction of the destination will replay with CTF packet. To consider a huge number of positive neighbours to be in the candidate neighbours list may improve the performance, in terms of increasing node availability for each forwarder. However, the overhead and complexity increase as the number of neighbour nodes compete to be optimal nodes. Hence, in this thesis, as suggested in [45] the maximum number of accepted candidate neighbours is limited to 3 neighbours. The rational in limiting the candidate nodes number is limiting the computational load on the head of the source node. In doing so a balance between robustness and overhead is achieved. Additionally, limiting the maximum number of the candidate nodes to 3 provides a benefit by preventing what can be called a "candidate nodes' CTF Storm" in the case of a very dense topology. In the dense networks, the probable ability of nodes to reply to RTF could result in a possible candidate nodes' CTF Storm, which incurs bandwidth wastage and a high probability of increasing the number of network collisions [164, 165]. Moreover, because a copy of the sub-optimal nodes will be maintained at each node, more memory resources will be consumed at each node. Finally, the information of the candidate nodes will be added to DATA packet header, and DATA packet size will be increased proportional to the candidate nodes number, thus introducing more overheads.

4.3.6 Destination Prediction Scheme

The effectiveness of any position-based routing protocol is dependent on the nodes' information accuracy level [49, 50]. With conventional GFS, as in that in which the packet is transmitted from sender node to the next relay-node, the sender does not care if the final destination changes its position during transmission time and results in IDPI problem [62].

The majority of previous proposed works with the position-based routing protocol has used the same destination location information as provided from the location service without re-checking its accuracy. On the other hand, a small number of researchers have proposed an enhanced method to increase the accuracy of the destination location while performing the forwarding process. Most current research which has used mobility predictions, such as [53], are based on piecewise linear node motion with no direction or speed changing considerations.

This research argues here that the final destination may change its position during packet transmission with regards to its speed and motion direction. Thus, the destination's position information in the header of DATA packet becomes obsolete. To address the problem and to support IDOTM technique to control the outgoing traffic, each sender node should predict the destination's location by using the Destination Prediction Scheme (DPS). DPS is inspired from [109, 137] based on dead-reckoning approach, with small alterations as stated earlier in Chapter 3, Sub-section 3.3.6. Each sender node should execute the prediction process before sending a RTF packet to append the result in the RTF header. As the sender node receives the CTF message, it again appends the predicted destination's position information to DATA packet header.

4.4 Fuzzy Logic Dynamic Nodes' Reliability Estimation Technique Detailed Design and Objectives

Greedy Forwarding Strategy (GFS) [23] forwards the data packet-by-packet and node-by-node. Thus, one main benefit of using the GFS routing algorithm is its tendency to weight individual next relay-node choices [57]. Moreover, the optimal route with GFS is always the shortest route [100]. As shown in Chapter 2, GFS cannot always

succeed in guaranteeing the optimal route in MANET. This is for two main reasons. First, GFS was proposed for use in static networks and thus is not compatible with MANET environment. Second, GFS adopted a single routing metric, the hop count, to achieve shortest path objective.

As a solution to GFS failure in MANET, this section introduces the concept of Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE) as a multi-metric approach. To construct the optimal route between communicating nodes, this research creates the concept of neighbour classification, based on its reliability index using the FLDRE technique. Routes can be modified to each DATA packet, node by node, simply by considering a reliability index of the next-hop. The next section presents detailed information about the proposed routing metrics that are used to select the next relay-node.

4.4.1 Detailed Design of the Proposed Prioritization Metrics

In Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE), multiple routing metrics are considered in making the routing decision. The proposed routing metrics give full and detailed information of the intermediate nodes. Those metrics are inspired from two resources. The first one is the preferred feature for any new proposed routing scheme stated in Chapter 2, Section 2.7, and the second resource is the work presented in [48, 166].

As shown in Figure 4.2 below, this research adopts five routing metrics to make the routing decision. These metrics belong to three different categories, which are: nodes' condition, mobility attributes and location accuracy. This classification scheme is adopted based on information contained in Chapter 2, Section 2.5 and [166]. The details of those metrics are introduced in the following Sub-sections.

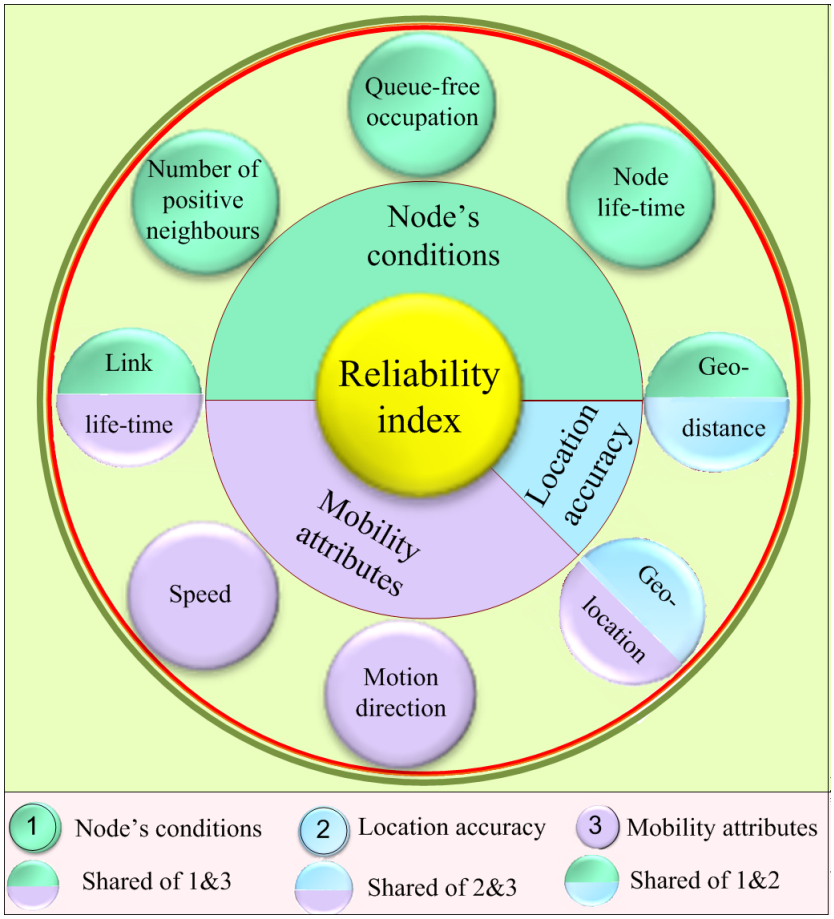


Figure 4.2. Formation of Proposed Metrics

4.4.1.1 Distance to Destination Identification

Distance-to-Destination is the popular traditional metric used with GFS as the selection metric. Distance-to-Destination allows the GFS to find the shortest route between two communicating nodes. The distance metric is used by this research as one of the adopted routing metrics. To clarify this metric, suppose the scenario in which a sender node S needs to calculate its distance to a destination node and the distance between its neighbour node C and destination node D at time t_S, The sender node uses the Pythagorean Theorem formula [147] as in Equation 4.5 below.

$$des_D^{SorC} = \sqrt{\left(x_{t_S}^{SorC} - x_{t_S}^{D}\right)^2 + \left(y_{t_S}^{SorC} - y_{t_S}^{D}\right)^2} \tag{4.5}$$

To normalize a neighbour node's distance to destination, convert the calculated distance to the range [0, 1], the sender node subtracts its distance to destination from the neighbour distance to the destination and divides the result on its transmission range

(R) as shown in Equation 4.6 below [167].

$$\eth_C^S(t_S) = \begin{cases} \frac{des_D^S - des_D^C}{R}, & des_D^S \geq des_D^C \\ \frac{|des_D^C - des_D^S|}{R}, & des_D^S < des_D^C \end{cases} \tag{4.6}$$

where, $\eth_C^S(t_s)$ is the normalized distance of the neighbour node C at time t_S, des_D^S is the distance between the sender and the destination nodes, des_D^C is the distance between neighbour node C and and the destination node, and R is the transmission range of all nodes.

The normalized result of a neighbour node is a sign of its closeness to the ultimate destination. If the ratio is close to 1, then the neighbour is the closest to final destination and vice versa.

4.4.1.2 Motion Speed and Direction

Mobility metrics in MANET affect the performance of routing protocols. Among these metrics, a node's movement speed and direction are two of the most critical metrics [27, 47, 48]. If next relay-node has a high difference in speed and\or direction in comparison with the sender node, then packet loss probability is increased due to an unstable link [109]. This inspired the researcher to take both motion's direction and speed into consideration when selecting an optimal next-hop. With this research, a trade-off between speed and direction as a velocity vector is used. By using this trade-off, the packet sender selects the next relay-node using the residual link lifetime metric in addition to the other metrics. A detailed mathematical model is introduced in the following section.

4.4.1.3 Residual Links Lifetime

The link duration of a routing path is limited to any single node's link age selected as a member of this path. Thus, the link lifetime between communicating nodes is one of the important issues to be considered in routing algorithm [106, 107, 108]. In this work the link lifetime between two communicating nodes is named as Residual Link Lifetime (RLT).

RLT is another metric to be considered in the selection process in this research. The intermediate nodes with a high value of RLT should be preferred with respect to those with a lower value. By performing this kind of fairness, all nodes can equally contribute to the packet's routing effort. The procedure to calculate RLT was mentioned in Chapter 3, Sub-section 3.4.1. In this research work, a node moves at a random distributed speed in the range [0, 40] $m\backslash s$ and distributed direction in the range [0, 2π]. Hence, the maximum and minimum magnitude of the relative velocity between two nodes is 80, and 0 respectively. Because the used transmission range is fixed for all participating nodes (R=250m), the maximum and minimum values of RLT is set to the range $[0.0s, 250s]$. To map RLT range to [0,1], as a normalization process, Equation 4.7 is used [167].

$$\Im_C^S(t_S) = \frac{\left(RLT_C^S(t_S) - \min\{0.0, 250\}\right)}{\left(\max\{0.0, 250\} - \min\{0.0, 250\}\right)} = \frac{RLT_C^S(t_S)s}{250s} \tag{4.7}$$

where, $\Im_C^S(t_S)$ is the normalized value of RLT between the nodes S, and C at time t_S.

The value $\Im_C^S$ of any neighbour is considered as an indicator for the reliability level of communication via this node. Thus, nodes are fully connected if $\Im_C^S$ is 1 and likely out of transmission range of each other if $\Im_C^S$ is 0.

4.4.1.4 Unoccupied Buffer Length

Queue size is a property of MANET's nodes. Usually, the queue size is limited to a few packets, and nodes start to drop packets when the queue is full [39, 40, 102]. The queue length is considered to be an indicator for the congestion level of the neighbour.

For optimal usage of network resources, this research balances the traffic load evenly. Nodes' buffers should be uniformly used and several nodes alone should not be overused. By achieving this goal, congestion in the centre area of MANET could be decreased, leading to fewer collisions and saving nodes' energy as well. Thus, the efficiency of GFS is improved.

A node's unoccupied buffer length (f_{bl}), set as a function, varies with time. The f_{bl} can be determined based on the size of data packets currently buffered to the node's

buffer size [39]. In this research, the buffer size for all mobile nodes is assumed to be equal. Thus, a node's f_{bl} (say node C) can be estimated using the Equation 4.8 below [39].

$$f_{bl}^{C}(t_S) = T_{buffersize}^{C} - T_{occupied}^{C}(t_S) \tag{4.8}$$

where, $f_{bl}^{C}(t_S)$ is the unoccupied buffer length at time t_S, $T_{buffersize}^{C}$ is the total buffer size of node C, and $T_{occupied}^{C}(t_S)$ is the occupied buffer at time t_S. To normalize the unoccupied length, a candidate node uses the Equation 4.9, as below [167].

$$\varphi_C(t_S) = \frac{\left(f_{bl}^{C}(t_S) - \min\left\{0.0, T_{buffersize}^{C}\right\}\right)}{\left(\max\left\{0.0, T_{buffersize}^{C}\right\} - \min\left\{0.0, T_{buffersize}^{C}\right\}\right)} = \frac{f_{bl}^{C}(t_S)}{T_{buffersize}^{C}} \tag{4.9}$$

The node is fully congested if the φ_C ratio is 0, and uncongested if the ratio is 1. In order to distribute traffic effectively, each neighbour sends the level of its φ_C to the source/forwarder node to help that node select the appropriate next relay-node with preference given to less-congested neighbour.

4.4.1.5 Residual Battery Power

In MANET, the node's resources are limited in several aspects. Mobile nodes depend on finite battery sources, and thus, power consumption is one of the most critical issues. In MANET, the battery power consumed for the communication is either higher or comparable with energy consumed by the processor [41, 42, 43]. However, with a conventional GFS approach, the batteries of certain nodes at the centre of MANET may drain, even though there are many nodes with plenty of energy, such that this drainage disables further information delivery and induces packet loss and more delay.

For optimal usage of network resources, a part of the proposed work is to be energy-adaptive. The intermediate nodes with a high value of the remaining battery lifetime should be preferred to those with a lower value. By performing this kind of fairness, all nodes can equally contribute to the packet routing effort.

A node's residual battery ratio b_{po} can be determined as a function varying with time.

b_{po} can be designated based on the ratio between the instant consumed node's battery power to the node's total battery energy size. In this research, the battery power for all mobile nodes is assumed to be equal. A node's (say node C) residual battery power status b_{po} at any instant time (say t_S) can be calculated using Equation 4.10 [168].

$$b_{po}^{C}(t_S) = Tb_{po}^{C} - TC_{po}^{C}(t_S) \tag{4.10}$$

where, $b_{po}^{C}(t_S)$ is the residual battery energy at time t_S, Tb_{po}^{C} is the battery energy size of node C, and $TC_{po}^{C}(t_S)$ is the consumed battery power size up to time t_S.

To normalize the nodes' residual battery power status, a node uses Equation 4.11 as below [167].

$$\vartheta_C(t_S) = \frac{\left(b_{po}^{C}(t_S) - \min\left\{0.0, Tb_{po}^{C}\right\}\right)}{\left(\max\left\{0.0, Tb_{po}^{C}\right\} - \min\left\{0.0, Tb_{po}^{C}\right\}\right)} = \frac{b_{po}^{C}(t_S)}{Tb_{po}^{C}} \tag{4.11}$$

High values of neighbour's $\vartheta_C(t_S)$ provide an indicator for the high reliability value of communication via this neighbour. The node has full battery energy if the ϑ_C ratio is 1 and likely out of energy if the ratio is 0. In order to distribute traffic effectively, each neighbour sends the level of its ϑ_C to the source/forwarder node to help that node select the appropriate next relay-node with preference given to the highly powered one.

4.4.1.6 Next Relay-node Positive Degree

A node degree is defined by how many other nodes can be reached from this node at a given moment, that is, it's the total number of its neighbour in all directions [104, 105]. However, with respect to the connectivity issue of MANET, a part of it is considered as a disconnected or an isolated part, in the case in which some nodes do not have neighbours in any direction. These types of nodes can result in degraded network performance.

This research is aimed primarily at developing a method that maximizes the connectivity of MANET. Thus, the appearance of GFS failure due to link unavailability is minimized. With the enhanced GFS, each positive neighbour receiving a RTF mes-

sage should first check its neighbours' matrix. As each positive neighbour becomes sure that it has at least one neighbour that guarantees the positive progress towards ultimate destination, it will send back CTF message and append the number of its positive neighbours (NPN) to the header of CTF packet. If the positive neighbour does not satisfy this criterion, then it will not send back CTF message.

The advantages of using the NPN check process is to assure that DATA packet does not halt at the next-hop, reducing the chances of hitting a hole. Also, by using this method at the sender side, the proposed solution is considered to be a 2-hop map like (a node's neighbours and a node's neighbour's neighbours). The 2-hop map like is useful for speeding up route set-up. The goal here is to achieve a route between source and destination nodes in which none of the route members is isolated.

With respect to a node's mobility, a node's NPN could be any value in the range from 0 to its degree at time t_S [169]. For a node (say node C) NPN range could be written as $[0, N^C_{degree}]$ [169]. To normalize NPN range, a node uses Equation 4.12 as below [167].

$$\varkappa_C(t_S) = \frac{\left(NPN(t_S) - \min\left\{0, N^C_{degree}(t_S)\right\}\right)}{\left(\max\left\{0, N^C_{degree}(t_S)\right\} - \min\left\{0, N^C_{degree}(t_S)\right\}\right)} = \frac{NPN(t_S)}{N^C_{degree}(t_S)} \tag{4.12}$$

where, for node C, $\varkappa_C(t_S)$ is its normalized value of its NPN at time t_S, $N^C_{degree}(t_S)$ is its degree at time t_S, and NPNt_S is its positive neighbours number in the direction of the ultimate destination.

A node that receives the CTF message with references to $\varkappa_C$ ratio can use it as an additional metric to determine the neighbour reliability index. The node is sufficient if $\varkappa_C$ ratio is 1 and insufficient if the ratio is 0.

4.4.2 Dynamic Reliability Estimation using Fuzzy Logic

The proposed metrics mentioned in the previous section are proposed to achieve the multi-routing objectives scheme. Such a routing process requires that these metrics be expressed together in a one analytical form for evaluation. If the routing metrics

mentioned in the previous section are utilized as one unit, they can lead to a tremendous computational burden. This is one area in which the intelligent method may be beneficial. As stated earlier in Chapter 2, Section 2.8.3, the Fuzzy Logic Controller (FLC) [130] is one of the most suitable systems to solve such a problem in MANET [60, 62, 136].

To estimate the candidate nodes' reliability index, a weighting method should be used adjustably, such that the combined proposed metrics can achieve an optimal utility. In this chapter, FLC is applied for finding out the reliability index of each candidate node, based on the 5 proposed routing metrics. After a sender node receives CTF message, it does not know which candidate neighbour is the most reliable, so Fuzzy Logic Dynamic Reliability Estimation (FLDRE) as FLC, is the answer for this ambiguous type of problem.

In the proposed FLDRE as FLC, the 5 proposed metrics are utilized as the input parameter and node's Reliability INdex (RIN) as the output parameter. To create and edit FLDRE, the researcher uses the Fuzzy Logic Toolbox. This toolbox is a collection of functions built on the MATLAB. It provides a graphical user interface (GUI) tools to help entirely accomplish the intent work from only the command line. Figure 4.3 below shows the FLC design of the FLDRE model.

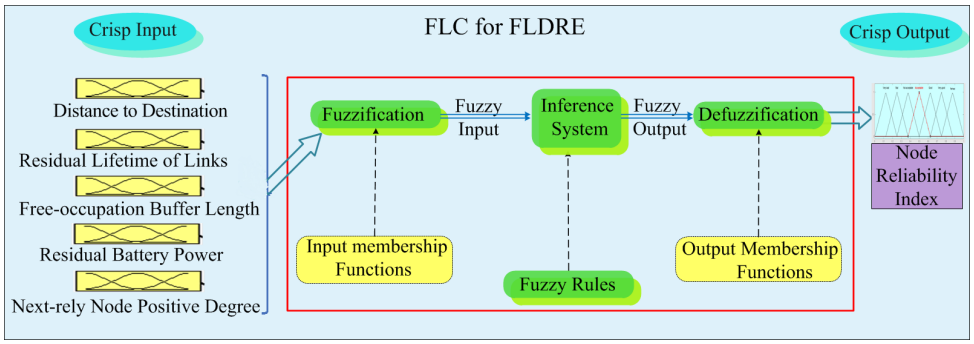


Figure 4.3. Generalized Fuzzy System FLC for FLDRE

As shown in Figure 4.3 above, the first step of designing FLDRE is to arrange the membership functions of the input and output fuzzy variables relying on the defined range. The next is to construct appropriate rules for FLDRE. The inference engine, with the aid of the proposed rules, is used to control the action in the linguistic form. Then, the fuzzy output is defuzzyfied using membership functions to generate the crisp

output.

With FLDRE, the fuzzifier maps the crisp input values to fuzzy sets and assigns degree of membership for each fuzzy set. Each one of the 5 metrics (crisp input) is fuzzified to 3 fuzzy sets and RIN crisp output is fuzzified to 7 fuzzy sets. The used membership functions for the FLDRE model are the Z-shaped, triangular, and S-shaped shapes as suggested by [60, 62, 134, 136]. As mentioned in Chapter 3, Sub-section 3.3.4.1 adjacent fuzzy sets are overlapped between 10% and 50% [131]. The overall process involved in estimating RIN of the candidate neighbours is elaborated in the following sub-sections.

4.4.2.1 Fuzzify Selected Metrics Input

The linguistic values used to represent the normalized $\eth_C^S$ (i.e., distance metrics des_C^S) are divided into three levels: close (*cs*), medium (*md*), and far (*fr*); and those used to represent $\Im_C^S$ (i.e., nodes' Residual Link Lifetime RLT), φ_C (i.e., nodes' Unoccupied Buffer Length f_{bl}), ϑ_C (i.e., nodes' Residual Battery Power b_{po}), and $\varkappa_C$ (i.e., next relay-node's positive neighbours number NPN) are also divided into three levels: low (*lo*), medium (*md*), and high (*hi*).

The crisp input values of $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, and $\varkappa_C$ variables are fuzzified between *min-value* = *zero* and *max-value* = 1. Table 4.5 and Figure 4.4 show the assignment of range and degree of membership functions of the five metrics.

Table 4.5. Linguistic variable: $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, or $\varkappa_C$

Numerical range (normalized)	Linguistic value	Notation
Z-curve parameters (a, b, c) = (0.0, 0.25, 0.5)	Far\Low	*fr\lo*
Tria.-curve parameters (a, b, c) = (0.25 0.5 0.75)	Medium	*md*
S-curve parameters (a, b, c) = (0.5, 0.75, 1.0)	Close\High	*cs\hi*

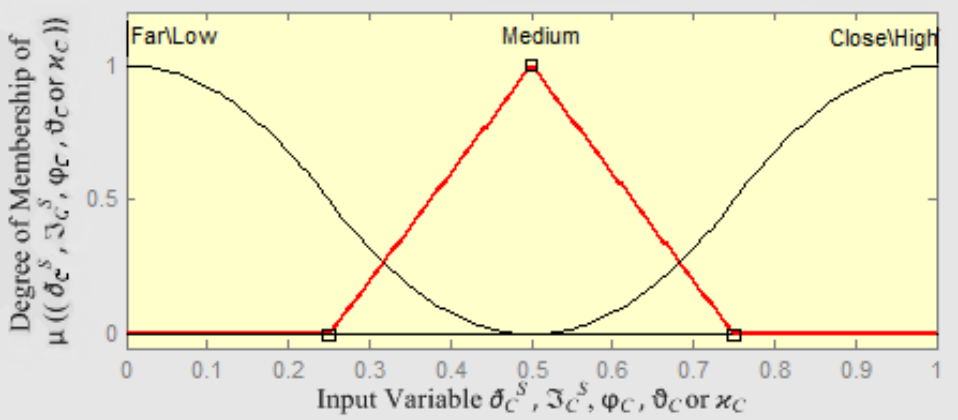


Figure 4.4. Membership Functions of $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, and $\varkappa_C$ Input Variable

The input values of $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, and $\varkappa_C$ can be interpreted as: "Far\Low" as "an input value below about 0.25", "medium" as "an input value close to 0.5", and "Close\High" as "an input value above about 0.75", otherwise, the input value can located in two range, for example if the $\eth_C^S$ is 0.3, then node distance to destination has a far membership value of 0.72 and it has a medium membership value of 0.2.

Based on the above discussion and the allowed overlapping ratio [131], this research uses 50% overlapping between adjacent fuzzy sets of the input variables $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, and $\varkappa_C$. Based on [130, 131, 132], the following equations gives the explicit formulas for the selected membership functions of the input variables $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C, and $\varkappa_C$.

$$\mu(x_{fr-lo}, 0, 0.25, 0.5) = \begin{cases} 1, & x \leq 0.0 \\ 1 - 2\left(\frac{x-0.0}{0.5-0.0}\right)^2, & 0.0 \leq x \leq 0.25 \\ 2\left(\frac{x-0.5}{0.5-0.0}\right)^2, & 0.25 \leq x \leq 0.5 \\ 0, & x \geq 0.5 \end{cases} \tag{4.13}$$

$$\mu(x_{md}, 0.25, 0.5, 0.75) = \begin{cases} \left(\frac{x-0.25}{0.5-0.25}\right), & 0.25 \leq x \leq 0.5 \\ \left(\frac{0.75-x}{0.75-0.5}\right), & 0.5 \leq x \leq 0.75 \\ 0, & otherwise \end{cases} \tag{4.14}$$

$$\mu(x_{cs-hi}, 0.5, 0.75, 1.0) = \begin{cases} 0, & x \leq 0.5 \\ 2\left(\frac{x-0.5}{1.0-0.5}\right)^2, & 0.5 \leq x \leq 0.75 \\ 1-2\left(\frac{x-1.0}{1.0-0.5}\right)^2, & 0.75 \leq x \leq 1.0 \\ 1, & x \geq 1.0 \end{cases} \tag{4.15}$$

The x in the previous equations represents any of the input variables $\eth_C^S$, $\Im_C^S$, φ_C, ϑ_C or $\varkappa_C$.

4.4.2.2 Fuzzify Reliability Index Output

The consequence part (the possibility that a node will be selected) is divided into 7 levels. Fuzzy sets for the RIN output variable have the following names: very bad (*vb*), bad (*bd*), not acceptable (*na*), acceptable (*ac*), good (*gd*), very good (*vg*), and perfect (*pt*). The assignment range for the output RIN is stated in Table 4.6 below and Figure 4.5 below depicts its membership functions. The RIN is fuzzified between RIN-*min* = *zero* and RIN-*max* = 1.

Table 4.6. Linguistic variable: Reliability Index (RIN)

Numerical range	Linguistic value	Notation
Tria.-curve parameters (a, b, c) = (0, 0.125, 0.25)	Very bad	*vb*
Tria.-curve parameters (a, b, c) = (0.125, 0.25, 0.375)	Bad	*bd*
Tria.-curve parameters (a, b, c) = (0.25, 0.375, 0.5)	Not acceptable	*na*
Tria.-curve parameters (a, b, c) = (0.375, 0.5, 0.625)	Acceptable	*ac*
Tria.-curve parameters (a, b, c) = (0.5, 0.625, 0.75)	Good	*gd*
Tria.-curve parameters (a, b, c) = (0.625, 0.75, 0.875)	Very good	*vg*
Tria.-curve parameters (a, b, c) = (0.75, 0.875, 1)	Perfect	*pt*

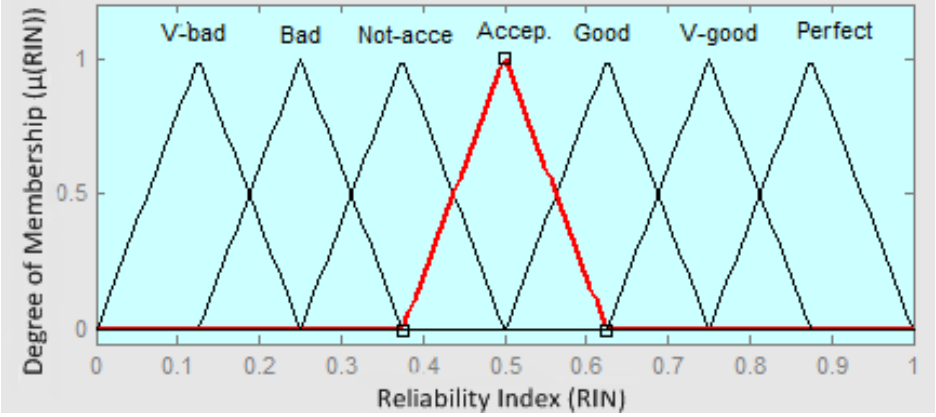


Figure 4.5. Membership Functions for RIN Output Variable

The RIN's output values can be interpreted as: "very bad" as "a RIN close to 0.125", "bad" as "a RIN close to 0.25", "not acceptable" as "a RIN close to 0.375", "acceptable" as "a RIN close to 0.5", "good" as "a RIN close to 0.625", "very good" as "a RIN close to 0.75" and "perfect" as "a RIN close to 0.875". Otherwise, the RIN can located in two ranges. For example if the RIN is 0.8, then RIN has very good membership value of 0.6 and it has perfect membership value of 0.4.

Based on the above discussion and the allowed overlapping ratio [131], this research uses 50% overlapping between adjacent fuzzy sets of RIN . The following equations gives the explicit formulas for the selected membership functions of RIN based on [130, 131, 132].

$$\mu(RIN_{vb}, 0.0, 0.125, 0.25) = \begin{cases} \left(\frac{RIN-0.0}{0.125-0.0}\right), & 0.0 \leq x \leq 0.125 \\ \left(\frac{0.25-RIN}{0.25-0.125}\right), & 0.125 \leq x \leq 0.25 \\ 0, & otherwise \end{cases} \tag{4.16}$$

$$\mu(RIN_{bd}, 0.125, 0.25, 0.375) = \begin{cases} \left(\frac{RIN-0.125}{0.25-0.125}\right), & 0.125 \leq x \leq 0.25 \\ \left(\frac{0.375-RIN}{0.375-0.25}\right), & 0.25 \leq x \leq 0.375 \\ 0, & otherwise \end{cases} \tag{4.17}$$

$$\mu(RIN_{na}, 0.25, 0.375, 0.5) = \begin{cases} \left(\frac{RIN-0.25}{0.375-0.25}\right), & 0.25 \leq x \leq 0.375 \\ \left(\frac{0.5-RIN}{0.5-0.375}\right), & 0.375 \leq x \leq 0.5 \\ 0, & otherwise \end{cases} \tag{4.18}$$

$$\mu(RIN_{ac}, 0.375, 0.5, 0.625) = \begin{cases} \left(\frac{RIN-0.375}{0.5-0.375}\right), & 0.375 \leq x \leq 0.5 \\ \left(\frac{0.625-RIN}{0.625-0.5}\right), & 0.5 \leq x \leq 0.625 \\ 0, & otherwise \end{cases} \tag{4.19}$$

$$\mu(RIN_{gd}, 0.5, 0.625, 0.75) = \begin{cases} \left(\frac{RIN-0.5}{0.625-0.5}\right), & 0.5 \leq x \leq 0.625 \\ \left(\frac{0.75-RIN}{0.75-0.625}\right), & 0.625 \leq x \leq 0.75 \\ 0, & otherwise \end{cases} \tag{4.20}$$

$$\mu(RIN_{vd}, 0.625, 0.75, 0.875) = \begin{cases} \left(\frac{RIN-0.625}{0.75-0.625}\right), & 0.625 \leq x \leq 0.75 \\ \left(\frac{0.875-x}{0.875-0.75}\right), & 0.75 \leq x \leq 0.875 \\ 0, & otherwise \end{cases} \tag{4.21}$$

$$\mu(RIN_{pt}, 0.75, 0.875, 1.0) = \begin{cases} \left(\frac{RIN-0.75}{0.875-0.75}\right), & 0.75 \leq x \leq 0.875 \\ \left(\frac{1.0-RIN}{1.0-0.875}\right), & 0.875 \leq x \leq 1.0 \\ 0, & otherwise \end{cases} \tag{4.22}$$

4.4.2.3 Fuzzy Rules and Fuzzy Inference

With the proposed FLC system, 5 input variables are used, each of which is comprised of 3 fuzzy linguistic values. To map the fuzzy 5 metrics input sets into fuzzy RIN output sets, a set of fuzzy if-then rules is in need. A large number of the candidate fuzzy if-then rules cause high processing cost. The task is to select a small and comprehensible number of simple fuzzy if-then rules that can represent the proposed model with high classification performance.

Because the 5 input variables have similar fuzzy linguistic values, this assists the researcher to select a small number of simple fuzzy if-then rules. This is because many

of fuzzy if-then rules are very similar. Others, while appearing to be different, may in fact match similar sets of records. On other words, the weight (w) of the proposed metrics is equal, i.e., $w_{\eth_D^C} = w_{\Im_C^S} = w_{\varphi_C} = w_{\vartheta_C} = w_{\varkappa_C} = 0.2$. Appendix *B* introduces the proposed if-then rules. Below, a part of the proposed rules is introduced.

RULE **1**: IF the values of the five metrics are *low* THEN RIN is *very bad*

⋮ ⋮ ⋮ ⋮ ⋮

RULE **21**: IF the values of the five metrics are *high* THEN RIN is *perfect*

Mamdani's fuzzy inference method [133] is used to evaluate all the fuzzy rules and finds their antecedent part firing strength, then applies this firing strength to the consequence part of the rules.

4.4.2.4 Defuzzification

Defuzzification is the final step; it refers to the way a crisp value is extracted from a fuzzy set as a representation value. The single output RIN for each candidate node is calculated by using Centroid method [135] as described in Equation 4.23 below.

$$RIN = \Sigma_{j=1}^{w} \mu(x_j) * x_j / \sum_{j=1}^{w} \mu(x_j) \tag{4.23}$$

All the five different input criteria in the defuzzification procedure have the same weights, i.e., a fifth of the final weight.

$$w_{\eth_D^C} + w_{\Im_C^S} + w_{\varphi_C} + w_{\vartheta_C} + w_{\varkappa_C} = 1 \tag{4.24}$$

$$where, w_{\eth_D^C} = w_{\Im_C^S} = w_{\varphi_C} = w_{\vartheta_C} = w_{\varkappa_C} = 0.2$$

4.5 Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism Functionality with An Illustrative Example

This section presents the functionality of the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism Functionality (DRESM) from the perspective of participating nodes. Also, it presents the workflow of the proposed DRESM.

4.5.1 Participating Nodes Procedures

In Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM) scheme, all participating nodes in the network should perform DRESM when a node has DATA packets to be forwarded. To perform routing process, each node in the network should maintain three matrices and one table. These matrices are Destinations' Location Information (DLI), nodes' neighbourhood's location-matrix (NLM), and Reliability Index of Candidate Neighbours (RICN). The table is the DATA LisT (DALT), which is used by a node to keep the information received in the DATA packet's header if it is assigned as sub-optimal node. DLI is used to maintain the destinations' information, NLM is used to maintain neighbours' information, and RICN is used to save the calculated reliability index of the candidate nodes. These matrices and the table should be empty as a node enters the network.

This section explains the operations of both IDOTM and FLDRE schemes as one unit under DRESM. As shown in Figure 4.6 below, the discussion is from the perspective of source node, positive, candidate, optimal, sub-optimal and other neighbours.

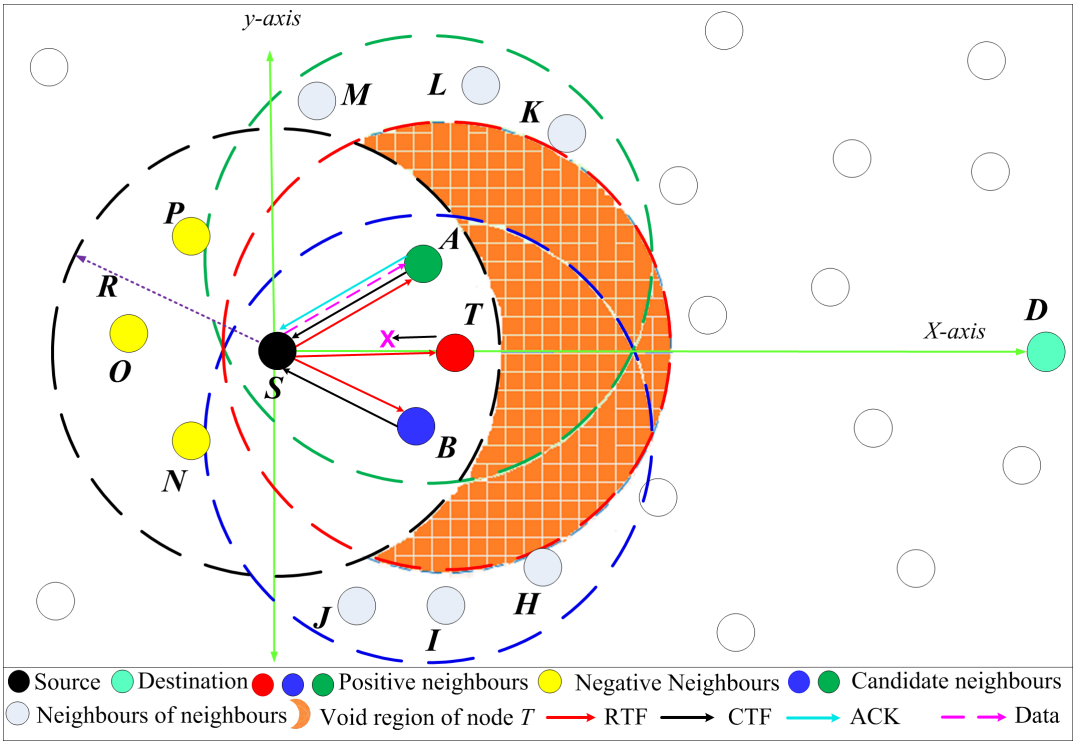


Figure 4.6. Network Topology to Show DRESM Functionality

As shown in Figure 4.6 above, source node *S* intends to transmit a DATA packet to the destination node *D*. The scenario, supposes that the source node *S* is located at the

centre of a circle with (*0*,*0*) coordinates. The radius of the circle is R, where R is the S's maximum transmission range. Source node S has positive neighbours A, B, and T, and negative neighbours P, O, and N. Hereunder are procedures that source node S and other nodes execute to send the DATA packet to node D.

4.5.1.1 Source Node

At a very early stage of the IDOTM algorithm, source node S obtains (or predicts) the destination D's information and updates its DLI matrix. The DLI matrix should be empty as the node enters the network area. Equation 4.25, shows the containment of DLI for i-destination nodes. Each entry for specific destination node holds 8 cells which are: destination address ID_i, destination's geographical location as (x_i, y_i) coordinates, velocity (v_i), acceleration (a_i), motion direction (θ_i) with respect to x-axis, the time that the location server sent the destination's location (t_S), and the instant time the source node receives\predicts the destination's location (t_r) respectively. The acceleration a_i of the destination is calculated by using the Equation 3.12 as in Chapter 3, of the latest two entries of the same destination.

$$DLI = \begin{bmatrix} ID_1 & x_1 & y_1 & v_1 & a_1 & \theta_1 & t_1 & t_2 \\ ID_2 & x_2 & y_2 & v_2 & a_2 & \theta_2 & t_1 & t_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_i & x_i & y_i & v_i & a_i & \theta_i & t_S & t_r \end{bmatrix} \tag{4.25}$$

After node S maintains destination information in DLI matrix, it calculates its distance to D, i.e., des_D^S, by using Equation 4.5. Next, it starts to look up its NLM to check if node D is one of its neighbours. In the case node D is not one of its neighbours, then it again looks up its NLM to check if valid positive neighbours for that destination are available. As a result of searching process, node S may or may not have positive neighbours. In the case that it has no neighbours; node S buffers the packet and moves until neighbours in the direction of destination become available rather than simply dropping the packet. The allowed carry and move time is equal to the packet's TTL. In this research, the proposed TTL is equal to $30s$. The carry and move strategy which is used with the proposed solution is based on the mobility features of the MANET

environment.

Back to the example in which node S found that it has the positive neighbours A, B, and T in the direction of targeted destination. Next, it makes sure that the channel is free for a DIFS period to avoid any contender. If the channel is idle, node S sends a RTF packet after the DIFS countdown timer comes to zero.

In case the channel is busy and the node S has to wait for at least interframe space DIFS, then it must additionally select a random backoff time (in order to minimize collision probability) in the interval $[0, \mathrm{CW}]$. After selecting the backoff time, node S sets the clock countdown for DIFS interval ($50\,\mu s$), and then it decrements the backoff timer in units of a SlotTime. As the backoff countdown timer comes to *zero*, node S ceases the check time (CHT) countdown timer, and performs one of the next two procedures. If the destination is one of its neighbours, it sends the RTF packet to this neighbour; or else it sends the RTF packet to all of its positive neighbours (i.e., nodes A, B, and T) to discover its candidate neighbours in its transmission range. At the header of RTF packet, node S adds the NAV_{ARTF} duration. Also, it sets the timer countdown to receive back CTF packets from maximum three of its neighbours. If node S did not receive a CTF packet, then it again tries to transmit the RTF packet until it receives CTF packet or until the 7 retry limit is reached, whichever occurs first [170]. If the timer and the retry number expire before receiving CTF from any neighbour, then node S will re-set the timer for a new CHT, buffers the packet and move.

In this scenario, it is assumed that S's candidate neighbours are A and B. Node T is not considered as candidate neighbour because it does not have positive neighbours in the direction of destination D. Thus, node T does not send back a CTF packet as a response to a RTF packet. Also, it is assumed that the $(ID_C, x_{ts}^C, y_{ts}^C, v_{ts}^C, \theta_{ts}^C, \varphi_C, \vartheta_C, \varkappa_C)$ in the received CTF packets of both A and B are: $(3, 90, 70, 28, 178^\circ, 0.65, 0.72, 0.72)$ and $(4, 90, -80, 32, 180^\circ, 0.6, 0.65, 0.7)$ respectively. Furthermore, it is presumed that the information of S and D nodes at the time of receiving CTF messages (tr) are : $(ID_S, x_{tr}^S, y_{tr}^S, v_{tr}^S, \theta_{tr}^S) = (1, 0, 0, 30, 0^\circ)$, $(ID_D, x_{tr}^D, y_{tr}^D, v_{tr}^D, \theta_{tr}^D) = (2, 780, 0, 20, 110^\circ)$respectively.

As the source node receives the CTF packets from the candidate nodes, it extracts the information from each packet's header in order to maintain its NLM. Based on the updated information in its NLM, the source node calculates and updates the RLT_C^S and ELT_C^S for each candidate neighbour in its NLM. As shown in Equation 4.26, it is assumed that the first two rows of node *S's* NLM are belonging to the candidate neighbours *A* and *B*.

$$NLM = \begin{bmatrix} 3 & 125 & 135 & 28 & 0 & 178^\circ & 63.4 & 63.4 & t_s & t_r \\ 4 & 114 & -150 & 32 & 0 & 180^\circ & 65 & 65 & t_s & t_r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_i^n & x_i & y_i & v_i & a_i & \theta_i & RLT_i^S & ELT_i^S & t_s & t_r \end{bmatrix} \tag{4.26}$$

where, for any node i, ID_i^n is the identity of each node with the sequence number n, (x_i, y_i) is the node's current position along *x-axis* and *y-axis* respectively, v_i is the velocity in $(m\backslash s)$, a_i is the acceleration in $(m^2\backslash s)$, θ_i is the motion direction with respect to x-axis, RLT is the residual link lifetime between the node S and its neighbour i, ELT is the entry lifetime for the neighbour i in node S's NLM, t_S is sending time of latest received packet from that neighbour, e.g., UBM, RTF, CTF or DATA packet, at the sender side, t_r is the receiving time of UBM, RTF, CTF or DATA packet at the receiver side.

After it maintained and updated its NLM for the two candidate nodes, source node S is ready to maintain the Reliability Index of Candidate Neighbours (RICN) matrix. In this research, the RICN capacity is designed for maximum three candidate nodes (i.e. 3 rows of 8 columns). A RICN network matrix for three candidate neighbours looks like the following as in Equation 4.27.

$$RICN = \begin{bmatrix} ID_1 & \eth_1 & \Im_1 & \varphi_1 & \vartheta_1 & \varkappa_1 & RIN_1 & RSN_0 \\ ID_2 & \eth_2 & \Im_2 & \varphi_2 & \vartheta_2 & \varkappa_2 & RIN_2 & RSN_1 \\ ID_3 & \eth_3 & \Im_3 & \varphi_3 & \vartheta_3 & \varkappa_3 & RIN_3 & RSN_2 \end{bmatrix} \tag{4.27}$$

The candidate neighbours are arranged in RICN in descending order with regards to their RIN values. The RSN value (0,1 or 2, for maximum 3 candidate nodes) is very

important to every candidate node, so that it can easily know when to forward DATA packet if no other node forwarded it. The node that has the lowest RSN in RICN, will have the highest priority to forward DATA packet, i.e. the optimal node (which has the highest reliability value), i.e. $RSN = 0$.

If all candidate nodes have the same reliability index, they are arranged in descending order based on their *ID* number. Likewise, if source node has to choose between two or more neighbours that have the same RIN level, it can do that by referring to their *IDs* number.

Back to the example, to maintain the RICN matrix for the two candidate nodes, source node *S* performs the following procedures. First, node *S* calculates des_D^C, $\eth_C^S$, and $\Im_C^S$ of the two candidate nodes by using Equations 4.5, 4.6, and 4.7 respectively. Next, source node *S* copies the ID_C, i.e., ID_i^n in NLM, φ_C, ϑ_C, and $\varkappa_C$, from the header of received CTF packet besides the calculated values $\eth_C^S$, and $\Im_C^S$ to the RICN matrix. Up to now, besides ID_C, RICN matrix is constructed from other 5 data elements that shape the crisp data to be used in FLDRE technique to calculate nodes *A* and *B* reliability index (RIN). The seventh column in RICN matrix will be the RIN_C, which is collected as a crisp output from FLDRE for each candidate node. The last column in RICN matrix is the Reliability Sequence Number (RSN) of each node.

Based on its calculation, node *S* finds that the values of $\eth_C^S$, and $\Im_C^S$ are {(0.348, 0.254), (0.344, 0.26)} for the two candidate nodes *A* and *B* respectively. The calculation is explained in Appendix *C*. Thus, RICN matrix looks like the following as in Equation 4.28.

$$RICN = \begin{bmatrix} 3 & 0.348 & 0.254 & 0.65 & 0.72 & 0.72 & RIN_A & RSN_A \\ 4 & 0.344 & 0.26 & 0.6 & 0.65 & 0.7 & RIN_B & RSN_B \end{bmatrix} \tag{4.28}$$

Now, source node is ready to execute FLDRE algorithm to estimate RIN_C of the two candidate neighbours. Source node found that the RIN_C of node 3 and 4 are 0.552 and 0.469 respectively, the calculation is explained in Appendix *C*. Equation 4.29 shows the final formation stage after the source node executed the FLDRE functions.

$$RICN = \begin{bmatrix} 3 & 0.348 & 0.254 & 0.65 & 0.72 & 0.72 & 0.552 & 0 \\ 4 & 0.344 & 0.26 & 0.6 & 0.65 & 0.7 & 0.469 & 1 \end{bmatrix} \tag{4.29}$$

After finding the RIN for the two candidate nodes, the source node *S* is ready to send the DATA packet to the optimal next relay-node number 3. Before doing that, node *S* appends the intended details from RICN matrix to DATA packet header, sets the hop counter to 1, and transmits the packet as unicast to the optimal node (i.e. node *B*) in IP layer. The candidate details containment in DATA packet header is updated hop-by-hop, because every forwarder node, between the source and destination will follow the same procedure as the source node did. Based on Table 4.3, Table 4.7 below shows a part of DATA packet header with respect to optimal and sub-optimal nodes with their RSN.

Table 4.7. A part of the DATA Packet Header Shows the Optimal and Sub-optimal Nodes with their RSN

Field	**Values based on the example**
⋮	⋮
ONA	Node A:2
ONL	Node A(125,135)
RSN_0	0
SNA_2	Node B:3
SNL_2	Node B(114,-150)
RSN_2	1
⋮	⋮

Based on the number of candidate nodes, the source node sets its timer to receive one ACK packet from the optimal node, or from one of the sub-optimal nodes. In the instance in which it receives an ACK, node *S* empties the RICN matrix information for that packet's destination, and repeats the same procedure to send a next DATA packet (if any).

If a node does not receive an ACK packet from both of its neighbours, i.e., nodes *A* and *B*, the data packet is assumed to have been lost and node *S* schedules another 7 retransmission retries. If a DATA packet could not be delivered on the MAC layer (within the extra 7 retransmissions) to the optimal or one of the sub-optimal nodes, a failure notification will be sent to source node from MAC layer. That is followed by returning the packet from MAC layer to the routing protocol. In such an instance, the sender node will do a group of procedures. First, it removes the two neighbours from its neighbourhood's location-matrix. Second, it resets the CHT timer, and lastly, it buffers the packet and moves. Node *S* buffers the packet until it finds a suitable next relay-node, or it might discard the packet if TTL of the packet comes to *zero*.

4.5.1.2 Positive and Candidate Neighbour

Any positive neighbour that receives the RTF packet checks if it provides forward progress. If it has no neighbour(s) articulated towards the final destination, then it does not reply to the request of the source node at all, i.e., node *T*. By doing this, two benefits can be attained. First, the sender will recognize that there is a void region in that neighbours' direction, so that the sender will never send packets towards that direction. Second, refraining from sending responses contributes to keeping MANET from becoming congested.

Each other node that has neighbour(s) towards the target, i.e. nodes *A* and *B*, calculates the self φ_i, ϑ_i, and $\varkappa_i$ by using the Equations 4.9, 4.11 and 4.12 respectively. Next, it appends this information to the header of CTF packet that contains other data. Lastly, each node sends the CTF packet within the random time selected from the uniformly interval [0.5 $SIFS * N_{cn}\ \mu s$, 1.5 $SIFS * N_{cn}\ \mu s$] which is defined by the source node.

4.5.1.3 Optimal Neighbour

As a node receives the DATA packet as a response to the sent CTF packet, the node should know that it assigned as the optimal node. In such scenario, the optimal node should wait for the SIFS interval before it sends back an ACK to the sender node. The optimal node transmits the ACK packet without checking the status of the channel, because the channel is assumed to be reserved. Next, it will forward the received

DATA packet for the next relay-node, unless it is the destination node.

4.5.1.4 Sub-optimal Neighbours

When the optimal node fails to receive DATA packet for any reason, the sub-optimal neighbours have the ability to handle the forwarded packet instead of the optimal node because they eavesdropped on the DATA packet when it was sent. As a consequence, the forwarded packet reaches its destination, instead of being dropped. The neighbour knows if it is assigned as a sub-optimal node by the means of the information in the header of DATA packet.

Even though sub-optimal recipients do not respond to DATA packet immediately, they buffer the received packet. Next, they extract the information from DATA packet's header and add it to DATA LisT (DALT) for later use. Every node in MANET should maintain a DALT table. The DALT table is constructed from the IDs of the sender and destination nodes, and the amount of time that the node (as sub-optimal node) will wait before it sends the ACK. Table 4.8 below shows the structure of the DALT table in the nodes *A* and *B* for *i*-received- DATA packets. The first entry in the DALT table of both nodes is the information heard from node *S* in DATA packet.

Table 4.8. The DALT Table of Node B for *i*-Received- DATA Packets

ID_S	ID_D	t_S	t_r	RSN	$WTSA\mu s$
$*1$	2	t_S	t_r	1	10
$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$

$*$*This entry for this example.*

where, ID_S is the sender node identity, ID_D is the destination node identity, t_S is DATA packet sending time, t_r is DATA packet receiving time, RSN is the Reliability Sequence Number, and WTSA is the waiting time to send ACK.

In Table 4.8 above, the waiting time (WTSA) to send the ACK packet from any sub-optimal node can be calculated by using Equation 4.30 below.

$$WTSA = \left(node'sRSNnumber * 1SIFS\right) + 1SIFS\,\mu s \tag{4.30}$$

When the node's WTSA timer reaches *zero*, the ACK packet will be sent back to

source node. The DATA packet will be forwarded for the next relay-node. On the other hand, during the waiting time, the packet and its information in DALT will be discarded if an ACK for the same packet is heard from another neighbour.

One more thing to be mentioned here is that either the optimal or one of the sub-optimal nodes that succeeded in receiving the sent DATA packet will follow the same procedure as the source node to send the packet to the ultimate destination. Simultaneously, the hop counter in the header of DATA packet is increased monotonically.

4.5.1.5 Other Neighbours

The interaction of an RTF-CTF exchange that is followed by the DATA transfer and an ACK silences the neighbours in the vicinity of the sender node *S* and receivers, i.e. nodes *A* and *B*, and thus nodes (*N, O, P, K, L, M, J, I,* and *H*) must wait for the NAV_{ARFT} interval time before they can use the channel.

On the other hand, any neighbour in the vicinity of the sender node of RTF, CTF, DATA, or ACK packets, can benefit from these packets by updating the information in its NLM and DLI matrixes.

4.5.2 Workflow of the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism

This sub-section presents the proposed flowcharts that visualize a series of the workflows. The flowchart shows the procedure of the proposed DRESM algorithms from the perspective of both the sender node and the receiver nodes. There are five separate flowcharts. The first one shows the procedures when a node has a DATA packet to be sent. The second one shows the procedures when a node receives a CTF packet. The third one shows the procedures when a node receives the ACK packet. The fourth one shows the procedures when a node receives a RTF packet. Finally, the fifth one shows the procedures when a node receives a DATA packet. Below is a brief explanation of each flowchart.

4.5.2.1 Procedures Performed by Source (Originator) Node

As shown in Figure 4.7 below a node that has a DATA packet to be sent performs the following procedures;

A. If a node has the destination location in DLI matrix, it executes the DPS scheme to predict the current destination' location, otherwise the node gets the destination' location from location server.
B. Maintains the predicted or the obtained destination' location information in the DLI matrix.
C. Figures out self-distance to destination.
D. Looks up its NLM matrix for any positive neighbours. If the node has positive neighbours,

    a) Senses the channel,

        i. If the medium is free for a DIFS time interval,

            - Decrements DIFS, When counter down of DIFS reaches *zero* the node goes to step F.

        ii. Otherwise,

            - Defers until the end of the current transmission;
            - Waits an extra DIFS interval and selects a random backoff delay; and
            - When the backoff reaches *zero* the node goes to step F.

E. Otherwise,

    a) Sets counterdown timer TTL=30 s;

        i. Buffers the packet and moves; and

            - If TTL counter down reaches zero, discards this packet.
            - Otherwise, the node goes to step number D.

F. If destination is one of the positive neighbours OR has only one positive neighbour,

    a) Sets NAV_{ARTF} for one node,
    b) Increments RF by1;
    c) Sends RTF packet;

d) Sets count down timer to $T_{ACTF} = T_{CTF} + SIFS\ \mu s$ to sreceive CTF packet;

e) Reset countdown timer for next CHT; and

f) Goes to step H.

G. Otherwise,

a) Sets NAV_{ARTF} for maximum three nodes,

b) Increments *RF* by1;

c) Sends RTF packet;

d) Sets count down timer to $T_{ACTF} = [(T_{CTF} * N_{cn}) + (N_{cn} * 1.5 * SIFS)]\ \mu s$ to sreceive CTF packet;

e) Reset countdown timer for next CHT; and

f) Goes to step H.

H. While the timer counts down to *zero*,

a) If the node receives CTF packet, goes to receive CTF procedures.

b) Otherwise, waits.

I. Upon failure to receive any CTF packet, when T_{ACTF} reachs zero, resends RTF packet up to 7 times.

a) If the node receives CTF packet, ceases retransmission and goes to receive CTF procedures.

b) Otherwise, Goes to step F.

4.5.2.2 Procedures Performed when A Node Receive CTF Packet

As shown in Figure 4.8, a node receiving a CTF packet performs the following procedures;

A. If a CTF with the same { ID, sequence number} already had been received,

a) Discards this packet,

B. Otherwise,

a) If from existed neighbour in NLM matrix,

i. Updates the current entry,

b) Otherwise,

i. Creates a new entry for this new neighbour in NLM matrix.

C. If the node did not send a RTF packet for this received CTF packet,

a) Discards this packet,

D. Otherwise,

a) Increments Ncn number by one.

b) If after addition, the Ncn becomes greater than 3 or T_{ACTF} equals *zero*,

i. Discards this packet,

c) Otherwise,

i. Calculates des_D^C, $\Im_C^S$ and $\eth_C^S$ for this neighbour; and

ii. Adds the results to RICN matrix.

E. Copies the , ϑ_C, φ_C, and $\varkappa_C$ from the CTF 's header of this neighbour to RICN matrix.

F. When the node received 3 CTF packets from 3 candidate neighbours, i.e., Ncn equals 3 or T_{ACTF} counter down timer reachs *zero*, runs FLDRE to calculate the candidate neighbours RIN.

G. Adds the calculated RIN values to RICN matrix.

H. Arranges the candidate neighbours in descending order with respect to their RIN values, if two or all of them have same RIN, and then arranges them with respect to their ID numbers.

I. Sets the RSN number to zero for the neighbour with highest RIN value (or highest ID if they have same RIN values), and the other two sub-optimal neighbours with 1 and 2 RSN number with respect to their RIN values (or with respect to their ID numbers).

J. Sets the counter down timer to SIFS.

K. When the SIFS timer reaches *zero*, appends {ID, (x,y) coordinates, RSN} of the optimal and candidate neighbours to the header of the DATA packet.

L. Unicasts the DATA packet to the neighbour with highest RIN.

M. Sets the clock counter-down to T_{AACK} to receive the ACK packet from one of the candidate neighbours.

N. When the T_{AACK} reaches *zero*,

   a) If a node receives ACK,

      i. goes to receive an ACK procedures,

   b) Otherwise,

      i. retransmits the DATA packet up to 7 times

O. During the 7 retransmissions,

   a) If the node receives an ACK packet,

      i. Ceases the retransmission, goes to receive an ACK procedure.

   b) Otherwise,

      i. Goes to buffer and move procedures

4.5.2.3 Procedures Performed when A Node Receive an ACK Packet

As shown in Figure 4.9 below, a node receiving an ACK packet performs the following procedures;

A. If an ACK packet with the same {ID, sequence number} already had been received, then the node discards this packet, otherwise, it updates the information of the sending node in NLM matrix.
B. If the node sent the DATA packet, then the node deletes the information in RICN matrix, otherwise discards the ACK packet.
C. If the node still has more DATA packets to be sent, then goes to have data packet to be sent procedures, otherwise ends the process.

4.5.2.4 Procedures Performed when A Node Receive RTF Packet

As shown in Figure 4.10 below, a node receiving a RTF packet performs the following procedures;

A. If a RTF packet with the same {ID, sequence number} already had been received, then the node discards this packet, otherwise, it updates the information of the sending node in NLM matrix.

B. Looks up self NLM,

a) If the node has positive neighbours in the direction of the destination,

i. Appends self ϑ_C, φ_C, and $\varkappa_C$ to the CTF header;

ii. Selects random time from the interval $[0.5 * SIFS * N_{cn}, 1.5 * SIFS * N_{cn}]$;

iii. Counts down this timer, when the timer reachs zero sends a CTF packet; and

iv. Resets the CHT timer.

b) Otherwise, the node discards this RTF packet

4.5.2.5 Procedures Performed when a Node Receive a DATA Packet

As shown in Figure 4.11 below, a node that receives a DATA packet to be sent performs the following procedures;

A. If a DATA packet with the same {ID, sequence number} already had been received, then the node discards this packet, otherwise, it updates the information of the sending node in NLM matrix.

B. If the node is the final destination,

a) Receive the DATA packet to the next layer;

b) Sets the counter down to the SIFS;

c) When the counter reaches *zero*, sends back an ACK packet; and

d) Goes to step number D.

C. Otherwise,

a) If the node ID listed in the packet header as candidate neighbour,

i. If the node RSN value equals *zero*,

- Receives the data packet;
- Sends back ACK packet after SIFS interval time; and
- Goes to step D.

ii. Otherwise,

- Buffers the packet.
- Updates DALT list.

- Sets counter down to WTSA= Self RSN number × 2SIFS μs .
- If the node hears an ACK from other neighbour,
  - Ceases WTSA timer, deletes the packet's information from the DALT table.
- Otherwise, when the WTSA counter reaches *zero*,
  - Receives the data packet;
  - Sends back ACK packet after SIFS interval time; and
  - Goes to step D.

b) Otherwise,

i. Discards the packet

ii. Ends the process

D. Checks Hc,

a) If Hc equal 7,

i. Discard the packet

b) Otherwise,

i. Goes to has DATA packet to be sent procedures.

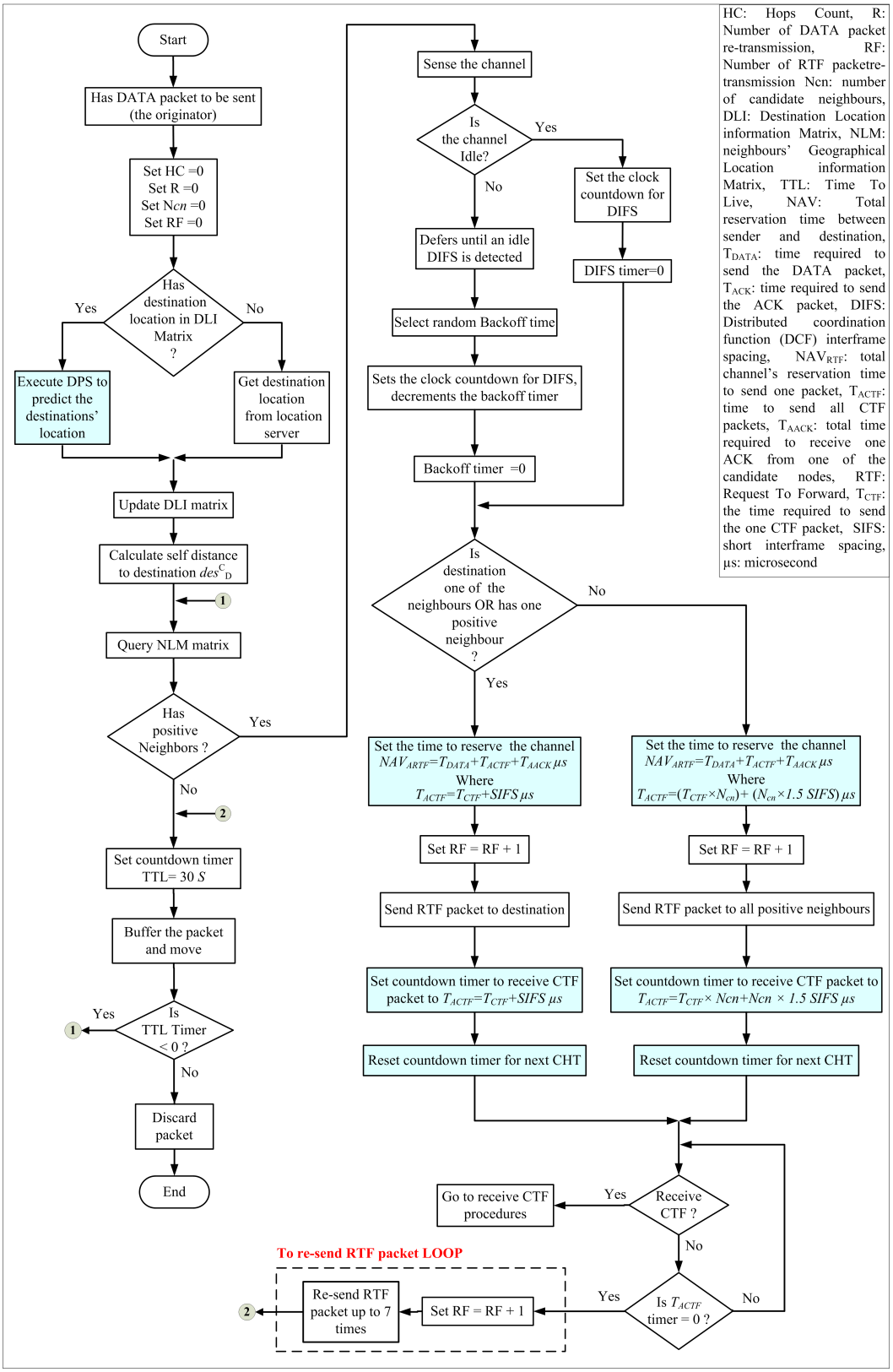


Figure 4.7. Flowchart of DRESM Executed by A Node when it has DATA Packet to be Forwarded, the Shaded Blocks are the Contribution of the Proposed Protocol

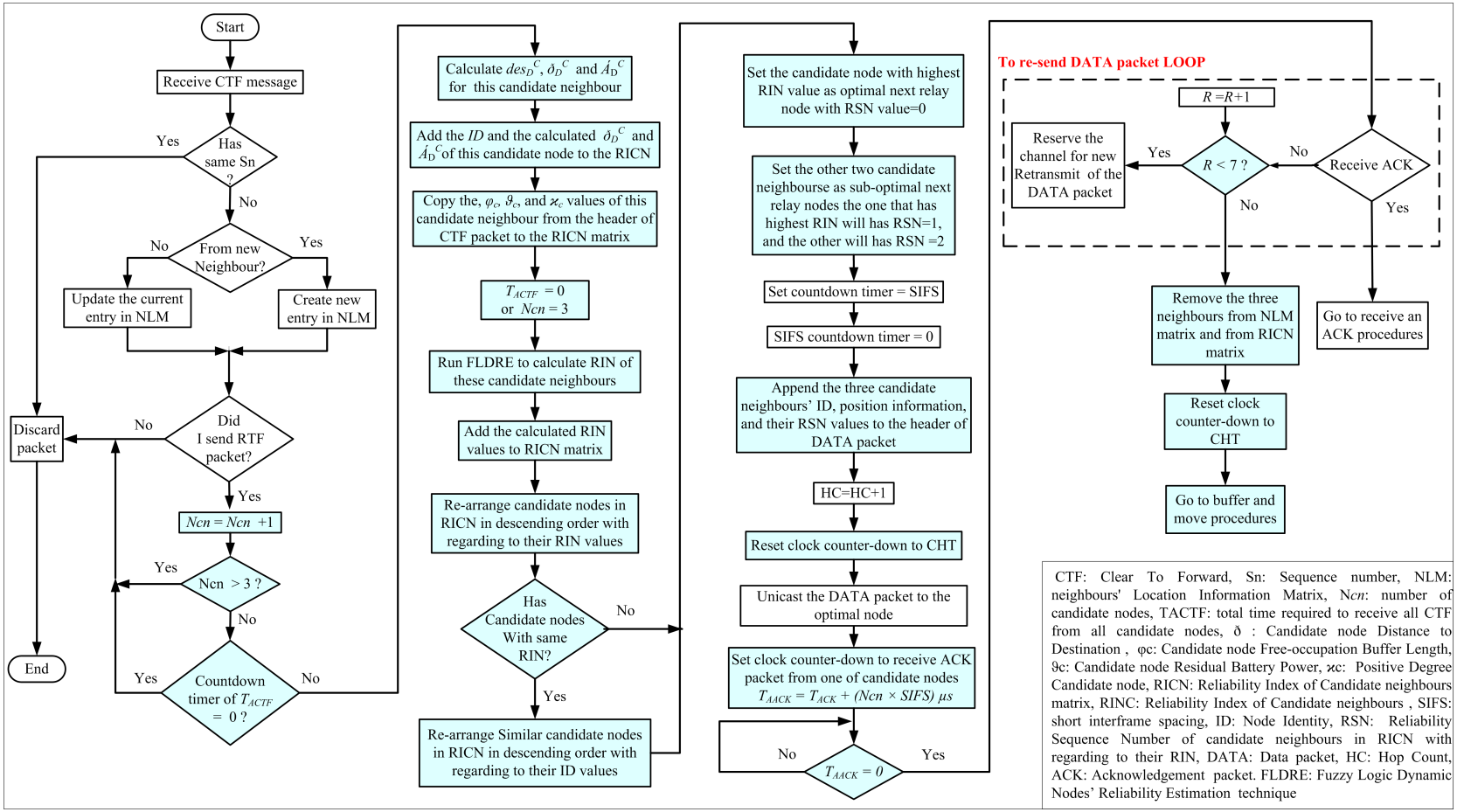
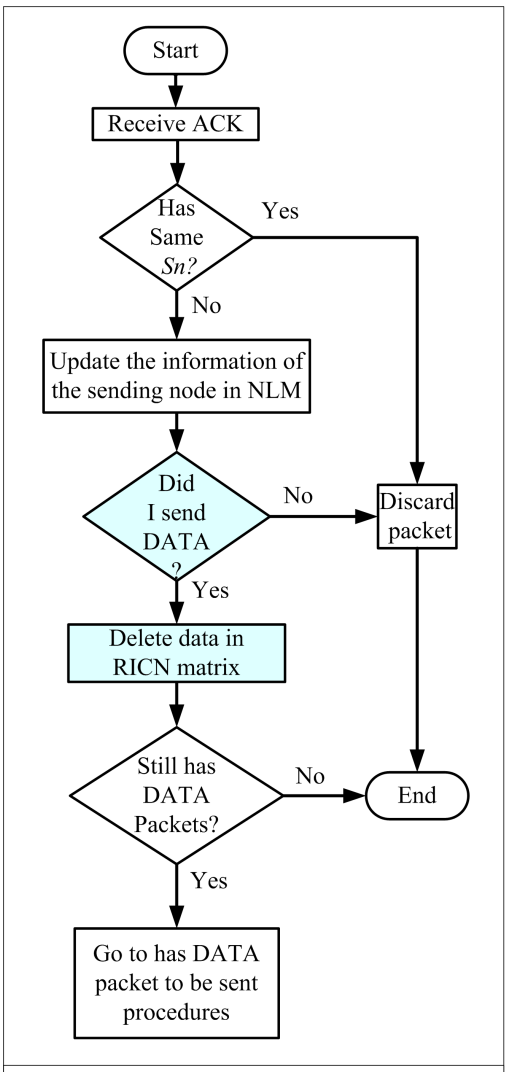
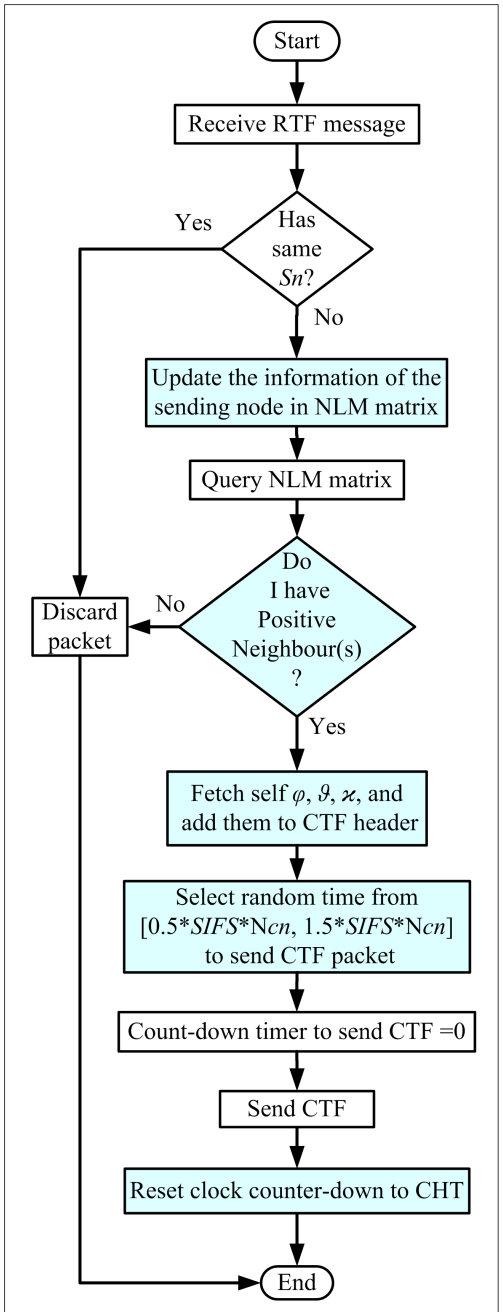


Figure 4.8. Flowchart of DRESM Executed by A Node After it Received CTF Until Sending DATA Packet, the Shaded Blocks are the Contribution of the Proposed Protocol



T_{AACK}: total time required to receive one ACK from one of the candidate nodes, T_{ACK}: Time Required to send ACK packet, Ncn: number of candidate nodes, SIFS: short interframe spacing, µs: microsecond, ACK: Acknowledgement message, R: retransmission number, *Sn*: Sequence number, DATA: Data packet

Figure 4.9. Flowchart of DRESM Executed by A Node after it Sent DATA Packet, and Waiting to Receive ACK Packet, the Shaded Blocks are the Contribution of the Proposed Protocol



RTF: Request To Forward, Sn: Sequence number, NLM: neighbours' Location Information Matrix, *φc*: Candidate node Free-occupation Buffer Length, *ϑc*: Candidate node Residual Battery Power, *ϰc*: Positive Degree Candidate node CTF: Clear To Forward, SIFS: short interframe spacing

Figure 4.10. Flowchart of the DRESM Executed when A Node Receives RTF Messages, and Send Back CTF Packet, the Shaded Blocks are the Contribution of the Proposed Protocol

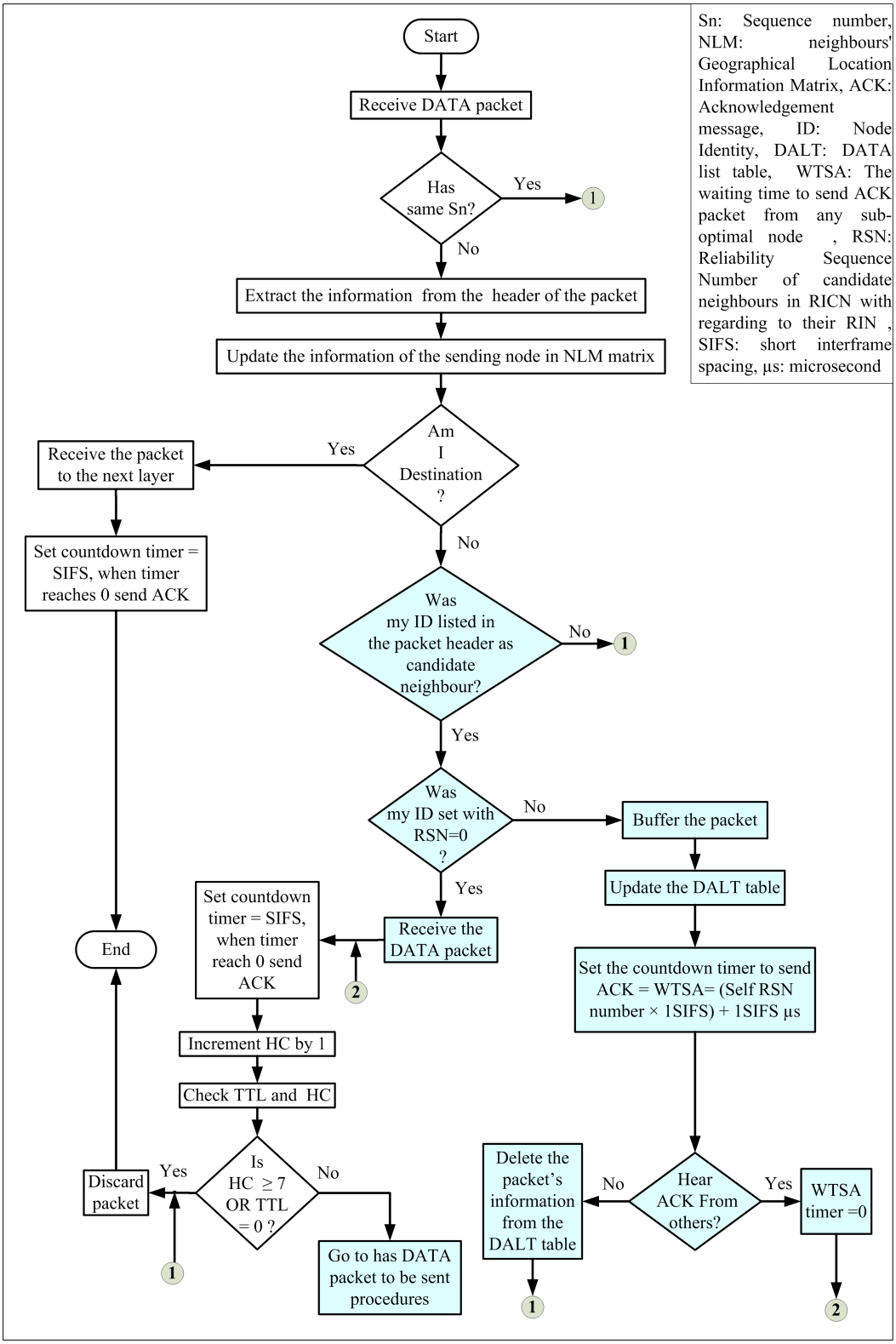


Figure 4.11. Flowchart of DRESM Executed when A Node Receives DATA Packet, and Send Back ACK Packet, the Shaded Blocks are the Contribution of the Proposed Protocol

4.6 Implementation of the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism

This section focuses on the implementation of Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM) in a simulated environment. In this research work, Ns2 simulation is considered in the implementation the proposed DRESM mechanisms and to validate the research assumptions and propositions. The same steps that are performed to implement the Dynamic Beaconing-Update Mechanism (DBUM) in Chapter 3, Section 3.6 will be performed to implement DRESM in this section flow. Thus, to prevent duplication, the same procedures that are required to implement DRESM will be not discussed fully.

4.6.1 Simulation Environment

This section presents the simulation environment to measure and evaluate the performance of the proposed DRESM under various network settings. The simulation setup includes the traffic pattern generation, mobility model, and simulation input parameters. This section follows the same procedures as in Chapter 3, Section 3.6.1.

4.6.2 Implementation and Verification of GPSR and POR

This section follows the same procedures as in Chapter 3, Sub-section 3.6.2.

4.6.3 Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism Validation

This section presents the validation of the proposed Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism DRESM. This section follows the same procedures executed in Chapter 3 Section 3.6.3.

4.6.3.1 DRESM Validation in Static Environment

This section applies the same procedures as in Chapter 3, Sub-section 3.6.3.1 to validate DRESM, but with a new addition. After implementing GPSR-DRESM on the Null MAC and atop the Ns2 802.11 MAC DCF layer (traditional version), the GPSR-

DRESM is again implemented atop the Ns2 802.11 MAC DCF layer (with the proposed improvements). This is done to verify GPSR-DRESM's response to MAC transmit failure callbacks in a different setting. In the three scenarios the network area is set to $2000\,m \times 450\,m$ and 25, 50, 75, 100, 150, 175 and 200 participants, meanwhile other parameters of the network are the same as described in Chapter 3, Table3.6 except the nodes speeds is set to zero.

Figure 4.12 below shows the simulation results of the three scenarios. The figure shows that, after running the GPSR-DRESM several time over the Null MAC, the success rate achieved 100% delivery. This means that the GPSR-DRESM algorithm can make correct forwarding decisions. The same figure shows that, because of running the GPSR-DRESM atop the Ns2 802.11 MAC DCF layer (traditional version), when the number of nodes is 100 nodes (the median number of nodes in this experiment), the packet delivery rate decreases to 91.592%. Also, when GPSR-DRESM is again implemented atop the Ns2 802.11 MAC DCF layer (with the proposed improvements), when the number of nodes is 100 nodes (the median number of nodes in this experiment), the packet delivery rate decreases to 94.541%. The decrement in the delivery rate in the last two implementations is attributed to the contention at the interfaces of the nodes to access the radio channel. The success and effectiveness of GSAR is attributed mainly to the improvement done on MAC DCF layer. Hence, based on the simulation results, the conclusion can be made that the GPSR-DRESM can make correct forwarding decisions.

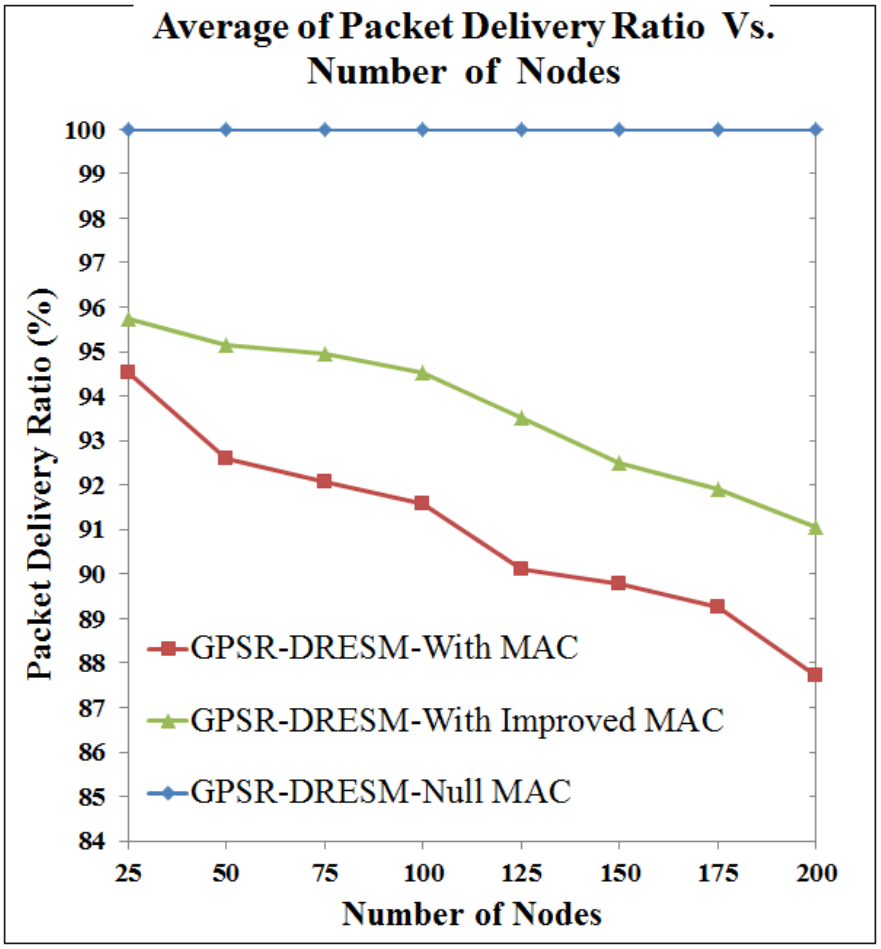


Figure 4.12. Average Packet Delivery Ratio via Number of Nodes

4.6.3.2 DRESM Validation in Mobile Environment (Simple Case)

The purpose of this section is to validate the correctness of the proposed DRESM algorithm. To accomplish this, the researcher monitored the ability of a node using the DRESM algorithm to avoid encountering hole problem and to select the next relay-node based on the proposed routing metrics. To accomplish this, a simulation scenario is build with a communication geographic hole in a simple network topology created in Ns2 as suggested by [122, 155]. In this experiment, the mathematical results of calculating the neighbours' reliability index, based on Equation 4.23, are compared with the simulation output result.

A network topology is created using Ns2.33 as shown in Figure 4.13. In this scenario, 40 participants are configured in a $800\,m \times 450\,m$ rectangle simulation area. The argument for using this simulation dimension is attributed to the number of nodes required to maintain the ratio of 1 node /9000 m^2 as suggested by [44, 45]. The other parame-

ters of the network are the same as described in Chapter 3, Table 3.6. All nodes apply the GPSR-DRESM routing protocol. Because the steady state is disabled in this research, the researcher starts collecting the results as the simulation starts, hence, the data flows starts from the beginning of the simulation time.

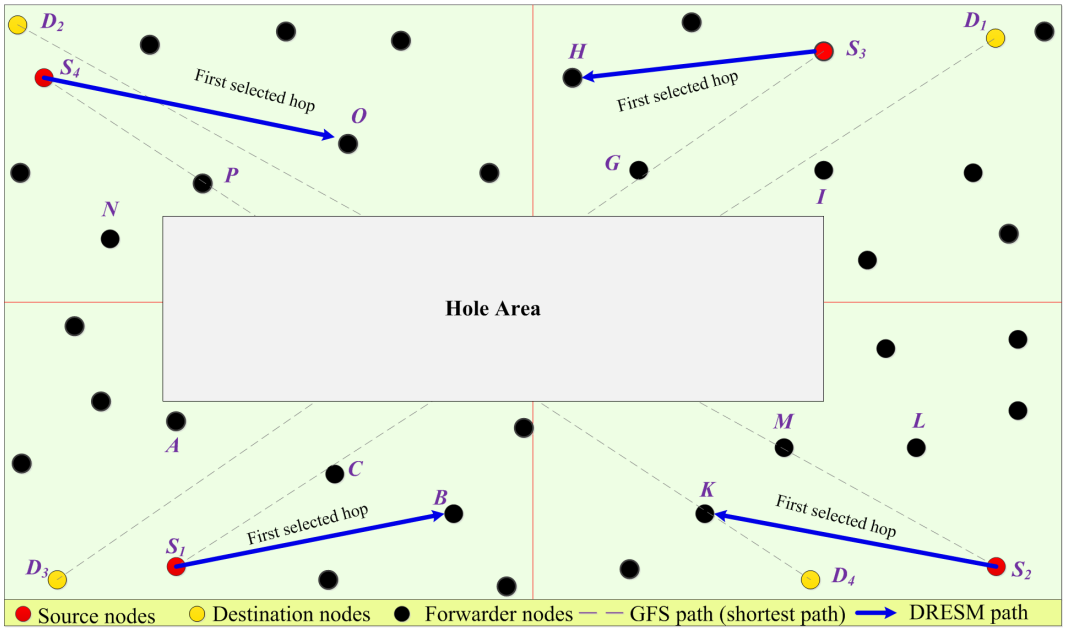


Figure 4.13. Network Model for Performance Validation of DRESM

The Analytical Results

In this scenario, the researcher configured four stationary pairs of nodes that wish to communicate with each other. At the sending time t, the specification of each pair as $S = (ID_S, x_t^S, x_t^S, v_t^S, \theta_t^S)$, and $D = (ID_D, x_t^D, x_t^D, v_t^D, \theta_t^D)$ are: [$S_1 = (1, -270, -200, 30$ $, 45^\circ)$, and $D_1 = (2, 350, 200, 20, 190^\circ)$], [$S_2 = (3, 350, -200, 25, 110^\circ)$, and $D_2 = (4, -390, 210, 22, 300^\circ)$], [$S_3 = (5, 220, 190, 35, 200^\circ)$, and $D_3 = (6, -360, -210, 25,$ $40^\circ)$], and [$S4 = (7, -370, 170, 28, 290^\circ)$, and $D_4 = (8, 210, -210, 20, 300^\circ)$.

At a very early stage, each source node obtains the destination information and updates its *DLI* matrix. Next, it starts to look up its NLM to check if valid positive neighbours for targeted destination are available. After a source node becomes sure that targeted destination is not one of its neighbours, and the channel is idle, it sends RTF packet to all of its positive neighbours.

Once the source node receives CTF packets from its candidate nodes, it updates the

candidate neighbours' information in its NLM. In this scenario, the candidate neighbours are; A and B for node S_1, K and L for node S_2, H and I for node S_3 and N and O for node S_4. Nodes *C, M, G* and *P* are not considered to be a candidate neighbours because they do not have positive neighbours in the direction of targeted destination (i.e., they are geographical hole nodes).

A part of CTF packets which is $(ID_c, x_c, y_c, v_c, \theta, \varphi_c, \vartheta_c, \varkappa_c)$ from nodes B and C to node S_1 hold the following information : $A = (9, -270, -90, 22, 120^\circ, 0.6, 0.75, 0.7)$, $B = (10, -60, -160, 30, 20^\circ, 0.8, 0.8, 0.9)$ respectively. A part of CTF packets which is $(ID_c, x_c, y_c, v_c, \theta, \varphi_c, \vartheta_c, \varkappa_c)$ from nodes K and L to node S_2 hold the following information : $K = (12, 130, -160, 28, 120^\circ, 0.7, 0.7, 0.65)$, $L = (13, 290, -110, 32, 80^\circ, 0.8, 0.7, 0.85)$ respectively. A part of CTF packets which is $(ID_c, x_c, y_c, v_c, \theta, \varphi_c, \vartheta_c, \varkappa_c)$ from nodes H and I to node S_3 hold the following information : $H = (15, 30, 170, 15, 210^\circ, 0.7, 0.75, 0.8)$, $I = (16, 220, 100, 18, 230^\circ, 0.75, 0.85, 0.9)$ respectively. A part of CTF packets which is $(ID_c, x_c, y_c, v_c, \theta, \varphi_c, \vartheta_c, \varkappa_c)$ from nodes N and O to node S_4 hold the following information : $N = (18, -320, 50, 20, 330^\circ, 0.9, 0.75, 0.6)$, $O = (19, -140, 120, 28, 350^\circ, 0.95, 0.8, 0.65)$ respectively.

Now, as shown in Table 4.9 below, each source node calculates $\eth_D^C$, and $\Im_C^S$ by using the Equations 4.6 and 4.7.

Table 4.9. The $\eth_C^S$ and $\Im_C^S$ Calculation of Candidate Nodes for each Source Node

Source	Node	des_D^S	des_D^C	$\eth_D^C$	des_C^S	RV_C^S	RLT_C^S	$\Im_C^S$
S_1	***A***	737.8	684.4	0.212	110	41.5	3.37	0.013
	B		545.6	0.768	213.7	58.5	0.62	0.0021
S_2	***K***	845.9	638.2	0.828	223.6	52.7	0.5	0.002
	L		751.5	0.376	108.1	55	2.58	0.0103
S_3	***H***	704.5	544.5	0.64	191	49.8	1.18	0.004
	I		657.6	0.188	90	51.3	3.11	0.0124
S_4	***N***	693.3	590.3	0.412	130	45.1	2.66	0.0106
	O		481.0	0.848	235.3	41.7	0.352	0.00141

Based on the information in CTF packets that were received from the candidate nodes

and the calculations in Table 4.9, each source node starts to maintain its RICN matrix. Next, each source node figures out the RIN of each candidate neighbour by using Equation 4.23 and adds it to RICN matrix.

The Final formation of RICN Matrices in source nodes S_1, S_2, S_3, and S_4 are stated in Equations 4.31, 4.32, 4.33, and 4.34 below.

$$RICN_{S_1} = \begin{bmatrix} 10 & 0.768 & 0.0021 & 0.8 & 0.8 & 0.9 & 0.75 & 0 \\ 9 & 0.212 & 0.013 & 0.6 & 0.75 & 0.7 & 0.475 & 1 \end{bmatrix} \tag{4.31}$$

$$RICN_{S_2} = \begin{bmatrix} 13 & 0.376 & 0.0103 & 0.8 & 0.7 & 0.85 & 0.666 & 0 \\ 12 & 0.828 & 0.002 & 0.7 & 0.7 & 0.65 & 0.648 & 1 \end{bmatrix} \tag{4.32}$$

$$RICN_{S_3} = \begin{bmatrix} 15 & 0.64 & 0.004 & 0.7 & 0.75 & 0.8 & 0.703 & 0 \\ 16 & 0.188 & 0.0124 & 0.75 & 0.85 & 0.9 & 0.625 & 1 \end{bmatrix} \tag{4.33}$$

$$RICN_{S_4} = \begin{bmatrix} 19 & 0.848 & 0.00141 & 0.95 & 0.8 & 0.65 & 0.75 & 0 \\ 18 & 0.412 & 0.0106 & 0.9 & 0.75 & 0.6 & 0.628 & 1 \end{bmatrix} \tag{4.34}$$

Now each source node copes *IDs*, RIN and RSN from RICN matrix and appends them to DATA packet header. Next, it sets the hop counter to 1, and unicasts the packet to the optimal node (e.g., node B with ID number 10 for source node S_1) in IP layer. Once an optimal node (i.e., nodes B, L, H and O) receives the data packet it will perform the same procedures done by source node until the routed packet reaches its final destination.

The Simulation Results

To be sure of the correctness of DRESM functionality, the researcher tests to determine if the sent packet is forwarded using the optimal path. Each optimal path in this scenario should start with the optimal next relay-node which found by the mathmatical calculation. To monitor all optimal paths construction, at the header of each DATA

packet, a recorder is initiated to list the *ID*s of the nodes that the routed packet passes through from each source node to the targeted destination.

At the targeted destination node of each source node, i.e., nodes D_1, D_2, D_3, or D_4, the reference recorder is checked to figure out the path traversed by the packet. The simulation results depicted in Figure 4.14 below show the data packet trip recorder, at the header of each routed packet from a source to its targeted destination. The researcher observes that the simulation results in Figure 4.14 are matching with the analytical view in Equations 4.31, 4.32, 4.33, and 4.34 above. Hence, because the simulation and analytical results are symmetrical, then the model of DRESM is valid because it can achieve its design objective.

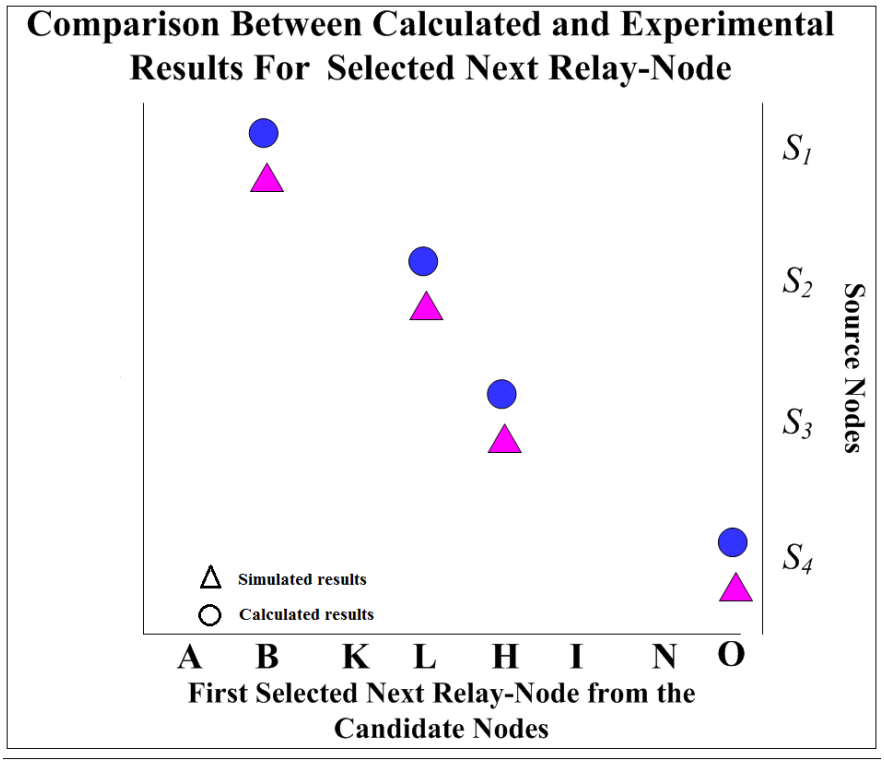


Figure 4.14. Comparison of Calculated and Simulated Results to Find the First Relay-node A Routed DATA Packet Traverse in its Trip from each Source Node to its Targeted Destination

4.7 Implementation of DRESM to Assign the Experience Fuzzy Processing Delay in FLDRE Technique

In DRESM, the nodal processing delay includes the time required to check the packet's header, the time required to look up the neighbours-matrix to find the neighbours in the direction of destination and the time required to find the optimal next relay-node based

on multi-routing metrics to direct the packet. The time needed to check for bit-level errors might also be considered to be part of the processing delay [171]. As stated in [136], total delay experienced by a packet to reach its final target is the sum of final experienced delay at each node.

This research uses fuzzy logic in the proposed FLDRE model. FLDRE figure out the reliability index of maximum three active neighbours based on multi-routing metrics. Thus, fuzzy-delay in FLDRE is a part of the processing delay experienced by a packet per-hop until it reaches its targeted destination. This research uses the same method suggested in [159, 160] to measure the FLDRE's fuzzy-delay. In the adopted method, a sender node that wishes to send DATA packet starts a counter when it is about to perform fuzzy algorithm to determine the reliability index of maximum three neighbours. After finishing executing the FLDRE algorithm, the sender node stops the counter and records the counter result in the timestamp say "FuzzyMonitor" at the header of the DATA packet. This FuzzyMonitor contains per-process fuzzy-delay counters.

To determine if the fuzzy-delay is harmful to the performance of the proposed routing protocol, the researcher conducted a comparative study. In this comparison, the processing delay experienced by a packet per-hop to execute FLDRE algorithm was computed. At the destination side, to find the average processing delay, the aggregated fuzzy-delay was divided by the total number of executing FLDRE algorithm. The result of the experienced fuzzy-delay is compared to the processing delay experienced by the traditional GPSR protocol when it executed to find the optimal next relay-node based on the geometric calculation.

The researchers in [172] argued that a node's processing delay is affected by the number of its neighbours within its transmission range. Based on this fact, to measure effectively the intended processing delay of both compared protocols, the researcher proposed a scenario of a 25, 50, 75, 100, 125, 150, 175 and 200 nodes. In this experiment, the speed and traffic sources are fixed to 20 $m\backslash s$ and 5 sources respectively. The other parameters of the network are set as described in Chapter 3, Table3.6.

Figure 4.15 below shows the average of the targeted processing delay experienced by GPSR and GPSR-DRESM protocols, as a function of number of nodes. The simula-

tion results revealed that as the number of nodes increases, the fuzzy processing delay in GPSR-DRESM slightly increases and becomes static when the number of participant exceeds 110 nodes. This is because in the specification design of DRESM, a node executing FLDRE algorithm can receive up to maximum three Clear To Forward (CTF) packets. This means that a node will execute the FLDRE algorithm to find the reliability index for only three of its neighbours.

The simulation results in Figure 4.15 revealed that as the number of nodes increases, the processing delay in GPSR increases as well. This is because as the number of nodes increases the number of a node's active neighbours increases as well. As a consequence, the time that is needed by a node to compare the active neighbours' distance to a specific destination in order to find the optimal next relay-node in terms of geometric calculation increases. In the specification design of GPSR, the maximum compared number of the active neighbours is not limited. This is why GPSR achieved lower processing delay when the number of the active neighbours below or equal 6 neighbours and experiences higher delay when the number of the active neighbours exceeds 6 neighbours compared to GPSR-DRESM.

To clarify these results, Table 4.10 below shows the average number of a node's active neighbours in the 8^{th} proposed scenarios during the simulation period. In GPSR, these active neighbours represent the number of neighbours a node considers to compare their distance to ultimate destination using geometric calculation. In GPSR-DRESM, theses active neighbours represent the number of neighbours a node considers to compare their reliability index using FLDRE algorithm.

To conclude, the simulation results in Figure 4.15 show that GPSR-DRESM increases the processing delay (fuzzy-delay) rate by about 0.13% when the number of nodes less than 110 nodes over GPSR. Also, GPSR-DRESM reduces the processing delay (fuzzy-delay) rate by about 0.26% when the number of nodes greater than 110 nodes over GPSR. Thus, the fuzzy-delay is an insignificant delay in effecting any used delay-sensitive or tolerant application in MANET.

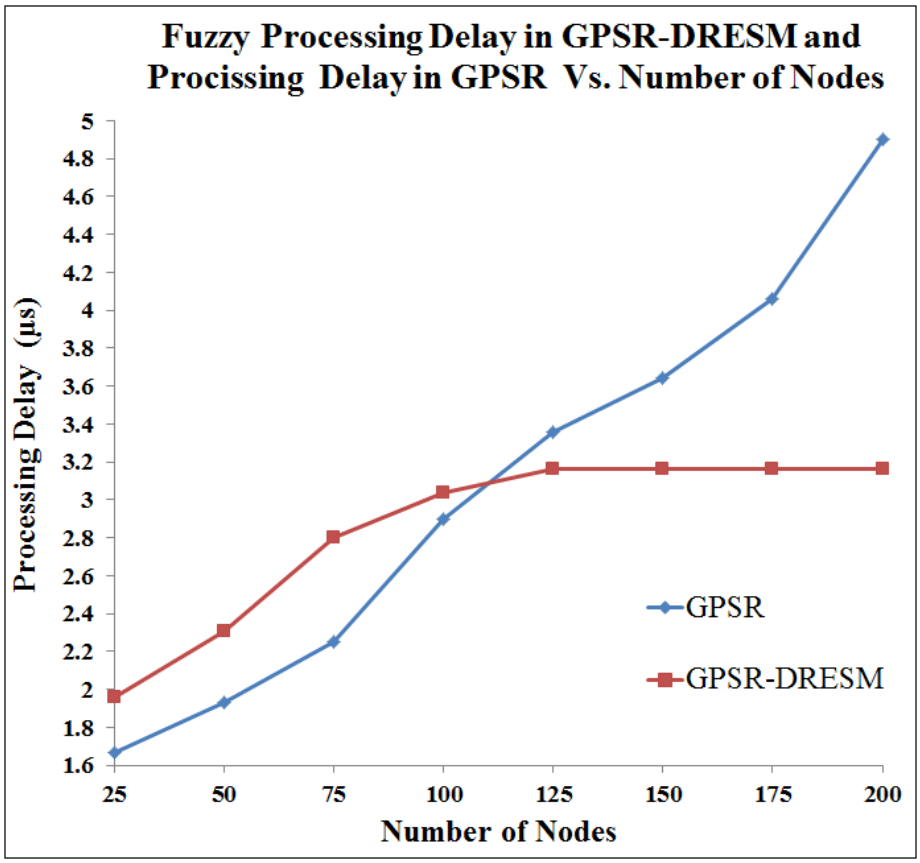


Figure 4.15. The FLDRE Fuzzy-Delay Compared to GPSR Processing Delay

Table 4.10. The Average Number of Active and Compared Neighbours in both GPSR and GPSR-DRESM

No. of Nodes	Average Number of A node's Active Neighbours	Average Number of Compared Neighbours in:	
		GPSR	GPSR-DRESM
25	3	3	1
50	4	4	1.5
75	4.5	4.5	2
100	6	6	2.5
125	6.5	6.5	3
150	7	7	3
175	7.5	7.5	3
200	9	9	3

4.8 Summary of Chapter

This chapter introduced the model of the second proposed mechanism. This chapter presents the design of an alternative prioritization and selection process, termed as DRESM. The goal of the design is to develop prioritization and selection processes that are primarily responsible for reliable and fair traffic load distribution. In order to achieve the above goal, DRESM encompasses two main parts:

i. The design of the Status Information Distribution and Outgoing Traffic Control Management IDOTM; and
ii. The design of the Fuzzy Logic Dynamic Reliability Estimation Technique FLDRE.

The functionality from the perspective of sender, positive and candidate neighbours is explained in detail. To ensure the smooth transition from the modeling to the implementation, the flowchart stages were shown and explained. The proposed mechanism is validated inside the Ns2 by observing the path selection process between communicating nodes. The validation is accomplished by examining the proposed mechanism in a proposed scenario. The results of the simulation prove the efficiency of the proposed mechanism.

The next chapter establishes the evaluation of the proposed mechanisms as they are embedded in the proposed Greedy StandAlone Routing protocol (GSAR). The evaluation aims to judge GSAR's performance compared with selected most popular and recent mechanisms in the research area.

CHAPTER FIVE

GREEDY STANDALONE POSITION-BASED ROUT ING PROTOCOL IN MANET

5.1 Overview

The aim of this chapter is to analyze and evaluate the performance of the Greedy Standalone Routing (GSAR) protocol versus other position-based routing protocols. To achieve this, several simulations were conducted. The remainder of this chapter is structured as follows: Section 5.2 shows how the Dynamic Beaconing Update Mechanism (DBUM) and the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM) are integrated to shape the intended routing protocol GSAR. Section 5.3 reports the simulations conducted to evaluate GSAR. Section 5.4 draws conclusions about the new features of GSAR based on the simulation results. Sections 5.5 and 5.6 present the advantages of DBUM and DRESM based on the simulation results. Finally, Section 5.7 summarizes the chapter.

5.2 Greedy Standalone Routing GSAR Protocol Design

This section presents the full Greedy StandAlone Routing (GSAR) protocol. In the proposed design, two complementary mechanisms are integrated in the traditional GFS strategy to shape the new routing protocol GSAR. These mechanisms are the Dynamic Beaconing Update Mechanism (DBUM) and the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM).

GSAR is a multi-metrics routing protocol. The main purpose of the GSAR protocol is to discover and establish a reliable route between communicating nodes. Each member in this reliable route is selected based on its reliability index. This reliability index is constructed based on five proposed routing metrics. The proposed routing metrics give full and detailed information of the intermediate nodes. In GSAR, the two new mechanisms are proposed to gather the information about these five routing metrics, to make the forwarding decision and to control the outgoing traffic. To accomplish this, the two mechanisms, DBUM and DRESM, should cooperate fully. A schematic

illustration of GSAR flow diagram is shown in Figure 5.1 below.

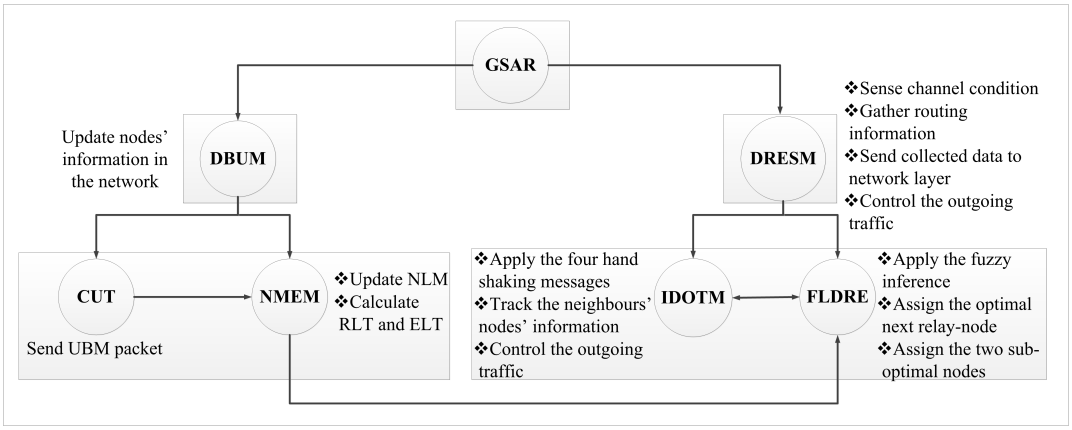


Figure 5.1. Overall Flow Diagram of the proposed Greedy Standalone Routing Protocol GSAR

The DBUM provides fresh information of the nodes' status in the network. This is accomplished by using DBUM's techniques, which are the Compulsory Update Technique (CUT), and Neighbourhood Matrix Entries Management (NMEM) technique. CUT emits an urgent beacon message (UBM) based on the change of a node's mobility features. The broadcasted UBM holds sender's information which are: address (ID), location and velocity. Each node maintains a neighbourhood's location-matrix (NLM) by the means of using NMEM to store information about the nodes within its transmission range. With NMEM, upon receiving a UBM packet, each node updates its matrix with new node information and calculates the Residual Links Lifetime (RLT) and the entry lifetime (ELT) for the sender node (the one that sent the UBM packet). The NLM matrix is checked periodically by the node to update (add/delete) it. Therefore, the source\relay nodes can easily find the IDs, locations, velocity and RLT for their neighbours by the means of using their NLMs. When a node has data packet to be sent for a destination, these information are then used by FLDRE technique in DRESM mechanism to make the forwarding decision.

The DRESM provides fresh information about the network condition, collects fresh information about nodes in the network and control the outgoing traffic. This is accomplished by using DRESM's techniques, which are the Status Information Distribution and Outgoing Traffic Control Management (IDOTM) and the Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE) technique.

In IDOTM the four hand shaking messages were modified to give the source/relay MAC layer the ability to track the neighbours' nodes' information and to control the outgoing traffic. The IDOTM technique allocates and distributes a part the participating nodes' status information. The collected information at the header of Request To Forward (RTF) packet are the node's buffer free size (φ_C); the reminder battery power as a percentage (ϑ_C) and the number of candidate node's positive degree ($\varkappa_C$). As the MAC layer senses the required routing metrics, the collected data are then forwarded to the network layer to be used by FLDRE technique.

At the network layer, a sender node uses fuzzy inference system to read all fetched factors from MAC layer. Next, the sender node estimates the reliability level of its candidate neighbours by using the FLDRE technique. The first three respondents from the positive neighbours are considered to be the candidate nodes and ordered with respect to their reliability level; the highest is chosen for communication as optimal next relay-node. In case the optimal node fails to send the forwarded packet to the next relay-node (for any reason), one of the sub-optimal nodes will act as the next relay-node.

The packet is transmitted as unicast in the IP layer, and multiple receptions is achieved using MAC interception (all nodes within the coverage of the sender would receive the signal). In this way, GSAR and by using IDOTM make full utilization of the collision avoidance supported by 802.11 DFC. While on the receiver side, based on the modification of the MAC layer address filter: even when the data packet's next hop is not the optimal or sub-optimal nodes, the receiver delivered the data packet to update the sender information in the receiver neighbours-matrix. Otherwise, the optimal or sub-optimal nodes delivered the data packet to the upper layer to process the data packet as required.

5.3 Simulation Results and Evaluations

To measure the success in meeting the design goals for GSAR, it is critical to choose an appropriate scenario to evaluate effectively its performance. There are various important parameters of a simulation scenario such as the number of nodes, the node

velocity, node pause time, the mobility model, transmission range of the node, the data traffic generation, and width and height of the simulation area [116]. As discussed earlier in Chapter 2, the nodes mobility causes frequent topology changes that have considerable impact on the performance of a routing protocol and overall network performance. Similarly, changes in network density and traffic density affecting the performance of a protocol and overall network performance. Therefore, for the purposes of diversity in analysis and comparison of various performance aspects of GSAR protocol, the simulations are conducted using three different scenarios based on the discussion in Section 3.6.1.3. The simulation parameters and network setting were kept as previously mentioned in Chapter 3 Table 3.6. These scenarios are:

i. The first scenario discusses the effect of varying participating nodes speed to investigate the performance of GSAR. In this simulation, 100 nodes are deployed and the number of sources is fixed to 5. The nodes speed ranging to 5, 10, 15, 20, 25, 30, 35, and 40 *m\s*, i.e., (18 km\h), (36 km\h), (54 km\h), (72 km\h), (90 km\h), (108 km\h), (126 km\h) and (144 km\h).
ii. The second scenario discusses the effect of varying numbers of nodes to investigate the scalability and robustness of GSAR. In this simulation, the speed and traffic sources are fixed to 20 *m\s* (72 km\h) and 5 sources respectively, and the deployed number of nodes varied to 25, 50, 75, 100, 125, 150, 175, and 200 nodes.
iii. The third scenario discusses the effect of varying number of traffic sources to investigate the performance of GSAR. In this simulation, 100 nodes are deployed; nodes' speed is fixed to 20 *m\s* and varied communication patterns corresponding to 5, 10, 15, 20 and 25 source-destination pairs.

In the simulation scenarios, the source-destination pairs are selected randomly. There are no obstacles and so nodes can always communicate within their transmission range. The reason is to show clearly the deficiencies of using other position-based routing protocols compared with the effectiveness of GSAR.

The researcher selected two position-based routing protocols to perform the intended performance comparison. The first selected protocol is the Greedy Perimeter Stateless Routing Protocol (GPSR) [44]. The second selected protocol is the Position-based

Opportunistic Routing Protocol (POR) [45].

To evaluate the performance of the compared routing protocols, six routing performance evaluation metrics are selected as suggested in [116, 157, 158]. These metrics are not the only possible metrics for use in analyzing a routing protocol for MANET. The six metrics were chosen by selecting those that concisely captured salient characteristics of GSAR.

The researcher calculates the average of the collected results from the simulation comparison with respect to a 95% confidence interval. The formulas to calculate each metric and how to calculate the 95% confidence interval are listed in Appendix *D*. These performance metrics are:

i. Packet Delivery Ratio (PDR);
ii. Control Overhead (COH);
iii. End-to-End Delay (EED);
iv. Routing Path Stretch (RPS);
v. Inconsistency of Neighbourhood Matrix (INM); and
vi. Number of Hole Problem Occurrence (NHPO).

The following sub-sections discuss the selected performance evaluation metrics in the three scenarios.

5.3.1 Simulation1: Impact of Varying Node Speed

This section evaluates the effects of node speed on the performance of GSAR, GPSR and POR routing protocols. In the following simulation, the speed of the nodes is the only parameter that controls network stability. This simulation examines eight scenarios by varying the speed of the nodes. Table 5.1 below shows the parameters that are used in this simulation.

Table 5.1. Varying Node Speed

Parameter	Value
No. of nodes	100
Terain size	2000 $m \times 450$ m
Node speed	5, 10, 15 20, 25, 30, 35, 40 m/s
simulation time	900 *sec*
Traffic type	CBR
Packet rate	2 *Kbps*
Packet size	256 bytes
Number of data traffic sources	5
Transmission range	250 m
Movement model	Modyfied Random Waypoint
MAC layer protocol	IEEE 802.11 DCF
Radio propagation model	Two Ray Ground Model
The other parameters are set as the same as in Table 3.6	

5.3.1.1 Packet Delivery Ratio Vs. Node Speed

Figure 5.2 shows the average packet delivery ratio (PDR) achieved by GPSR, POR and GSAR protocols with respect to node speed. A better protocol is the one that has more PDR. This simulation shows that GPSR and POR protocols perform worse than does GSAR. Under all mobility conditions, more than 97.93% data packets of GSAR can be delivered to specified destinations, which is 14.32% higher than GPSR's 85.65% and 10.31% higher than POR's 88.77% .

All investigated protocols delivered a high percentage of the sent packets with low node mobility (below 10 $m\backslash s$). However, Figure 5.2 shows that with increasing node speed, the performance of GPSR and POR drops sharply, while the PDR of GSAR degrades gracefully (when the maximum speed exceeds 20m/s). This is because the increment in node speed results in fast topology changes that result in increasing the stale information in a node's neighbours-list and increase the probability of link break problem. To track the topology changes, GPSR and POR use beaconing with fixed interval time (FBPIT) that cannot track the fast topology changes. Selecting one of these neighbours with stale information as the next relay-node results in sending the data packet to an inaccurate position that inevitably decreases the PDR at the desired destination. Also, because GPSR and POR are single-routing objective protocols, both

protocols do not consider link break problem when making the forwarding decision; this decreases the performance of GPSR and POR protocols in terms of PDR.

Figure 5.2 highlights the effectiveness of the GSAR's new features (i.e., the two new mechanisms). Even when the maximum speed increases to 40 *m/s*, GSAR still enables nearly 96.42% of the packets to reach the targeted destination. This is because the Dynamic Beaconing Update Mechanism (DBUM) in GSAR maintains accurate information about the neighbours in a node's NLM. This improvement can be attributed to several reasons: The first is due to using Compulsory Update Technique (CUT) that tune the beacon sending frequency with respect to nodes speeds. Second, the Neighbourhood Matrix Entries Management (NMEM) technique updates dynamically the entries lifetime (ELT) in a nodes' neighbourhood's location-matrix (NLM) based on the neighbours' Residual Links Lifetime (RLT). Third, the GSAR protocol utilizes the Destination Prediction Scheme (DPS), which is used in IDOTM technique to increase the accuracy of destination location information. Finally, GSAR considers the link lifetime between nodes as one of its multi-routing metrics to make its forwarding decision, that is the Residual Links Lifetime RLT metric. Thus, the data packets are delivered reliably at the destination side.

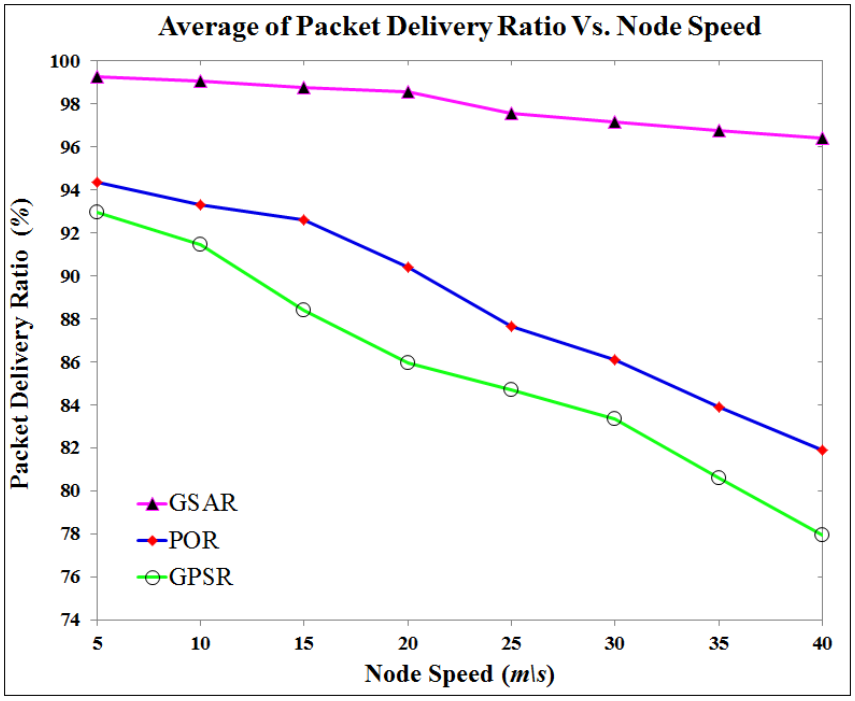


Figure 5.2. Average Packet Delivery Ratio via Node Speed

5.3.1.2 Control Overhead Vs. Node Speed

Figure 5.3 shows the average number of Control Overhead (COH) exchanged by GPSR, POR and GSAR protocols as a function of node speed. A better protocol is the one that has less COH. As demonstrated in Figure 5.3, in all simulated scenarios GPSR and POR send constant rate of control packets, that is 22375 beacon packets, regardless of the nodes mobility rate. On the other hand, GSAR's overhead grows at a smaller rate than GPSR and POR. The simulated scenarios conclude that GSAR reduces the generated routing control traffic by 47% compared to GPSR and POR.

The researcher observes that the overhead curves in Figure 5.3 of GPSR and POR remain stable regardless of the increment of the nodes speeds. This is because the GPSR and the POR require sending the beacon packets proactively in a fixed sending frequency FBPIT. Thus, both of them consume more network bandwidth than does GSAR, which has bad effects on the performance of MANET. The GPSR and the POR are plotted on the same scale as each other because both use the same FBPIT.

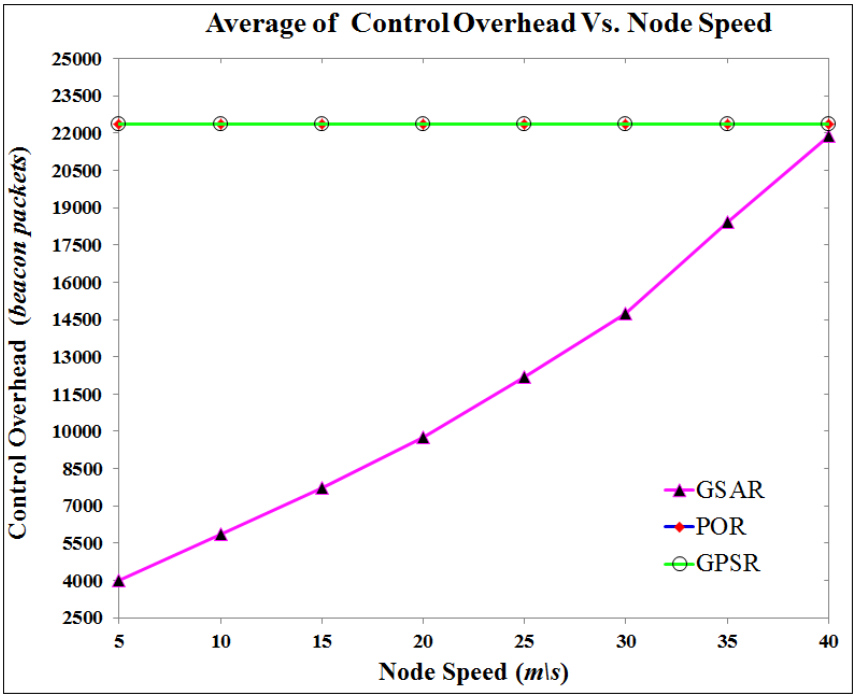


Figure 5.3. Average Control Overhead via Node Speed

However, as shown in Figure 5.3, GSAR shows an increase in the beacon packets sending rate due to the use of the Dynamic Beaconing Update Mechanism (DBUM). The

increment in the beacon packets sending rate of GSAR demonstrates the effectiveness of its adaptive DBUM scheme, in response to mobility's topology changes. Figure 5.3 verifies the utility of DBUM in the GSAR protocol. This figure demonstrates that in low mobility (below 10 *m\s*), DBUM reduces the number of the sent routing packets UBM. Conversely, as nodes speed increase, DBUM generates more routing control packets intelligently based on using the fuzzy logic controller approach DFLCH. This results in occupying less bandwidth by beacons that producing good effects on MANET performance.

5.3.1.3 End-to-End Delay Vs. Node Speed

Figure 5.4 shows the average end-to-end delay (EED) experienced by GPSR, POR and GSAR protocols as a function of node speed. A better protocol is the one that experience less EED. The figure illustrates that the general trend for all protocols is that the experienced delay increases with increased node speed. The eight scenarios of this simulation show that with low node speed, 5 *m/s*, and high node speed, 40 *m/s*, GSAR has EED's of 29.97 *ms* and 62.007 *ms* respectively, as compared to 54.127 *ms* and 99.15 *ms* for POR and 59.118 *ms* and 120.082 *ms* for GPSR. Of the three, GSAR has the lowest EED rate in that GSAR achieves 46.54% and 37.87% lower average EED comparing with GPSR and POR protocols respectively.

Under high mobility network condition, GPSR and POR protocols have low performance in terms of EED. This is due to two main reasons: First, because of using a FBPIT, GPSR or POR protocol suffers from increment in outdated topology knowledge that results in decreasing PDR as discussed in Sub-section 5.3.1.1. Second, GPSR and POR protocols remove neighbours' entries independent of their mobility degree. Thus, the probability of selecting one of these stale neighbours is very high and proportional to nodes mobility rate. As proved in Sub-section 5.3.1.1, forwarding a data packet to a stale neighbour as the next relay-node results in decreasing PDR. To solve the packet loss problem that is caused by the two above reasons, the sender node retransmits the lost packet. Packet retransmission increases the delay because, during that retransmission, the data packet is buffered for extra time.

On the other hand, with GSAR, as the nodes' mobility increases, executing the Dy-

namic Fuzzy Logic Controller Check-time (DFLCH) in CUT algorithm increases to emit UBM. The increment of UBM sending rate keeps more accurate neighbours' information in a node's NLM. As proved in Chapter 3, Sub-section 3.7.2, the fuzzy processing delay in the DFLCH algorithm increases as the node speed increases. The simulation showed that the fuzzy-delay in the DFLCH algorithm method is too small (very few microsecond). This fuzzy-delay is negligible for the GSAR functionality and has no practical relevance in the total EED. Also, the increment in nodes mobility activates the NMEM algorithm to function more rapidly to track and remove the out-dated neighbours in a sender's NLM based on the link lifetime between nodes. As a consequence, in GSAR, the routed data packets reach their final destination in a timely manner with very low EED.

In addition, in a highly dynamic MANET, the DRESM in GSAR contributes in main improvements. Using forwarding candidates' collaboration and selection process to avoid forwarding packets to a hole node by using the DRESM algorithm reduce the delay significantly. This is because these two approaches mitigate the EED required in case of packet loss.

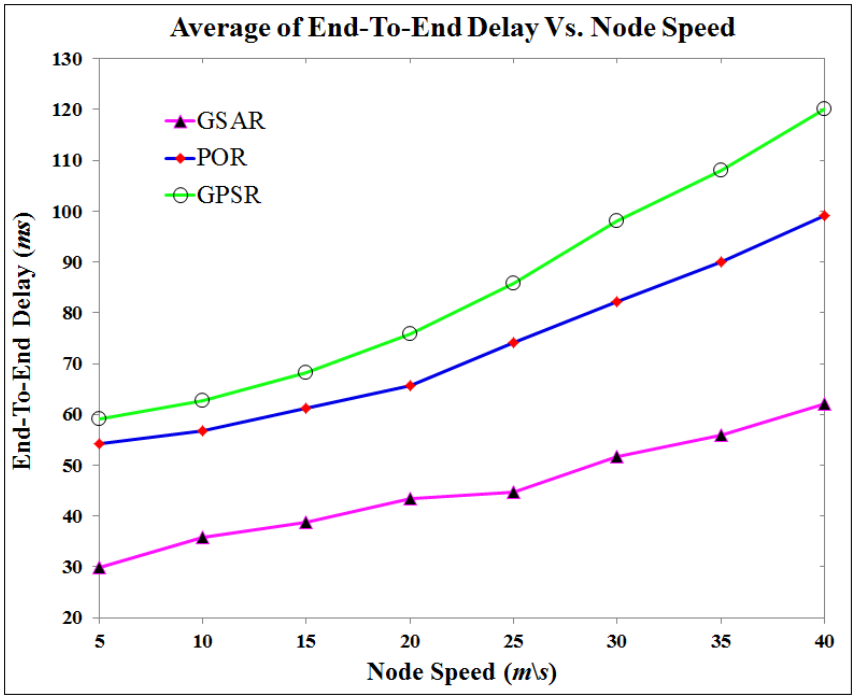


Figure 5.4. Average End-to-End Delay via Node Speed

5.3.1.4 Routing Path Stretch Vs. Node Speed

Figure 5.5 shows the average of routing path stretch (RPS), i.e., path length that represents the routing path average hops count, in GPSR, POR and GSAR protocols as a function of number of speed. A better protocol is the one that has less RPS. This simulation shows that in low and high mobility network conditions GPSR and POR protocols perform worse than GSAR. A comparison in Figure 5.5 below shows that the GSAR protocol reduces the average routing path stretch by about 35.27% and a 44.32% over POR and GPSR respectively.

Figure 5.5 shows that there is a downward trend in the number of packets GPSR and POR deliver in optimal path length as the node speed increases. This trend occurs because of two main reasons. The first reason is because that the proactive beaconing of the both protocols is unable to efficiently track the new mobility's topology change. Hence, selecting a next relay-node with stale position information from the sender's NLM increases as well. The second is that, as node speeds increase, the instances of a data packet encountering a hole will increase as well. Both reasons result in the fact that the packet forwarder becomes unable to make constant progress, unless it uses the recovery mode (face routing) in GPSR or the virtual destination-based void handling (VDVH) in POR. Both approaches lead to using low-efficiency, long detours compared to the shortest path. Further, the results show that GPSR is worse when compared to POR. This is because the alternative path found by VDVH approach in POR is slightly shorter than the one found by the perimeter mode in GPSR.

As Figure 5.5 shows that there is an upward trend in the number of packets GSAR deliver in optimal path length as the node speed increases. This trend occurs because GSAR protocol explicitly considers node mobility in the DBUM scheme by using CUT and NMEM to maintain up-to-date neighbours' information in the nodes' NLMs. This leads to the selection of a more suitable next-relay routing neighbour from the sender NLM. Moreover, due to the nature of the selection process in GSAR, the FLDRE approach, which adapts FLC in the selection process, also considers node mobility. FLDRE, uses the Residual Links Lifetime (RLT) to be one of the five routing metrics to make routing decision; the RLT approach considers the nodes' velocity (direction and speed value).

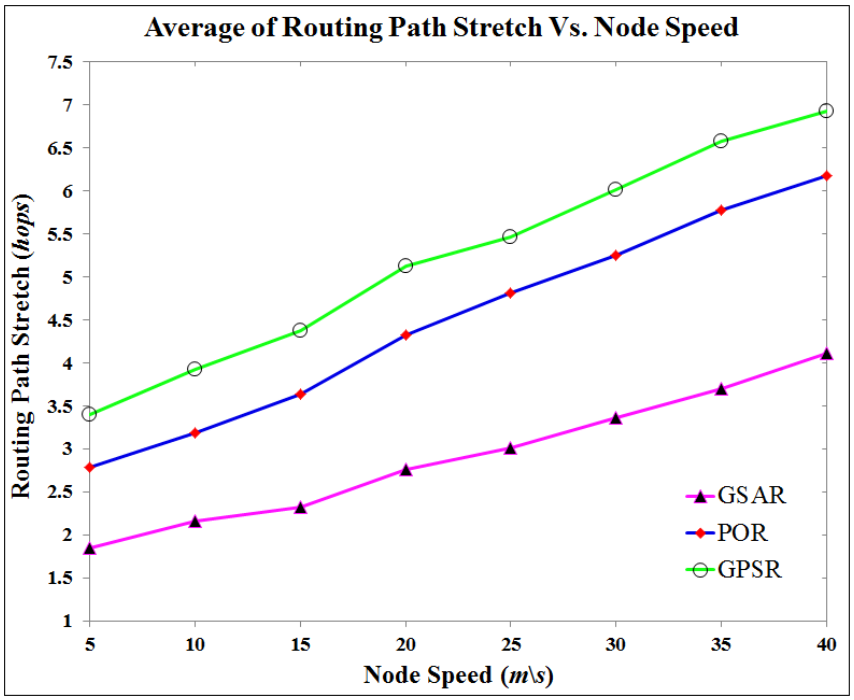


Figure 5.5. Average Routing Path Stretch via Node Speed

5.3.1.5 Inconsistency of Neighbourhood Matrix Vs. Node Speed

Figure 5.6 shows the average of Inconsistency of the Neighbourhood Matrix (INM) ratio in GPSR, POR and GSAR protocols as a function of node speed. A better protocol is the one that has less INM ratio. A comparison in Figure 5.6 shows that the GSAR protocol reduces the INM ratio by about 77% over GPSR and POR.

Compared to the GSAR protocol, the simulation results show that both GPSR and POR routing protocols achieved reasonable average inconsistency of neighbourhood matrix with low speed (below 10 *m\s*). The reason behind their improvement in achieving a higher consistency of neighbourhood matrix rate is attributed to the high beacon packet frequency-sending rate. As the beacon packet sending rate increases the consistency of neighbourhood matrix in the nodes' NLM increases as well. However, as the node speeds increase the inconsistency of neighbourhood matrix rate in both GPSR and POR routing protocols increases as well. The reason behind this increment is that while neighbours move through the transmission range of a node they do not send updated messages because of using FBPIT, which bounds the ELT for fixed time intervals. During this FBPIT, some neighbours might leave the node's transmission

range while at the same time new neighbours might enter its transmission range. This results in increasing the average of INM of both GPSR and POR routing protocols.

GSAR has advantages over GPSR and POR protocols in high mobility's to topology changes due to its routing features. As discussed in Sub-section 5.3.1.2, when node speed increases the beacon packet frequency-sending rate in GSAR routing protocol increases as well. The increment in beacon packet sending rate in GSAR is attributed to the use of CUT approach that increases the transmission of UBM because of using DFLCH scheme. Using CUT to emit UBM in GSAR increases the number of accurate neighbours' entries in a node's NLM matrix rapidly; thus, the INM rate decreases too. Another main reason behind the decrement of INM is attributed to the use of the NMEM algorithm that decreases dramatically the number of obsolete neighbours' entries in nodes' NLMs.

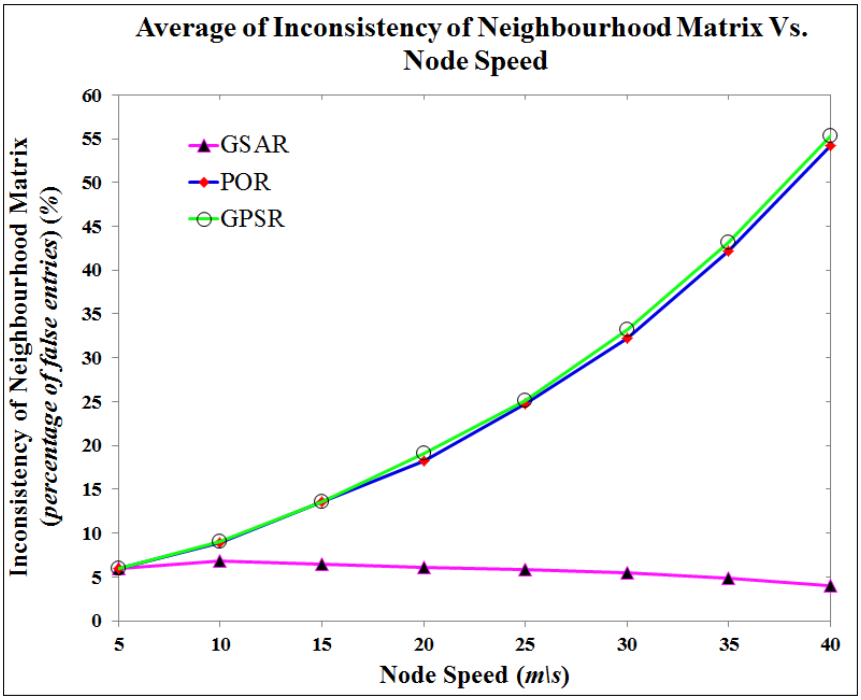


Figure 5.6. Average Inconsistency of Neighbourhood Matrix Ratio via Node Speed

5.3.1.6 Number of Hole Problem Occurrence Vs. Node Speed

Figure 5.7 shows the average number of the packets encountering a hole problem (NHPO) as a function of node moving speed in GPSR, POR and GSAR protocols. A better protocol is the one that has less NHPO. A comparison in Figure 5.7 shows that

the GSAR protocol reduces the average of NHPO by about 79% and a 87% over POR and GPSR respectively.

Figure 5.7 indicates that the increment in nodes speeds results in higher occurrences of hole problem while using GPSR and POR protocols. The situation is better in POR compared to GPSR. The reason is the modification done for MAC in POR protocol, in which the DATA packet can be received by the next relay node and another two neighbours. Hence, in case the ACK packet was not heard from by the optimal next relay-node due to any reason, one of the two other candidate nodes will forward the DATA packet. However, POR still suffers from a high rate of encountering a hole problem because it just uses one routing metric to make the forwarding decision, which is distance. In GPSR and POR, when a packet encounters a hole situation during forwarding, the perimeter mode and VDVH mode are activated to enable the packet to bypass the hole. As discussed earlier, this decreases the PDR, increases EED, and lengthens the average of RPS.

The results shown in Figure 5.7 reveal the effectiveness of the GSAR protocol in decreased encountering of the hole problem during the packets forwarding process in most scenarios. The reason behind this success is attributed to three main reasons. The first reason is the efficiency of the proposed DRESM in GSAR. In FLDRE as a part of DRESM, the source node should be sure that the next relay-node has at least one positive neighbour in the direction of the destination; otherwise, it will not forward the packet to that neighbour. In case none of the neighbours satisfy this condition, because of a disconnected network for any reason, the sender node caches the packet and moves until either TTL counter reaches zero or it finds a neighbour closer to the destination than itself. The second reason is attributed to the IDOTM approach, which is a part of DRESM. In IDOTM, once a node does not reply with a CTF message it is considered to be a stuck node, so, it will not be selected as the next relay-node. Hence, the probability of the packet encountering the hole problem is lowered. The third reason is attributed to the DBUM mechanism. With DBUM, as nodes speeds increase, the CUT approach is activated to send more UBM packets. More and more UBM decreases inaccurate node information in a node's NLM. Further, using the NMEM approach, which is a part of DBUM, increases the information accuracy in a node's NLM. This

decreases the probability of a packet encountering a hole problem by avoiding selecting an invalid neighbour. In GSAR, the decrement in encountering a hole problem causes a packet to be routed via the close to optimal (i.e., shortest) route. This increases the packet delivery ratio, decreases end-to-end delay, and shortens the average of the routing path stretch.

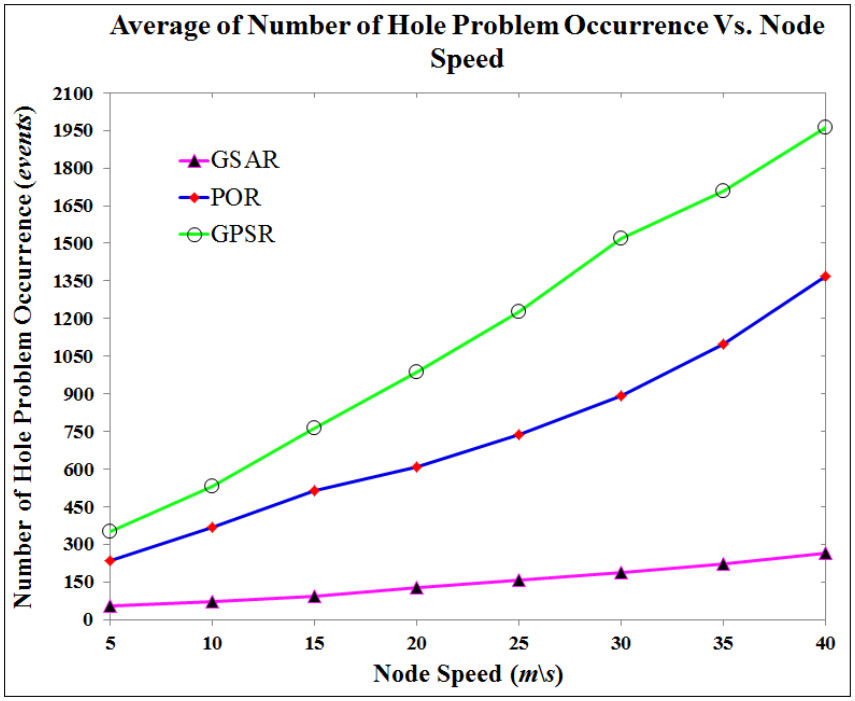


Figure 5.7. Average Number of Hole Problem Occurrence via Node Speed

5.3.1.7 Summary Results of Varying Node Speed

This section summarizes the average performance results of GSAR, POR and GPSR routing protocols under various node speeds. The results in the Table 5.2 below indicate that GSAR, POR and GPSR routing protocols have approximately good performance under low node speeds. This is because low mobility results in low topology changes that have low bad influence on the performance of the routing protocols. However, as the speeds of the nodes increase, the performance of routing protocols decreases. This is because high mobility results in frequent topology changes that affect the performance of the routing protocols.

Table 5.2 below indicates that GSAR routing protocol can performs well under high nodes speeds and outperforms POR and GPSR protocols in terms of PDR, COH, EED,

INM, RPS and NHPO for the reasons discussed in the previous sections.

Table 5.2. The Performance Comparison between GSAR, POR and GPSR vs. Node Speed

Metric	Protocol	Minimum	Maximum	Mean	GSAR Improvement
PDR	**GSAR**	96.421	99.263	97.931	-
	POR	81.871	94.327	88.771	10.31%
	GPSR	77.935	92.958	85.657	14.32%
COH	**GSAR**	3980	21890	11823	-
	POR	22375	22375	22375	-47.159%
	GPSR	22375	22375	22375	-47.159%
EED	**GSAR**	29.97	62.007	45.304	-
	POR	54.127	99.15	72.918	-37.869%
	GPSR	59.118	120.082	84.748	-46.542%
RPS	**GSAR**	1.85	4.112	2.908	-
	POR	2.78	6.174	4.493	-35.27%
	GPSR	3.4	6.921	5.223	-44.32%
INM	**GSAR**	4.04	6.03	5.702	-
	POR	6.03	54.27	25.037	-77.225%
	GPSR	6.03	55.275	25.565	-77.696%
NHPO	**GSAR**	53.561	263.351	146.606	-
	POR	235.8	1370.061	728.29	-79.86%
	GPSR	351	1960	1130.88	-87.03%

GSAR improvement $= \frac{GSAR\,mean - GPSR\,or\,POR\,mean}{GPSR\,or\,POR\,mean} \times 100\%$, Positive values means increment in the percentage and negative values means decrement in the percentage For example, GSAR outperforms GPSR by increasing PDR by 17.39%, decreasing COH by 47.159%, and decreasing EED by 46.542%, etc.

5.3.2 Simulation 2: Impact of Varying Number of Nodes

This section evaluates the effects of varying the number of nodes on the performance of GSAR, GPSR and POR routing protocols. In the following simulation, the number of nodes is the only parameter that controls the network connectivity. This simulation examines eight scenarios by varying the number of nodes. Table 5.3 below shows the parameters that are used in this simulation.

Table 5.3. Varying Number of Nodes

Parameter	Value
No. of nodes	25, 50, 75, 100, 125, 150, 175, 200
Terain size	2000 $m \times 450$ m
Node speed	20 m/s
simulation time	900 sec
Traffic type	CBR
Packet rate	2 $Kbps$
Packet size	256 bytes
Number of data traffic sources	5
Transmission range	250 m
Movement model	Modyfied Random Waypoint
MAC layer protocol	IEEE 802.11 DCF
Radio propagation model	Two Ray Ground Model
The other parameters are set as the same as in Table 3.6	

5.3.2.1 Packet Delivery Ratio Vs. Number of Nodes

Figure 5.8 shows the average packet delivery ratio (PDR) achieved by GPSR, POR and GSAR protocols as a function of the number of nodes. A better protocol is the one that has more PDR. As the figure clearly illustrates, more than 98.8% of the data packets of GSAR can be delivered successfully to specified destinations in all simulated scenarios. The GPSR achieves the lowest performance success delivery rate with only 87.18%.

In all protocols, as the number of nodes increases in the network, a node's degree increases too. As a result of this increment, the number of a node's neighbours' entries that the node should track (i.e., add\delete) increases as well. GPSR and POR use the inefficient technique to track the neighbours' entries in nodes' NLMs. Thus, the probability of selecting one of these outdated neighbours as the next relay-node in-

creases as well. This leads to decrease the performance of POR and GSAR protocols in terms of PDR. Moreover, as the number of nodes increases in the network (above 100 nodes), both GPSR and POR use greedy routing mode more frequently. This results in two consequences. First, while using the greedy mode in both protocols, the next selected forwarder node is closer to the border of the transmission range; this results in frequent link failure that results in packet loss. Second, the hotspot problem at the centre of the network appears frequently, which results in more congestion and collision problems. This collision and congestion greatly reduces the percentage of packets successfully delivered at respective destinations. Figure 5.9 demonstrates the effect of the increment of the participants' number on the congestion and collision problems in the network.

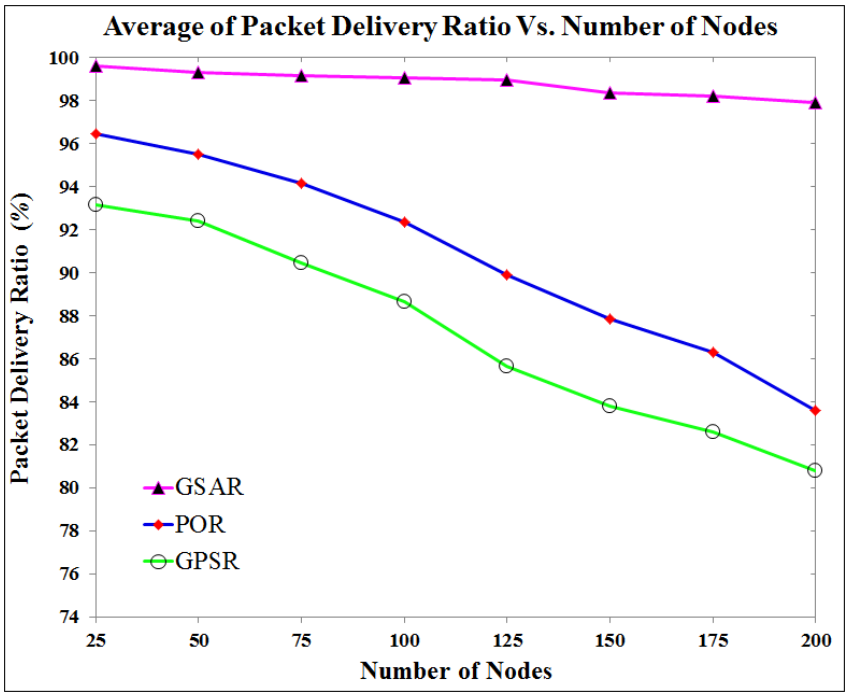


Figure 5.8. Average Packet Delivery Ratio via Number of Nodes

On the other hand, as shown in Figure 5.8above, due to its new features, GSAR is much better than both protocols. GSAR seems to have a stable delivery ratio due to the following reasons. By using NMEM with GSAR, the neighbours' ELT in a node's NLM is updated dynamically, regardless of the sender's degree. Moreover, using FL-DRE in GSAR guarantees that hotspot problem does not appear. Thus, the final packet ratio delivered at destination side is increased. Also, by using the DPS algorithm in

GSAR, each intermediate node forwards the data packet based on accurate position information of the targeted destination. This also contributes to increase of the delivered packets at respective destinations.

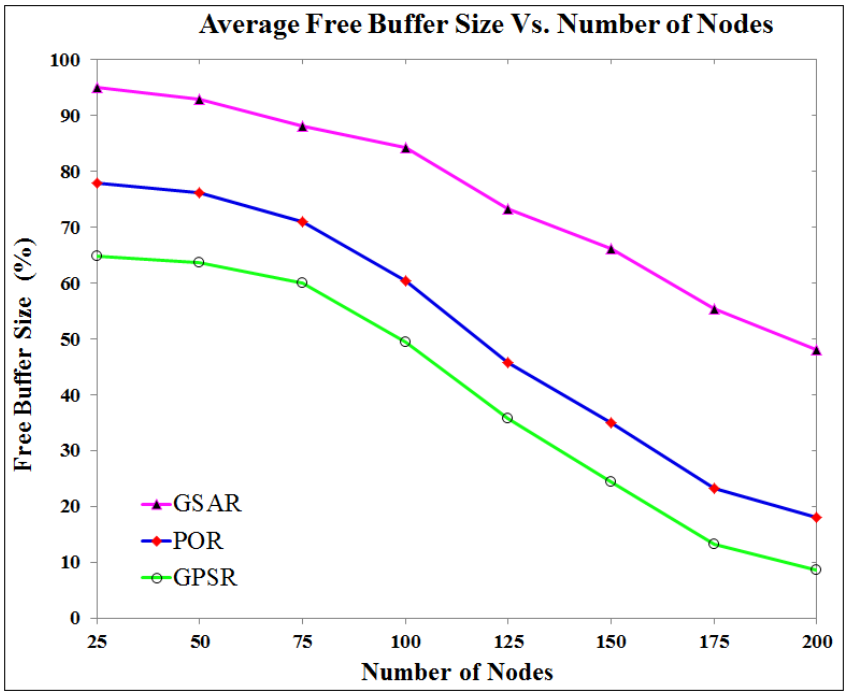
The Effect of Number of Nodes on Contention and Congestion

Figure 5.9 shows the average of collision and congestion in GPSR, POR and GSAR protocols, as a function of number of nodes in the network. To monitor the collision status, the researcher observes the number of collide Request To Forward (RTF) packets occur on the network as a pointer to the collision state of the network. To monitor the congestion status, the researcher observes the changes occur on the nodes free buffer size as a pointer to the congestion state of the network.

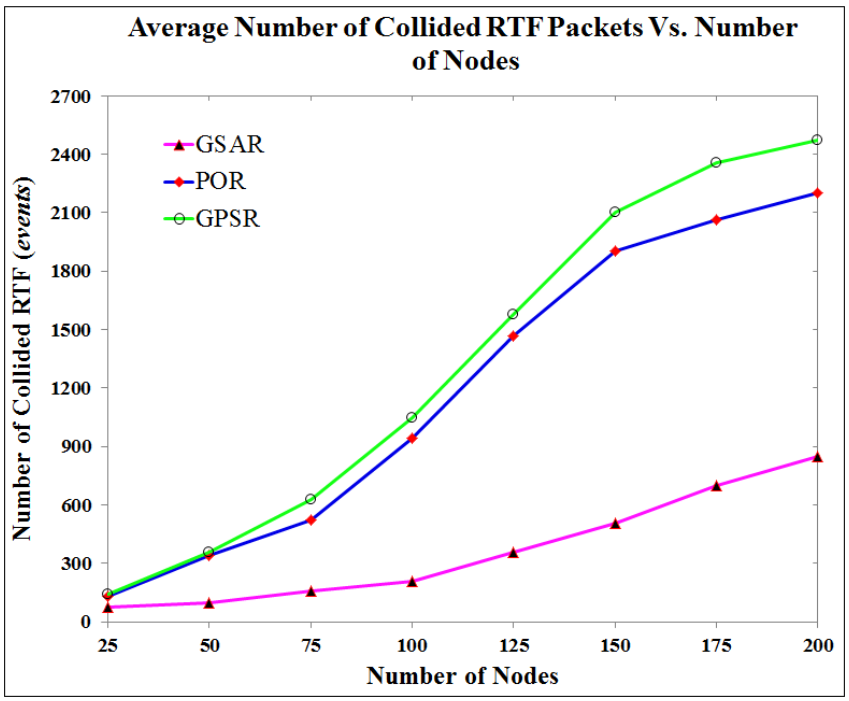
Figure 5.9 (a) shows the congestion rate by observing the changes occur on the nodes free buffer size. The figure reveals that as the number of the participants in the network increases, the nodes' buffers free size decreases (i.e., congestion increases). Figure 5.9 (a) shows that GSAR always has better performance as it degrades more gracefully comparing to GPSR and POR. It shows that GSAR reduces the congestion rate by about 48% and a 88.73% over POR and GPSR respectively.

Figure 5.9 (b) shows that as more nodes join the network the collision rate increases, this is because more and more nodes try to enter the shared medium. This leads to severe collisions because the number of participant nodes that overhearing each other's radio transmissions increases as well. For example, in GPSR, POR and GSAR, when the number of nodes increases from twenty five to fifty nodes in the network, the number of packet suffer from collision in the three protocols increases from 75, 130, and 142.5 to 95, 341, and 345 respectively. The increment in the collision between transmitted packets causes a receiving node to fail to receive a part of data packets.

Figure 5.9 (b) shows that GSAR reduces the collision packet rate by about 69.25% and a 72.41% over POR and GPSR respectively. This is because, in GSAR, the IDOTM approach uses an improved MAC protocol that contributes in reducing the collision rate.



(a)



(b)

Figure 5.9. (a): Average Free Buffer Size via Number of Nodes, (b): Average Number of Collided RTF Packets via Number of Nodes

5.3.2.2 Control Overhead Vs. Number of Nodes

Figure 5.10 shows the average number of Control Overhead (COH) exchanged by GPSR, POR and GSAR protocols as a function of number of nodes. A better protocol is the one that has less COH. As the figure clearly illustrates, for all protocols, the number of beacon packets sent by nodes is directly proportional to the number of nodes. The simulated scenarios show that GSAR reduces the generated routing control traffic by about 41.13% compared to GPSR and POR protocols.

GPSR and POR suffer from excessive control overhead. This is because as the number of the nodes increases in the network, the sum of the emitted beacon packets (PBs) increases as well. The reason behind this increment is that GPSR and POR protocols use FBPIT strategy to emit PB that achieve the highest number of sent BPs. This simulation shows that due to using DBUM approach, GSAR attains the lowest number of the sent beacon packets. This is because that in this simulation the number of BP is sent with respect to using constant velocity. Thus, the nodes that use CUT send UBM very rarely because of using the Dynamic Fuzzy Logic Controller Check-time (DFLCH).

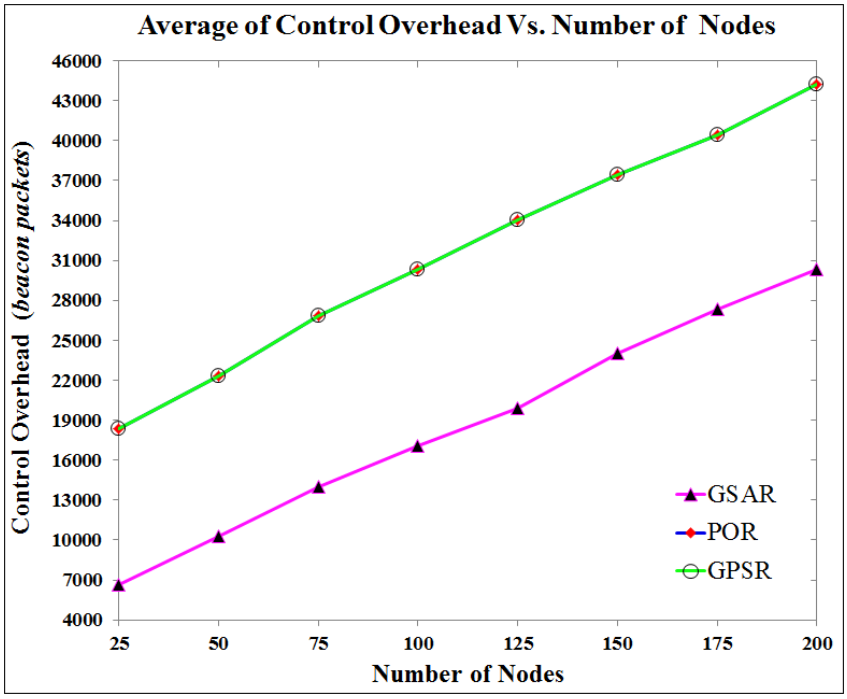


Figure 5.10. Average Control Overhead via Number of Nodes

To conclude, the curves in Figure 5.10 shows that in small and large number of nodes in the network, the GSAR protocol outperforms both GPSR and POR routing protocols in terms of routing overhead. When the average nodes number is 100 (the median number of nodes in this simulation), GSAR emits 43.74% beacon packets fewer than GPSR and POR. GPSR and POR are plotted on the same scale as each other.

5.3.2.3 End-to-End Delay Vs. Number of Nodes

Figure 5.11 shows the average end-to-end delay (EED) in the GPSR, POR and GSAR protocols as a function of the number of nodes. A better protocol is the one that has less EED. The figure illustrates that the general trend in all protocols that the delay experienced increases with the increase of the number of nodes. Figure 5.11 shows that in small and large number of nodes in the network, GPSR and POR routing protocols perform worse than GSAR in terms of EED. For small networks, GSAR has an end-to-end delay lower than GPSR and POR; for a network of 25 nodes GSAR has 55.64 *ms* end-to-end delay while GPSR and POR have 98.21 *ms* and 86.73 *ms*, respectively. However, when the network has 200 nodes, GSAR's end-to-end delay is 94.77 *ms*, which is 37.46% lower than GPSR's 151.55 *ms* and 28.55% lower than POR's 132.64 *ms*.

The increment in the EED values when using both GPSR and POR is attributed to two reasons. The former reason is the use of alternative path between the source and destination nodes when the forwarded packet reaches hole, namely, using the perimeter mode in GPSR and VDVH with POR. The influence of the first reason is decreased as the number of the nodes in the network increases. The later reason for increasing the experienced EED values is that as the sender's degree increases the number of outdated neighbours in its NLM increases too. Hence, the probability of selecting one of these outdated neighbours as the next relay-node increases as well. If an outdated neighbouring node is chosen as the next relay-node, the routed data packet will be lost. This incurs more delay to buffer the data packet during retransmission and during selecting alternative next relay-node. As a consequence, the average end-to-end delay is increased significantly. POR is better than GPSR due to selecting proactively the forwarding candidate nodes. This proactive selecting can prevent wasting more routing

time and decreases the end-to-end delay.

The success and effectiveness of GSAR is attributed mainly to its new features, which contribute in decreasing the experienced end-to-end delay to deliver packet at the destination side. With GSAR, as the number of participants in the network increases, the NMEM algorithm tracks and removes the outdated neighbours in the senders' NLM, independent of sender's degree. Therefore, the outdated neighbouring node is prevented from being chosen as the routing node. Also, NMEM approach is supported by the effective selection process in DRESM because of FLDRE advantages. In addition, as proved in Chapter 4, Sub-section 4.7, the fuzzy processing delay incurred by FLDRE in the DRESM algorithm show that when the number of nodes is 110 nodes GPSR-DRESM has the same processing delay as GPSR, also it achieves higher 0.13% average processing delay (fuzzy-delay) when the number of nodes less than 110 nodes compared to GPSR. Also, GPSR-DRESM achieves less 0.26% average processing delays (fuzzy-delay) when the number of nodes greater than 110 nodes compared to GPSR. This fuzzy-delay is negligible for the GSAR functionality and has no practical relevance in the total EED.

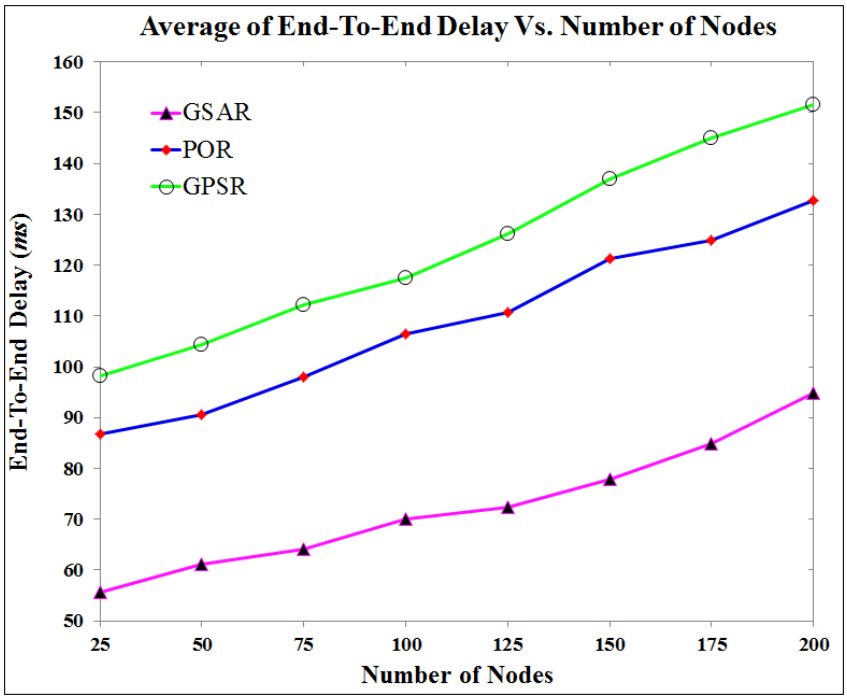


Figure 5.11. Average End-to-End Delay via Number of Nodes

5.3.2.4 Routing Path Stretch Vs. Number of Nodes

Figure 5.12 shows the average of routing path stretch (RPS), i.e., path length that represents the routing path average hops count, in GPSR, POR and GSAR protocols, as a function of number of nodes. A better protocol is the one that has less RPS. It is obvious from the curves in Figure 5.12 that for all protocols, when number of nodes increases, the average hop count decreases. Generally, it can be concluded from this simulation that as the number of nodes increases, routing paths between communicating nodes become increasingly close to the optimum. Also, the simulation results showed that GSAR is the best among the other protocols. When the number of nodes is 100 nodes (the median number of nodes in this simulation), the GSAR protocol reduces the routing path stretch of about a 48.19% and a 40.99% over GPSR and POR respectively.

As it will be shown in Figure 5.14, for both GPSR and POR, when the number of nodes is less than 75 (network was sparsely connected) the probability of encountering the hole problem in which more destinations become disconnected is high. The presence of disconnected destinations affects both GPSR and POR in that they both must use a longer inefficient alternative route. The result also shows that GPSR is worse when compared to POR. This is because the alternative path found by virtual destination in POR is slightly shorter than the one found by perimeter mode in GPSR. However, when the network has 200 nodes, when using GPSR, the most forwarded packets are delivered to their destination with pure greedy forwarding. This results in shortening the used routing paths. When using POR, there is no need to use the VDVH scheme very often and using the greedy approach results in shortening of the routing paths.

From the curves in Figure 5.12, the researcher observed that using GSAR achieves better improvement in the number of travelled hops compared to GPSR and POR. This is because GSAR can track the mobility's topology changes by using CUT and NMEM and also because, in a sparse network, the disconnected destinations problem is handled proactively by using the destination prediction scheme (DPS) approach. In case of a hole occurrence, GSAR can easily avoid this problem by using the carry-and-move approach. Moreover, using the multi-objectives FLDRE approach, adapting FLC in the selection process, helps GSAR to forward the data packet using a close to

the optimum path in most cases with lowest hop count.

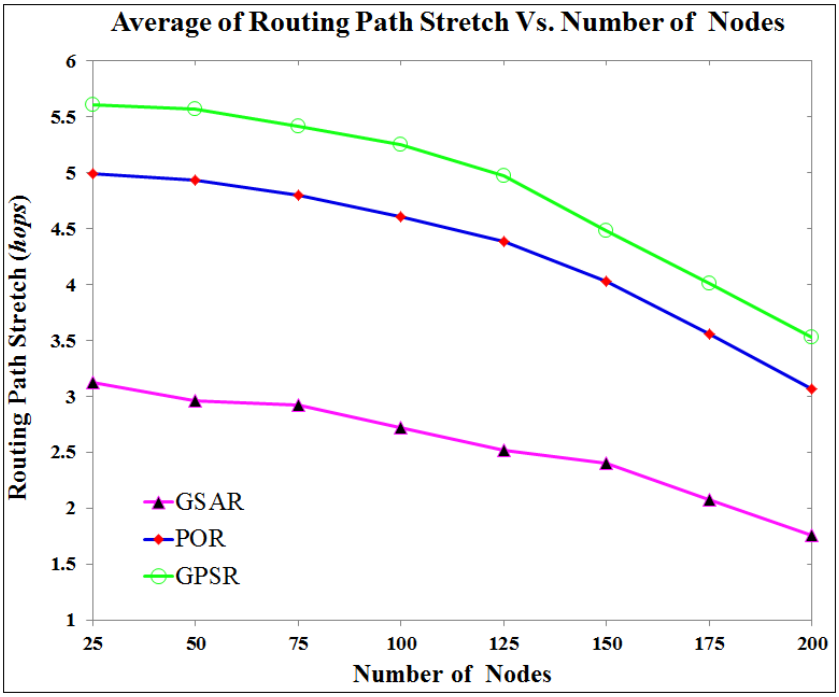


Figure 5.12. Average Routing Path Stretch via Number of Nodes

5.3.2.5 Inconsistency of Neighbourhood Matrix Vs. Number of Nodes

Figure 5.13 shows the average of the inconsistency of the neighbourhood matrix (INM) ratio in GPSR, POR and GSAR protocols as a function of the number of nodes. A better protocol is the one that has less INM ratio. In all protocols, as the number of nodes increases with the same network area, the number of a node's degree increases too (i.e., node's neighbours in its transmission range). This simulation shows that in small and large number of nodes in the network, GSAR and POR routing protocols perform worse than GSAR. When the number of nodes is 100 nodes (the median number of nodes in this simulation), the inaccuracy percentage of information rate of GSAR is 72.59% lower than POR and 73.44% lower than GPSR.

As shown in Figure 5.13, in GPSR and POR, when the node's degree increases, the number of the detected incorrect neighbours' entries in the node's neighbours-list increases too. This is because deleting the neighbour's entry is only based on the sending frequency of the beacon packets.

However, Figure 5.13 above also shows the effectiveness of GSAR protocol whereby

the number of incorrect neighbours listed in nodes' NLM is much lower and the average inconsistency of neighbourhood matrix seems to be stable. The success and effectiveness of GSAR is mainly attributed to its new features, the RLT and NMEM approaches. Those features contribute to decreasing the average inconsistency of neighbourhood in nodes' matrices. The nodes using GSAR protocol remove the outdated entries of its neighbours, relying on RLT between the communicating nodes regardless of the increment in a node's degree.

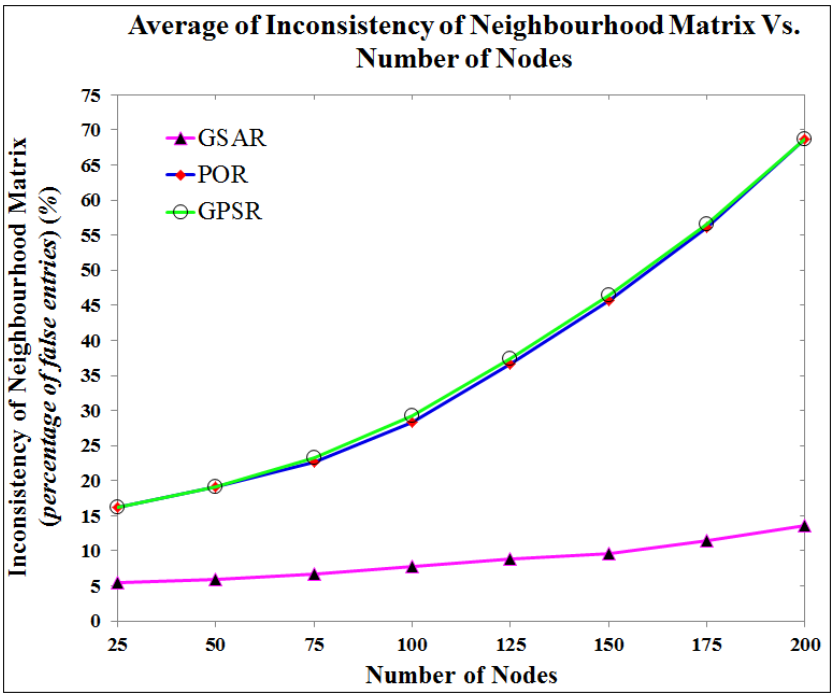


Figure 5.13. Average Inconsistency of Neighbourhood Matrix Ratio via Number of Nodes

5.3.2.6 Number of Hole Problem Occurrence Vs. Number of Nodes

Figure 5.14 shows the average number of the packets encountering a hole problem (NHPO) as a function of number of nodes in GPSR, POR and GSAR during the DATA packets forwarding process. A better protocol is the one that has less NHPO. From the simulation results presented in Figure 5.14, it is observed that GPSR and POR routing protocols perform worse than GSAR. When the number of nodes is 100 nodes (the median number of nodes in this simulation), the encountering a hole problem of GSAR is 95.75% lower than GPSR and 92.62% lower than POR.

The number of nodes dominates the hole problem for GPSR and POR. When the network was sparsely connected (i.e., less than 75 nodes), most routed packets encountered hole problems. Even after the number of nodes is increased four times to be 100 nodes (the median number of nodes in this simulation), a high ratio of the routed packets still reached a hole situation. Figure 5.14 shows that GPSR is the worst of the two routing protocols, and POR is better than GPSR due to the modification of MAC that enabling three nodes to handle the routed data packet instead of just only one next relay-node as in GPSR.

As Figure 5.14 reveals, the increment in the number of nodes in the network has a tiny effect on the performance of GSAR. This is due to several reasons. First, GSAR utilizes the DRESM mechanism to select the next relay-node based on multi-routing criteria that guarantee that the selected next relay-node has at least one positive neighbour in the direction of the destination. Second, GSAR utilizes the IDOTM technique in DRESM to route the data packet by another two candidate nodes along with the optimal next relay-node. And finally, in case none of the selected three nodes can forward the packet, the sender node caches the packet and move until either the TTL counter reaches zero or it finds a neighbour closer to the destination than itself. These three reasons aided GSAR to take the non-hotspot path rather than the shortest path. As the number of encountering a hole situation substantially reduced to match zeroes in GSAR, a packet becomes routed via a close-to-shortest route (optimal rout). The use of the close-to-shortest route (optimal rout) increases the packet delivery, decreases end-to-end delay, and shortens the average of routing path stretch as shown previously in Sections 5.3.2.1, 5.3.2.3 and 5.3.2.4.

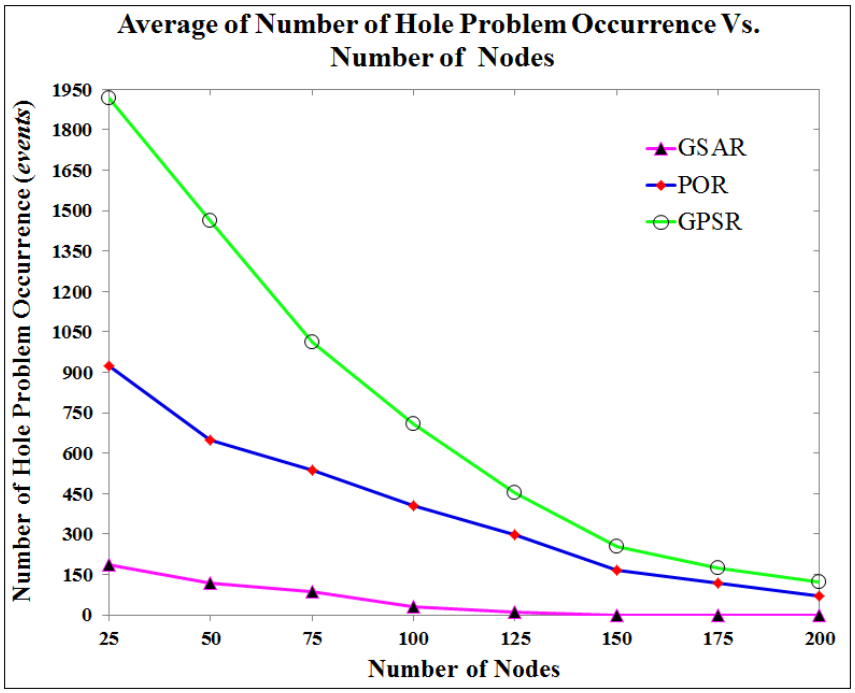


Figure 5.14. Average Number of Hole Problem Occurrence via Number of Nodes

5.3.2.7 Summary Results of Varying Number of Nodes

This section summarizes the average performance results of GSAR, POR and GPSR routing protocols under various number of nodes. The results in Table5.4 indicate that varying the number of the nodes in the same network affects the overall performance of GSAR, POR and GPSR routing protocols. This is because more and more numbers of nodes try to access the shared medium, creating a bad influence on the performance of the routing protocols. In this simulation, as the number of nodes increases the performance of the GSAR, POR and GPSR routing protocols decreases.

Table 5.4 below indicates that GSAR routing protocol can perform well under high number of nodes and outperforms POR and GPSR protocols in terms of PDR, COH, EED, INM, RPS and NHPO for the reasons discussed in the previous sections.

Table 5.4. The Performance Comparison between GSAR, POR and GPSR vs. Number of Nodes

Metric	Protocol	Minimum	Maximum	Mean	GSAR Improvement
	GSAR	97.888	99.604	98.818	-
PDR	**POR**	83.584	96.456	90.757	8..88 %
	GPSR	80.804	93.171	87.185	13.34 %
	GSAR	6656	30301	18690.629	-
COH	**POR**	18379	44210	31754.125	-41.139%
	GPSR	18379	44210	31754.125	-41.139%
	GSAR	55.643	94.772	72.6	-
EED	**POR**	86.732	132.642	108.865	-33.311%
	GPSR	98.214	151.552	123.975	- 41.439%
	GSAR	1.756	3.12	2.558	-
RPS	**POR**	3.07	4.99	4.296	-40.45%
	GPSR	3.52	5.61	4.854	-47.3%
	GSAR	5.443	13.55	8.675	-
INM	**POR**	16.16	68.68	36.675	-76.346%
	GPSR	16.16	68.68	37.117	-76.627%
	GSAR	0	187	54	-
NHPO	**POR**	70.965	923.132	395.531	-86.347%
	GPSR	121	1919	762.75	-92.92%

GSAR improvement $= \frac{GSAR\,mean - GPSR\,or\,POR\,mean}{GPSR\,or\,POR\,mean} \times 100\%$, Positive values means increment in the percentage and negative values means reduction in the percentage

5.3.3 Simulation 3: Impact of Varying the Number of Traffic Sources

This section evaluates the effects of traffic load in the network on the performance of GSAR, GPSR and POR routing protocols. In the following simulation, the number of sources is the only parameter that varies in the network; the other parameters are fixed. This simulation examines five scenarios by varying the number of load traffic sources. Table 5.5 below shows the parameters that are used in this simulation.

Table 5.5. Varying Number of Traffic Sources

Parameter	Value
No. of nodes	100
Terain size	2000 $m \times 450$ m
Node speed	20 m/s
simulation time	900 sec
Traffic type	CBR
Packet rate	2 $Kbps$
Packet size	256 bytes
Number of data traffic sources	5, 10, 15, 20, 25
Transmission range	250 m
Movement model	Modyfied Random Waypoint
MAC layer protocol	IEEE 802.11 DCF
Radio propagation model	Two Ray Ground Model
The other parameters are set as the same as in Table 3.6	

5.3.3.1 Packet Delivery Ratio Vs. Number of Data Traffic Sources

Figure 5.15 shows the average packet delivery ratio (PDR) achieved by GPSR, POR and GSAR protocols as a function of number of sources. A better protocol is the one that has more PDR. For all protocols, increasing the number of sources means that the number of data packets to be rerouted at the interface of nodes increases too. The simulation results in Figure 5.15 revealed that the general trend in all protocols is that the packet delivery decreases as the number of sources increases. Of the three, GSAR has the highest packet delivery rate. When the number of CBR flows is less than 5, GSAR's packet delivery rate is more than 97%, while GPSR's ranges from 87% to 77% and POR's from 90% to 80%. GSAR gains high data packets due to its effective new features.

GPSR and POR are single routing objective protocols that use the shortest path routing

as the only criterion. This results in heavily loading those nodes located at the centre of the network. Such an unequal use of the nodes causes them to be more congested than other nodes in the network. The routed packet to those nodes has a high probability of being dropped. This leads to decreased performance of the POR and GPSR protocols in terms of packet delivery. Figure 5.16 demonstrates the effect of the increament of injected traffic on the congestion problems.

On the other hand, with GSAR and because of using DRESM algorithm, the information of a neighbour in any node's NLM is always accurate because of using IDOTM algorithm, namely using RTF\CTF\DATA as beacon equivalent. Moreover, the unique benefit of using FLDRE appears clearly as the traffic load increases. Due to adopting FLC, FLDRE gives GSAR the ability to handle the traffic load, distribute it evenly all over the network and prevent to traverse the data packet through the high congested and out-of-battery power nodes at the centre of the network. Thus, the routed packets correctly reach their final destination.

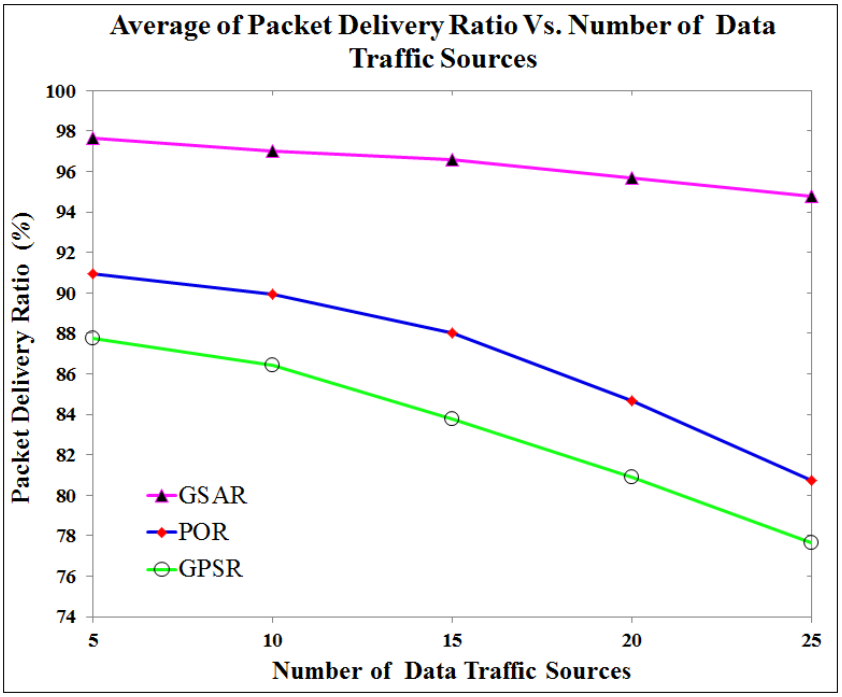
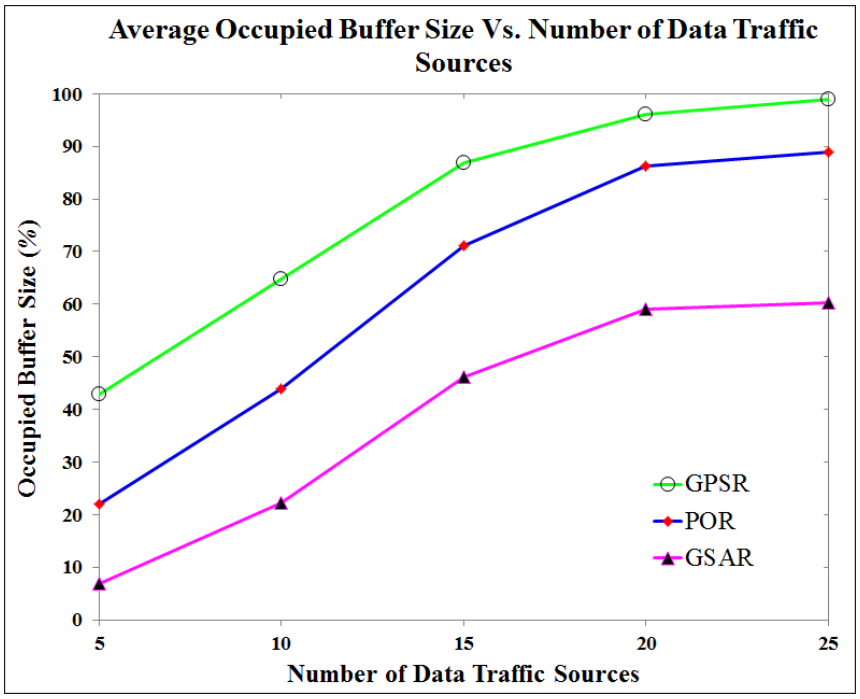


Figure 5.15. Average Packet Delivery Ratio via Number of Data Traffic Sources

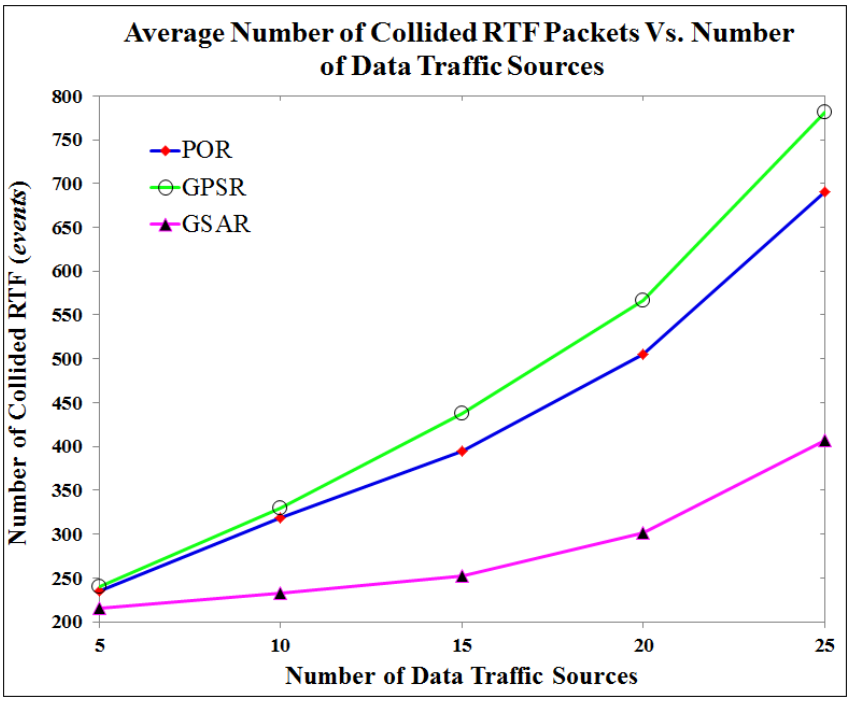
The Effect of Number of Sources on Contention and Congestion

Figure 5.16 shows the average of collision and congestion in GPSR, POR and GSAR protocols as a function of number of sources in the network. To monitor the collision status, the researcher observes the number of collide RTF packets occur on the network as a pointer to the collision state of the network. To monitor the congestion status, the researcher observes the changes occur on the nodes free buffer size as a pointer to the congestion state of the network. The figures reveal that as the number of sources increases, this introduces a great number of packet collisions. Due to MAC layer collisions, congestion will occur. Thus, the nodes' free buffer size decreases very fast that may results in packet drops.

Figure 5.16 (a) shows that GSAR reduces the congestion rate by about 50.08% and a 37.73% over GPSR and POR respectively. Also, Figure 5.16 (b) shows that GSAR reduces the collision packet rate by about 40.14% and a 34.27% over GPSR and POR respectively. The improvement in GSAR protocol is attributed to the new features of its new mechanisms. DRESM can distribute the load evenly all over the network. DRESM selects the next relay-node that has the maximum free buffer size. Also, as discussed in Section 4.3.1, in GSAR, the IDOTM approach uses an improved MAC protocol that contributes in reducing the collision rate.



(a)



(b)

Figure 5.16. (a): Average Occupied Buffer Size via Number of Data Traffic Sources, (b): Average Number of Collided RTF Packets via Number of Data Traffic Sources

5.3.3.2 Control Overhead Vs. Number of Data Traffic Sources

Figure 5.17 shows the average number of Control Overhead (COH) exchanged by GPSR, POR and GSAR protocols as a function of number of data traffic sources. A better protocol is the one that has less COH. This simulation shows that GSAR emits much lower beacon packets than both GPSR and POR. Only less than 10123 beacon packets are sent by GSAR. GPSR and POR have a similar high beacon packets count in that about 22107 beacon packets are sent.

For both GPSR and POR protocols, the number of beacon packets sent by participating nodes seems to be slightly static, because beaconing in both protocols is performed in static manner regardless of the traffic-sending rate. Because a data packet piggybacks the information of its sender, the increase of traffic could help update the information of nodes and slightly contribute in reducing the beacons sending rate. GPSR and POR

are plotted on the same scale as each other.

The success and effectiveness of GSAR is mainly attributed to its new features, that are the CUT and IDOTM techniques. In this simulation, participating nodes use constant speed; hence, GSAR sends UBM packets very rarely by using DBUM mechanism. This results in decreasing the sent beacon packets in the network by GSAR regardless of the increment in data traffic sources. Another essential thing that contributes in decreasing the sent beacon packet is that GSAR piggybacks the information of a sender node on all communicated DATA, RTF and CTF packets when performing the IDOTM technique. In such a case, as the sent traffic increases, the sending rate of urgent control packets UBM decreases as well.

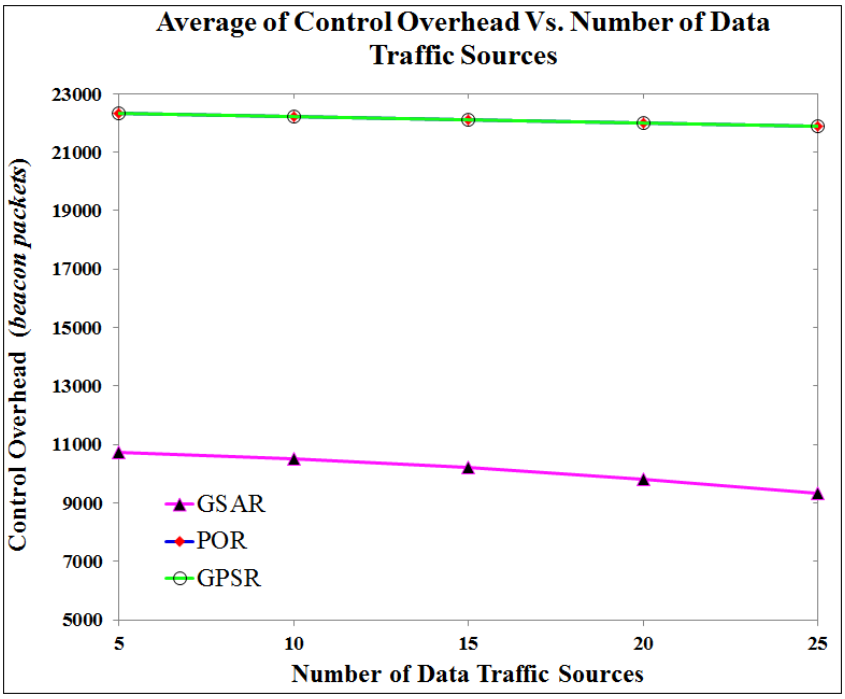


Figure 5.17. Average Control Overhead via Number of Data Traffic Sources

5.3.3.3 End-to-End Delay Vs. Number of Data Traffic Sources

Figure 5.18 shows the average end-to-end delay (EED) in GPSR, POR and GSAR protocols as a function of number of data traffic sources. A better protocol is the one that has less EED. This simulation shows that GSAR experiences a lower EED than both GPSR and POR. With low number of sources, i.e., 5 sources, and high number of sources, i.e., 25 sources, GSAR has an EED of 48.154 *ms* and 98.871 *ms* respectively,

as compared to 103.32 *ms* and 208.456 *ms* for GPSR and 84.601 *ms* and 155.143 *ms* for POR. Thus, in all scenarios, GSAR achieves 51.627% and 37.626% lower average end-to-end delays comparing with GPSR and POR protocols respectively.

As discussed earlier in Sub-section 5.3.3.1, in GPSR and POR the increment in the number of data traffic sources results in more collision and congestion at the centre of the network that increases the probability of packet loss. With GPSR and POR, this incurs more delays in buffering the data packet during retransmission and during selecting the new next relay-node, resulting in a significant longer average end-to-end delay. POR is better in this instance compared to GPSR, due to applying the proactive selection of the forwarding candidate nodes.

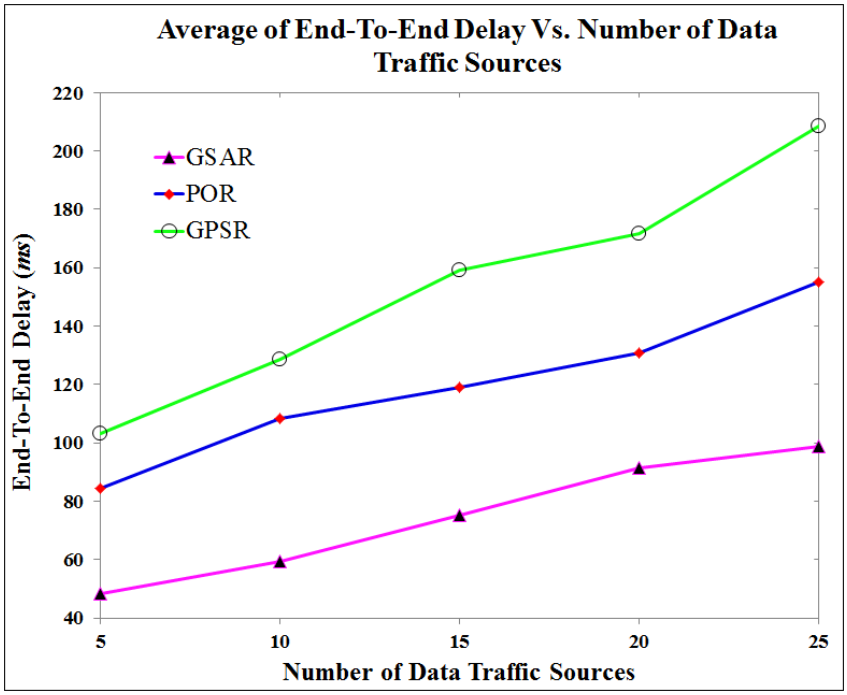


Figure 5.18. Average End-to-End Delay via Number of Data Traffic Sources

On the other hand, while using GSAR, and because of using DRESM algorithm, the information of the neighbours in any node's NLM is always accurate because of using IDOTM algorithm. Moreover, the unique benefit of using FLDRE, which adapts the selection process based on FLC, appears clearly as the traffic load increases. FLDRE gives GSAR the ability to handle the traffic load and to distribute it evenly all over the network. Also, because of using DRESM, a packet in transit is prevented from

reaching a dead end or hole area. Thus, the routed packets correctly reach their final targets through the close-to-optimum path with very low end-to-end delay.

5.3.3.4 Routing Path Stretch Vs. Number of Data Traffic Sources

Figure 5.19 shows the average of routing path stretch, i.e, path length (RPS) that represents the routing path average hops count, in GPSR, POR and GSAR protocols, as a function of number of data traffic sources in the network. A better protocol is the one that has less RPS.

This simulation shows that with low and high number of data traffic, GPSR and POR protocols perform worse than GSAR. A comparison in Figure 5.19 shows that the GSAR protocol reduces the average routing path stretch by about 51.39% and 46.58% over both GPSR and POR protocols respectively. GSAR achieves better improvement in the number of travelled hops. Again, CUT and NMEM contribute to maintaining up-to-date neighbours' information in a node's NLM in GSAR strategy. This leads to selecting a more suitable next-relay routing neighbour from the sender's NLM.

In GPSR and POR protocols, due to using fixed number of nodes, this means that encountering hole problem is high. As the number of data traffic increases, more source-destination pairs being involved in the routing process, which yields more packets to face more non-shortest (i.e., non-optimal) route hops. The simulation results in Figure 5.19 show that GPSR is worse than other routing protocols. This is because as more and more data packet injected in the network, the number of data packets that use the non-shortest path increases as well for the reason explained in figure 5.15.

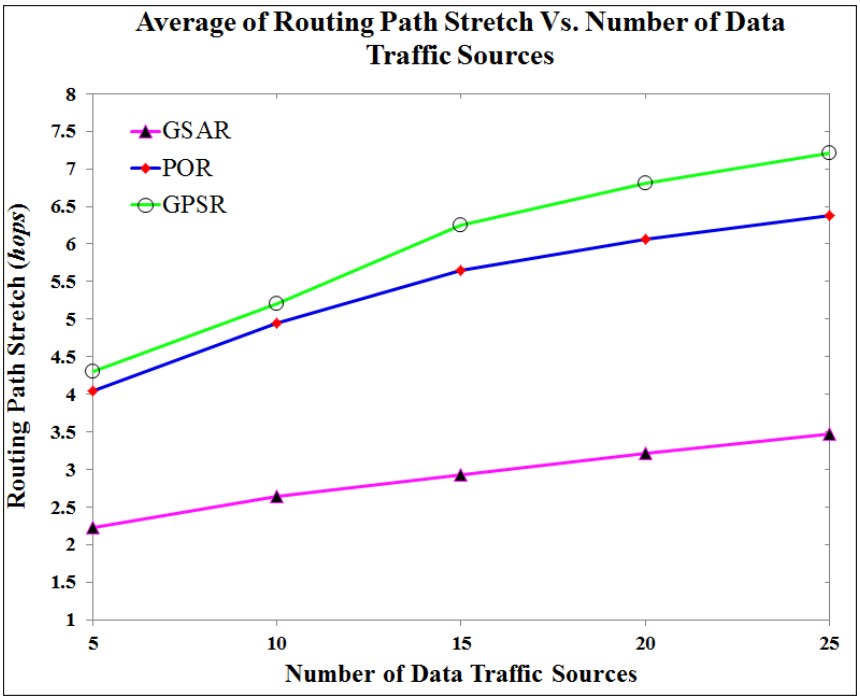


Figure 5.19. Average Routing Path Stretch via Number of Data Traffic Sources

5.3.3.5 Inconsistency of Neighbourhood Matrix Vs. Number of Data Traffic Sources

Figure 5.20 shows the average of the inconsistency of neighbourhood matrix (INM) ratio in GPSR, POR and GSAR protocols, as a function of the number of the data traffic sources. A better protocol is the one that has less INM. A comparison in Figure 5.20 shows that the GSAR protocol reduces the average inconsistency of neighbourhood matrix ratio by about 63.35% and 61.9% over GPSR and POR respectively.

For all protocols, as shown in Figure 5.20, as the number of the data traffic sources increases, the inconsistency of neighbourhood matrix rate slightly decreases. This is because the node speed in this simulation is fixed, thus, the compulsory update technique is rarely activated in GSAR. As a consequence, the CUT algorithm in GSAR contributes to a fixed level in correcting the entries' information in nodes' NLM. The situation is worse with the GPSR and POR protocols because the beacon frequency only relies on a pre-specified interval time, independent of the number of CBR pairs.

The success and effectiveness of GSAR is attributed mainly to its new features that are NMEM in DBUM and RTF\CTF\DATA in IDOTM techniques. In GSAR, NMEM

technique contributes in keeping up-to-date information in nodes' NLMs. In addition, in IDOTM technique, the information of the sender nodes is piggybacked on the header of all communicated DATA, RTF and CTF packets. In such a case, as the sent traffic increases, the accuracy of the information in nodes' NLM increases. For both the GPSR and POR protocols, the information of the sender nodes only piggybacks on DATA packets. In addition, the lifetime of entries only relies on the pre-specified interval time. Thus, the improvement in nodes' NLMs is lower than that in GSAR.

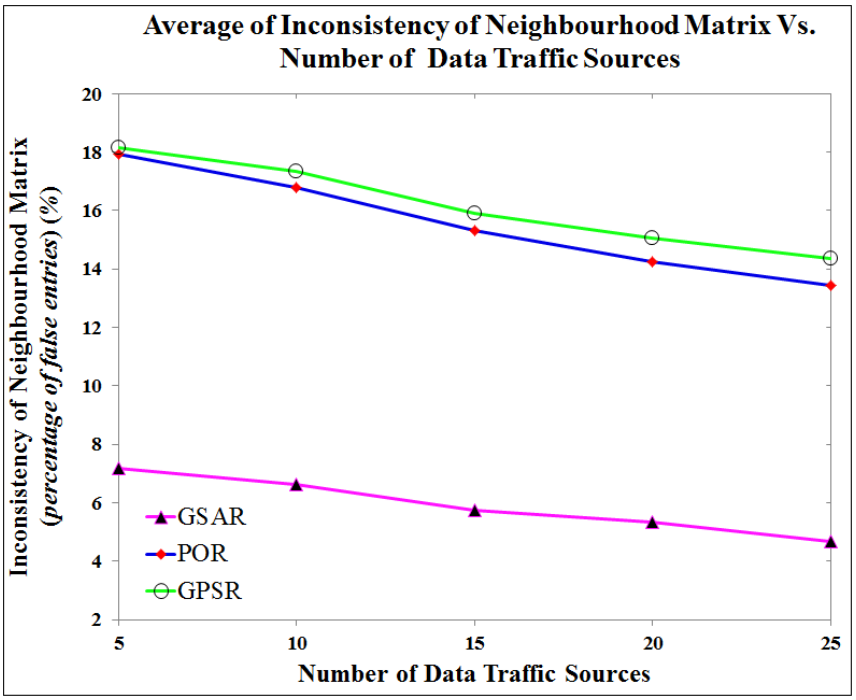


Figure 5.20. Average Inconsistency of Neighbourhood Matrix Ratio via Number of Data Traffic Sources

5.3.3.6 Number of Hole Problem Occurrence Vs. Number of Data Traffic Sources

Figure 5.21 shows the average numbers of the packets encountering a hole problem (NHPO) in GPSR, POR and GSAR as the function of number of the data traffic sources in the network. A better protocol is the one that has less NHPO. As the amount of data traffic increases, the number of DATA packets encountering hole problem remains static for all protocols because the hole problem occurrence is independent on the amount of data traffic. A comparison in Figure 5.21 shows that the GSAR protocol reduces the average number of hole problem occurrence by about 88.94% and a 83.7% over GPSR and POR respectively.

This simulation shows that in low and high amounts of the data traffic sources in the network, the GPSR and POR protocols perform worse than GSAR. The high success rate of GSAR is attributed mainly to the DRESM functionality, in which a sender node does not forward a data packet for any node located in a hole area by the means of using the four handshaking messages in IDOTM technique.

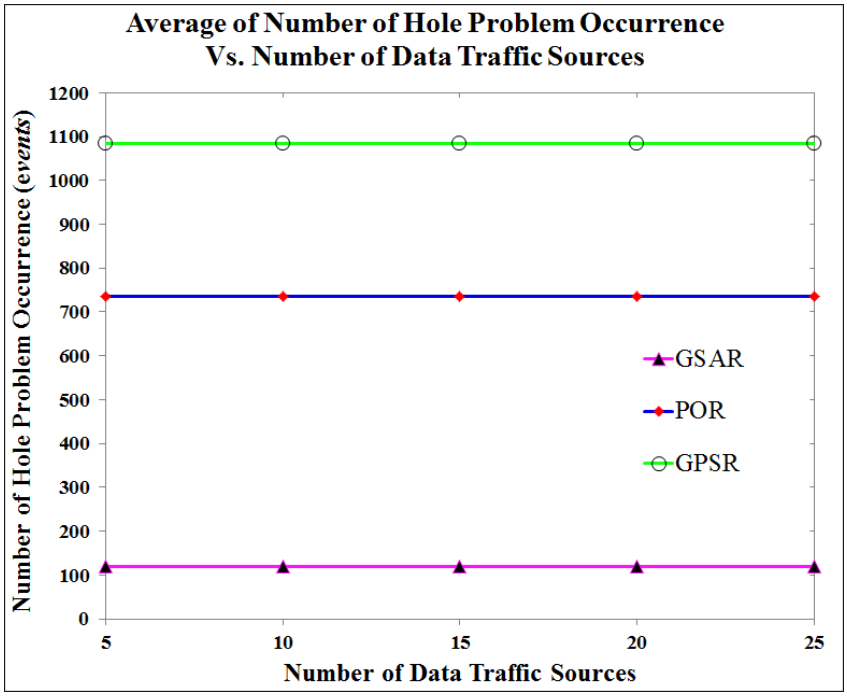


Figure 5.21. Average Number of Hole Problem Occurrence via Number of Data Traffic Sources

5.3.3.7 Summary Results of Varying Number of Data Traffic Sources

This section summarizes the average performance results of GSAR, POR and GPSR routing protocols under various number of data traffic sources. The results in Table 5.6 indicate that varying the number of data traffic sources affects the overall performance of GSAR, POR and GPSR routing protocols. This is because more and more amounts of data traffic will be injected in the network. This requires more effort from the participating nodes to process the new increment of data packets, which might have a bad impact on the performance of the routing protocols. In this simulation, as the number of sources increases the performance of GSAR, POR and GPSR routing protocols decreases.

Table 5.6 below indicates that GSAR routing protocol can perform well under high number of sources and outperforms POR and GPSR protocols in terms of PDR, COH, EED, INM, RPS and NHPO for the reasons discussed in the previous sections.

Table 5.6. The Performance Comparison between GSAR, POR and GPSR vs. Number of Data Traffic Sources

Metric	Protocol	Minimum	Maximum	Mean	GSAR Improvement
	GSAR	94.786	97.631	96.346	-
PDR	**POR**	80.748	90.934	86.864	10.91 %
	GPSR	77.66	87.752	83.3	15.66 %
	GSAR	9329	10719	10123	-
COH	**POR**	21886	22330	22107	-54.201%
	GPSR	21886	22330	22107	-54.201%
	GSAR	48.154	98.871	74.2179	-
EED	**POR**	84.601	155.143	119.629	-37.626%
	GPSR	103.32	208.456	154.255	-51.627%
	GSAR	2.22	3.47	2.89	-
RPS	**POR**	4.05	6.38	5.41	- 46.58%
	GPSR	4.3	7.21	5.95	-51.39%
	GSAR	4.689	7.196	5.922	-
INM	**POR**	13.447	17.949	15.546	-61.906%
	GPSR	14.353	18.14	16.163	-63.36%
	GSAR	120	120	120	-
NHPO	**POR**	736.29	736.29	736.29	-83.702%
	GPSR	1085.4	1085.4	1085.4	-88.944%

GSAR improvement $= \frac{GSAR\,mean - GPSR\,or\,POR\,mean}{GPSR\,or\,POR\,mean} \times 100\%$, Positive values means increment in the percentage and negative values means reduction in the percentage

5.4 Greedy Standalone Routing GSAR Protocol Features

This research proposed an improved position-based routing protocol GSAR. From the performance evaluation results it can be concluded that this research succeeded in improving GFS as a standalone position-based routing protocol for MANET. Also, the performance evaluation results showed that GSAR is superior over other unicast position-based routing protocols. In sum, GSAR has the following characteristics.

i. GSAR is a beacon-based unicast position-based routing protocol for MANET and it is not coupled with any kind of recovery approaches.
ii. GSAR is a localized position-based routing protocol. It is stateless in the sense that it does not need global knowledge.
iii. The proposed GSAR is a dynamic routing protocol that not only finds a route between the communicating nodes, but also responds swiftly to the mobility's rapid topology changes and optimally uses limited resources (e.g., power and bandwidth).
iv. GSAR uses a modified, reliable dynamic beaconing update scheme and fuzzy controller, the distribution of fresh node status information. Thus, it reduces the frequency of sending control messages compared to other routing protocols.
v. GSAR is a multi-hop and multi-objective routing protocol. It introduces forwarding decisions and actions at each source\forwarder node based on the reliability degree of each neighbour as calculated from local information. Thus, the selection process is distributed and not bound to the source nodes. The simulation results revealed that GSAR as a load-balanced routing protocol works much better than other shortest path routing protocols. Simultaneously, GSAR produced paths noticeably shorter than the paths produced by other routing protocols when they used a recovery mode.
vi. The reliability of the next-hop uses the notion of multi-metrics approach. Hence, GSAR can construct a reliable route between communicating nodes, which is close to optimum.
vii. The GSAR algorithm has a proper reliable and dynamic mechanism to remove stale information from a node's NLM at a very low cost.
viii. In case the optimal neighbour is unavailable in a node's transmission range,

GSAR can continue to work based on the carry-move mode.

ix. Clock synchronization among nodes is essential in executing GSAR operations.

x. From implementation and evaluation results, GSAR achieves very high delivery rates for dense as well sparse networks under various mobility values. The DATA packets are routed reliably to the ultimate destination node with a lower delay. GSAR also expends minimal COH; thus it is considered to be bandwidth-efficient protocol in which the remaining bandwidth is saved for actual data communication. Also, GSAR can construct paths between communicating nodes, which are close to optimum.

5.5 Advantages of Dynamic Beaconing-Update Mechanism

From the implementation and evaluation results, the actual utility of CUT in DBUM becomes clearer under different network settings. By using DFLCH with CUT, the maximum time period that can transpire can be optimized adaptively before a node broadcasts UBM. DFLCH controls CHT period scheduling. Optimization of CHT directly affects the number of sent UBMs. Thus, CUT helps to get a clearer view of the more recent local network topology changes. Consequently, CUT increases the efficiency of position-based routing.

CUT has the ability to overcome the non-optimal problem discussed in Chapter 2 by avoiding sending packets to outdated neighbours' location information by using the NMEM technique. By solving this problem, the packet's end-to-end delay, as well overhead, are decreased. Besides, by performing CUT, a packet loss is minimized, due to solving the inconsistency problem in a node's NLM. The only disadvantage of using CUT is that performing the checking process may consume more energy.

NMEM is very effective under different network settings. By applying NMEM, packet loss will be minimized, due to solving the inconsistency problem in a node's NLM; this results in decreasing the broken links along the path. Also, the NMEM technique reacts fast enough to very high topological changes, by adjusting ELT based on the increment in mobility values.

5.6 Advantages of the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism

The implementation and evaluation results suggest that in IDOTM, the RTF/CTF/ DATA messages can act as BP; thus using IDOTM reduces UBM sending rate. Also, IDOTM keeps the information in the nodes' NLM maximally fresh in regions under different network settings. Another main advantage of using IDOTM, through exchange RTF\CTF messages, is that IDOTM can perform as a two-hops approach-like. Thus, it improves the accuracy of neighbours' status information that increases the opportunity to avoid holes occurring in the routing path. Additionally, by using the principle of candidate node can only send back CTF message, refrain from sending many responses that may contribute to a congestion problem. RTF/CTF/DATA provides the sender node with full information about its neighbours' status. Thus, it can easily select the most optimal one amongst others.

The FLDRE technique is a fully distributed approach in which each node only needs the proposed routing metrics to make a routing decision. The optimal next relay-node is determined dynamically by using the FLDRE approach. The FLDRE can determine the reliability degree of the candidate's neighbours. Therein, a reliable route is constructed from most reliable nodes, and the packet is reliably delivered at the destination side. Simultaneously, the constructed path is a hole free-optimal route. Therefore, the proposed algorithm in this basic form offers an effective solution to GFS failure as a primary routing protocol.

5.7 Summary of Chapter

In this chapter, the performance of the proposed GSAR is evaluated. The evaluation was accomplished by comparing GSAR's performance with two different carefully chosen protocols and in three different simulation settings. The results show that GSAR outperforms both selected protocols in terms of six performance metrics in the following aspects:

i. GSAR can prevent the geographical hole problem from occurring in the path of

the forwarded packet, and efficiently uses all network's population in data routing based on multi-objectives.

ii. Nodes with GSAR have very accurate information in their NLMs.

iii. GSAR achieved a high packet delivery rate using the close to optimum path.

iv. GSAR achieved the lowest routing overheads; thus, the available bandwidth to benefit the real data packet is increased.

v. GSAR achieved the lowest end-to-end delay; thus, the routed packet reaches its final destination in timely manner.

The next chapter presents the conclusion, also it presents some future scope that are suggested to extend this research..

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Overview

This chapter draws conclusions about the work with respect to this research's objectives. Also, this chapter presents the contributions and the significance of these contributions. Finally, this chapter suggests possible future research directions.

6.2 Summarization

1. Frequent mobility's topology changes and limited resources in the Mobile Ad hoc Network (MANET) impose great challenges for developing efficiently the Greedy Forwarding Strategy (GFS) algorithm in MANET. Several important factors effect the proper operation of the GFS function when it is used in MANET. These factors mainly include the conditions of the participating nodes, their mobility attributes, and the accuracy of the participants' position information.
2. The shortest path adoption in terms of the lowest hop counts in GFS is likely to have a detrimental effect on position-based routing performance in MANET. To reduce GFS failure in MANET, several solutions have been proposed, including recovery strategies, beaconing update mechanisms, prioritization, and selection processes based on the adoption of additional routing metrics. However, no method has improved GFS enough to eliminate completely its failure in MANET. Instead, only a few solutions for GFS failure in MANET have been achieved, and each solution method still suffers from drawbacks yet to be solved.
3. This research's concerted efforts look to improve GFS and make GFS compatible with MANET features, eliminating the reasons behind GFS failure. The improvement of GFS has been achieved by means of an adaptive beaconing update mechanism and an effective prioritization and selection process mechanism. The frequent mobility's topology changes and limited resources of MANET were considered when building the improved GFS. Hence, the conditions of the participating nodes, their mobility-based attributes, and the nodes' position

accuracy were considered.

4. The research flow adopted three stages. These were: the design, the implementation, and the evaluation. These three stages led accomplishing the research objectives. The output of the design stage was the Greedy Standalone Position-based Routing (GSAR) protocol, featuring its two incorporated mechanisms. The overall aim of the design stage was designing a routing protocol that alleviates GFS failure in MANET and eliminates the need for using a recovery mode. The implementation of GSAR, with its two incorporated mechanisms, was accomplished in a simulated environment with an extensive validation process. The evaluation of the proposed protocol was explicated highly by comparing the GSAR protocol with other, carefully selected proposed position-based routing protocols.
5. The evaluation results of GSAR showed great improvements over current position-based routing protocols in several aspects. The results showed that GSAR utilizes fully the network's limited resources and swiftly responds to the mobility-based topology changes. Also, GSAR eliminates the need for using a recovery mode. This achievement makes GSAR a reliable, efficient and reasonable alternative position-based routing protocol in MANET. In completing the GSAR performance demonstration, the objectives of this research were satisfied.

6.3 Contributions

The major contributions of this thesis were the design of two new mechanisms and a new routing protocol. The two new mechanisms are the Dynamic Beaconing Update Mechanism (DBUM) and the Dynamic and Reactive Reliability Estimation with Selective Metrics Mechanism (DRESM). The new protocol is the Greedy Standalone Position-based Routing (GSAR) protocol.

1. DBUM is an efficient extension to the current beaconing update used in GFS. The DBUM overcomes the drawbacks of current beaconing extensions to GFS in MANET. The DBUM reduces routing packet overhead and increases the accuracy of nodes' status information in a highly mobile environment. The DBUM significantly improves information accuracy in a node's NLM, giving a node the

ability to make a reliable routing decision. Making a reliable routing decision leads to increased packet delivery and reduced communication cost. The DBUM comprises two techniques that are minor contributions of this research. These are:

a) The Compulsory Update Technique (CUT). The CUT optimally reacts fast to the node's mobility-based attribute changes. The CUT can trade off between routing packet overhead and the accuracy of the nodes' status information in the network.
b) The Neighbourhood Matrix Entries Management (NMEM) technique. The NMEM aims to adapt dynamically the neighbours' entries lifetime (ELT) in a node's Neighbourhood's Location-Matrix (NLM). The NMEM allows reliable and timely removal of neighbours' entries based on the Residual Links Lifetime (RLT) value. NMEM provides more accurate information in nodes' NLM matrix that helps in improving packet delivery.

2. DRESM is an efficient extension to the prioritization and selection of the next relay-node process used in the current GFS. The DRESM overcomes the drawbacks of the current prioritization and selection process extensions to GFS in MANET. The DRESM controls the out-going traffic and makes reliable forwarding decisions based on multi-routing metrics. The DRESM significantly increases the efficiency of delivering the packet to the destination using the closest-to-optimum path that results in reducing the communication cost. The DRESM comprises two techniques that are two minor contributions. These are:

   a) The Status Information Distribution and Outgoing Traffic Control Management (IDOTM). The IDOTM technique distributes status information between communicating nodes and controls the outgoing traffic of the network. As a consequence, every node in the network can get reactively comprehensive and receive accurate information about its neighbours. The IDOTM technique optimally reacts fast to send DATA packets between communicating nodes; thus, the DATA packets are routed reliably and their loss probability is minimized.
   b) The Fuzzy Logic Dynamic Nodes' Reliability Estimation (FLDRE) tech-

nique. The FLDRE dynamically evaluates the reliability index of a node's candidate neighbours based on multi-routing metrics. Therein, a reliable (optimal) route is constructed by using the most reliable nodes. Due to the construction of a reliable route between the communicating nodes, the DATA packet is delivered reliably to the destination side. In the FLDRE technique, the retransmission and rerouting process is decreased because of the reduction in packet loss that, in turn, leads to reduced communication cost.

3. GSAR is a reliable and efficient routing protocol that is build based on the current GFS and the two new mechanisms DBUM and DRESM. Both mechanisms were incorporated in the conventional GFS to form the proposed GSAR. DBUM and DRESM complement each other, creating good synergy. Thus, both reinforce GSAR. GSAR optimizes packet loss, routing packet overhead, end-to-end delay, routing path stretch, and reduces inconsistency of a neighbour's entry in a node's neighbours-list and the occurrence of dead-end problem. These are critical disadvantages in GFS performance and current extensions to GFS in MANET.

6.4 Significance of the Contributions

The proposed GSAR protocol outperforms the current position-based routing protocol. This is because GSAR has new features that make GFS compatible with MANET features. Due to DBUM and DRESM, GSAR can build a reliable path between communicating nodes reducing network communication costs. Therefore, GSAR would be beneficial to MANET applications such as those for the military, mobile conferencing, and emergency services.

This research constructs the proposed DBUM and DRESM mechanisms in such a manner that both can efficiently and effectively satisfy MANET's features. DBUM can dynamically ensure and maintain up-to-date neighbours' status information in a node's neighbours-matrix. The DBUM can balance between the beacon sending frequency and control overhead. Thus, the retransmission and rerouting required for inaccurate position information that increases packet loss probability can be prevented. Hence,

DBUM could be beneficial as a general beaconing-update approach with any position-based routing protocol. DRESM can dynamically select the next relay-node based on selective routing metrics to achieve multi-routing objectives. Also, the DRESM can distribute the load traffic in a highly mobile network. Thus, retransmission and rerouting required in the case of packet loss are decreased, and DRESM could be beneficial as a general forwarding strategy that should work with any position-based routing protocol

6.5 Future Work

This section summarizes potential research areas for future work.

1. The researcher strongly believes that GSAR is ready for implementation. Thus, research about the actual deployment of GSAR can be carried out.
2. The MANET is more susceptible to physical security risks than other networks. In GSAR, a node's neighbours can eavesdrop the on-going traffic. Investigation in the context of security for GSAR routing protocol would guard against attacks, such as spoofing and generating deceptive routing messages.
3. Because the scope of this research is limited to improving the routing process in position-based routing protocol, investigating the GSAR protocol using a modified location service that can reduce significantly MANET overheads would be beneficial.
4. Lastly, the GSAR algorithm can be verified via more mobility models in Ns2. This is because using GSAR with different applications in MANET may require a different mobility model to mimic participant movement in the application used.

REFERENCES

[1] S. Ilarri, E. Mena, and A. Illarramendi, "Location-dependent Query Processing: Where we are and where we are Heading," *ACM Computing Surveys*, vol. 42, no. 3, pp. 12:1–12:73, Mar. 2010. [Online]. Available: http://doi.acm.org/10.1145/1670679.1670682.[Accessdate10/12/2013]

[2] J.-Z. Sun, "Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing," in *the Proceedings of International Conferences on Infotech & Infonet*, Beijing, China, 29 Oct -01 Nov 2001, pp. 316–321. [Online]. Available: http://dx.doi.org/10.1109/ICII.2001.983076.[Accessdate10/12/2013]

[3] F. Magnus, J. Per, and L. Peter, "Wireless Ad Hoc Networking-The Art of Networking without A Network," *Ericsson Review*, vol. 77, no. 1, pp. 248–262, 2000, ISSN: 0014-0171. [Online]. Available: http://www.citeulike.org/user/sergiocabrero/article/1943206-AT.[Accessdate10/12/2013]

[4] D. Bein, "Self-Configuring, Self-Organizing, and Self-Healing Schemes in Mobile Ad Hoc Networks," in *Guide to Wireless Ad Hoc Networks*, S. Misra, I. Woungang, and S. C. Misra, Eds. Springer-Verlag London Limited, 2009, ch. 2, pp. 27–42. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-328-6_2.

[5] S. Sesay, Z. Yang, and J. He, "A Survey on Mobile Ad Hoc Wireless Network," *Information Technology Journal*, vol. 3, no. 2, pp. 168–175, 17-19, Mar. 2004. [Online]. Available: manet.eurecom.fr/Sesay.pdf.[Accessdate10/12/2013]

[6] P. Mohapatra and S. Krishnamurthy, "Ad Hoc Networks, Emerging Applications, Design Challenges and Future Opportunities," in *Ad Hoc Networks: Technologies and Protocols*. Springer, Oct. 2010, pp. 1–22. [Online]. Available: http://books.google.com.my/books?id=JhQIkgAACAAJ

[7] A. Boukerche, D. Camara, A. Loureiro, and C. Figueiredo, "Algorithms for Mobile Ad Hoc Networks," in *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks*, A. Boukerche, Ed. John Wiley & Sons, Inc., 2009, ch. 1, pp. 1–20, ISBN 978-0-470-38358-2. [Online]. Available: http://dx.doi.org/10.1002/9780470396384.ch1.

[8] I. Chlamtac, M. Conti, and J. J. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1016/S1570-8705(03)00013-1. [Accessdate10/12/2013]

[9] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, "An Overview of Mobile Ad Hoc Networks: Applications and Challenges," *Journal Communications Network*, vol. 3, no. 3, pp. 60–66, Sep. 2004. [Online]. Available: cwi.unik.no/images/Manet_Overview.pdf.[Accessdate10/12/2013]

[10] J. Jain, M. Fatima, R. Gupta, and K. Bandhopadhyay, "Overview and Challenges of Routing Protocol and MAC Layer in Mobile Ad-Hoc Network," *Journal of Theoretical and Applied Information Technology*, vol. 8, no. 1, pp. 6–12, Oct. 2009. [Online]. Available: www.jatit.org/volumes/research-papers/Vol8No1/2Vol8No1.pdf.[Accessdate10/12/2013]

[11] M. K. Marina and S. R. Das, "Routing in Mobile Ad Hoc Networks," in *Ad Hoc Networks Technologies and Protocols*, P. Mohapatra and S. V. Krishnamurthy, Eds. Springer Science + Business Media, Inc.: New York, NY, USA, 2005, ch. 3, pp. 63–90. [Online]. Available: http://dx.doi.org/10.1007/0-387-22690-7_3.

[12] V. Narsimha, B. Sujatha, and T. SampathKumar, "A Survey of Wireless Mobile Ad-Hoc Networks (MANET)," *International Journal of Science and Advanced Technology*, vol. 1, no. 5, pp. 189–192, Jul. 2011. [Online]. Available: http://www.ijsat.com/pdf.php?pdf_id=98.pdf.

[13] Z. Ren and W. Guo, "Unicast Routing in Mobile Ad Hoc Networks: Present and Future Directions," in *the Proceedings of the Fourth International Conference In Parallel and Distributed Computing, Applications and Technologies*. Chengdu, Sichuan, China: IEEE, 27-29 Aug. 2003, pp. 240–243. [Online]. Available: http://dx.doi.org/10.1109/PDCAT.2003.1236318.

[14] L. K. Qabajeh, L. M. Kiah, and M. M. Qabajeh, "A Qualitative Comparison of Position-Based Routing Protocols for Ad-Hoc Networks," *International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 131–140, Feb. 2009. [Online]. Available: http://eprints.um.edu.my/id/eprint/4945.[Accessdate10/12/2013]

[15] D. Chen and V. Pramod K., "Geographic Routing in Wireless Ad Hoc Networks," in *Guide to Wireless Ad Hoc Networks*. Springer, 2009, ch. 7, pp. 157–181. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-328-6_7.[Accessdate10/12/2013]

[16] M. L. M. Kiah, L. K. Qabajeh, and M. M. Qabajeh, "Unicast Position-based Routing Protocols for Ad-Hoc Networks," *Acta Polytechnica Hungarica*, vol. 7, no. 5, pp. 16–46, Oct.-Dec. 2010. [Online]. Available: http://www.uni-obuda.hu/journal/Issue26.htm.[Accessdate10/12/2013]

[17] C. Lemmon, S. Lui, and I. Lee, "Geographic Forwarding and Routing for Ad-Hoc Wireless Network: A Survey," in *the Proceeding of the 5th International Joint Conference on INC, IMS and IDC, (NCM'09)*, Myongji Univ., Yongin, South Korea, 25-27 Aug. 2009, pp. 188–195. [Online]. Available: http://dx.doi.org/10.1109/NCM.2009.80.

[18] A. Maghsoudlou, M. St-Hilaire, and T. Kunz, "A Survey on Geographic Routing Protocols for Mobile Ad hoc Networks," Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, Tech. Rep. 3, Oct. 2011. [Online]. Available: www.csit.carleton.ca/~msthilaire/Tech_Report/2011-GRPreport.pdf.[Accessdate10/12/2013]

[19] S.-H. Cha, M.-W. Ryu, and K.-H. Cho, "A Survey of Greedy Routing Protocols for Vehicular Ad Hoc Networks," *Smart Computing Review*, vol. 2, no. 2, pp. 125–137, Apr. 2012. [Online]. Available: http://dx.doi.org/10.6029/smartcr.2012.02.003.[Accessdate10/12/2013]

[20] A. M. Popescu, G. Tudorache, B. Peng, and A. H. Kemp, "Surveying Position Based Routing Protocols for Wireless Sensor and Ad-hoc Networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 1, pp. 41–67, Apr. 2012. [Online]. Available: www.ijcnis.org/index.php/ijcnis/article/download/85/90.

[21] C. Lemmon, S. M. Lui, and I. Lee, "Review of Location-Aware Routing Protocols," *Advances in Information Sciences and Service Sciences (AISS)*, vol. 2, no. 2, pp. 132–143, Mar. 2010. [Online]. Available: http://dx.doi.org/10.4156/aiss.vol2.issue2.15.

[22] F. Araujo, L. Rodrigues, F. Araujo, and L. Rodrigues, "Survey on Position-Based Routing1," University of Lisbon, Tech. Rep. Jan., 2006. [Online]. Available: http://eden.dei.uc.pt/~filipius/publications.html.

[23] G. G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," University of Southern California, Marina del Rey. Information Sciences Inst., Tech. Rep., Mar. 1987. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a180187.pdf.[Accessdate10/12/2013]

[24] S. Ruhrup, "Position-Based Routing Strategies," Ph.D. Dissertation, Faculty of Electrical Engineering, Computer Science and Mathematics, University of Paderborn, Jul. 2006. [Online]. Available: www2.cs.uni-paderborn.de/cs/ag-madh/WWW/sr/thesis.pdf.[Accessdate10/12/2013]

[25] G. Xing, C. Lu, R. Pless, and Q. Huang, "On Greedy Geographic Routing Algorithms in Sensing Covered Networks," in *the Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Roppongi Hills, Tokyo, Japan: ACM, 24-26 May 2004, pp. 31–42. [Online]. Available: http://doi.acm.org/10.1145/989459.989465.[Accessdate10/12/2013]

[26] S. Ruhrup, "Theory and Practice of Geographic Routing," Department of Computer Science, University of Freiburg, Germany, Tech. Rep., Feb. 2009. [Online]. Available: http://hondo.informatik.uni-freiburg.de/people/ruehrup/georouting-chapter-draft.pdf.[Accessdate10/12/2013]

[27] J. Li and S. M. Shatz, "Toward Using Node Mobility to Enhance Greedy-Forwarding in Geographic Routing for Mobile Ad Hoc Networks," in *the Proceeding of the International Workshop on Mobile Device and Urban Sensing (MODUS 08)*, St. Louis, MO, USA, 21 Apr. 2008, pp. 1–8. [Online]. Available: www.cs.uic.edu/~shatz/papers/modus08.pdf.[Accessdate10/12/2013]

[28] N. Arad and Y. Shavitt, "Minimizing Recovery State in Geographic Ad Hoc Routing," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 203–217, Feb. 2009. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TMC.2008.86.[Accessdate10/12/2013]

[29] P. J. Wan, C. W. Yi, L. Wang, and X. Yao, F.and Jia, "Asymptotic Critical Transmission Radii for Greedy Forward Routing in Wireless Ad Hoc Networks," *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1433–1443, May 2009. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2009.05.070307.

[30] F. Cadger, K. Curran, J. Santos, and S. Moffett, "A Survey of Geographical Routing in Wireless Ad-Hoc Networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 621–653, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1109/SURV.2012.062612.00109.

[31] M. Lukic, B. Pavkovic, N. Mitton, and I. Stojmenovic, "Greedy Geographic Routing Algorithms in Real Environment," in *the Proceeding of the 5th International Conference on Mobile Ad-hoc and Sensor Networks*. Wu Yi Mountain, China: IEEE, 14-16 Dec. 2009, pp. 86–93. [Online]. Available: http://dx.doi.org/10.1109/MSN.2009.11.[Accessdate10/12/2013]

[32] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," *Ad Hoc Wireless Networking*, vol. 8, no. 64, pp. 1–20, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.3076.[Accessdate10/12/2013]

[33] A. K. Haboush, M. A. Nabhan, M. Al-Tarazi, and M. Al-Rawajbeh, "Load Balancing Using Multiple Node Disjoint Paths," *Computer and Information Science*, vol. 5, no. 3, pp. 83–89, May 2012. [Online]. Available: http://dx.doi.org/10.5539/cis.v5n3p83.[Accessdate10/12/2013]

[34] D. S. J. D. Couto, D. Aguayo, B. A. Chambers, and R. Morrist, "Performance of Multihop Wireless Networks: Shortest Path is not Enough," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 83–88, Jan. 2003. [Online]. Available: http://doi.acm.org/10.1145/774763.774776.[Accessdate10/12/2013]

[35] G. I. Ivascu, S. Pierre, and A. Quintero, "QoS Routing With Traffic Distribution in Mobile Ad Hoc Networks," *Computer communication*, vol. 32, no. 2, pp. 305–316, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2008.10.012.

[36] S. Venkatasubramanian and N. Gopala, "A Quality of Service Architecture for Resource Provisioning and Rate Control in Mobile Ad Hoc Networks," *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, vol. 1, no. 3, pp. 106–120, Sep. 2010. [Online]. Available: http://dx.doi.org/10.5121/ijasuc.2010.1309.[Accessdate10/12/2013]

[37] F. De Rango and M. Tropea, "Energy Saving and Load Balancing in Wireless Ad Hoc Networks Through Ant-based Routing," in *the Proceeding of the International Symposium on Performance Evaluation of Computer & Telecommunication Systems*. Istanbul, Turkey: IEEE, 13-16 Jul. 2009, pp. 117–124. [Online]. Available: http://ieeexplore.ieee.org.www.ezplib.ukm.my/ielx5/5204040/5224098/05224137.pdf?tp=&arnumber=5224137&isnumber=5224098.

[38] S.-B. Lee and A. Campbell, "HMP: Hotspot Mitigation Protocol for Mobile Ad hoc Networks," in *the Proceeding of the Eleventh International Workshop on Quality of Service*, ser. Lecture Notes in Computer Science, K. Jeffay, I. Stoica, and K. Wehrle, Eds., vol. 2707. CA, USA: Springer Berlin Heidelberg, 2–4 Jun. 2003, pp. 266–283. [Online]. Available: http://dx.doi.org/10.1007/3-540-44884-5_15.

[39] V. Thilagavathe and Dr. K. Duraiswamy, "Cross Layer Based Congestion Control Technique for Reliable and Energy Aware Routing in MANET," *International Journal of Computer Applications*, vol. 36, no. 12, pp. 1–6, Dec. 2011. [Online]. Available: http://www.ijcaonline.org/archives/volume36/number12/4549-6288.[Accessdate10/12/2013]

[40] S. Sheeja and R. V. Pujeri, "Effective Congestion Avoidance Scheme for Mobile Ad Hoc Networks," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 5, no. 1, pp. 33–40, 2013. [Online]. Available: www.mecs-press.org/ijcnis/ijcnis-v5-n1/IJCNIS-V5-N1-4.pdf.[Accessdate10/12/2013]

[41] G. Wang, G.and Wang, "An Energy-Aware Geographic Routing Protocol for Mobile Ad Hoc Networks," *International Journal of Software and Informatics*, vol. 4, no. 2, pp. 183–196, Jun. 2010. [Online]. Available: GWang,GWang-InternationalJournalofSoftwareandInformatics,2010-st.ewi.tudelft.nl.[Accessdate10/12/2013]

[42] Y. Wang, "Study on Energy Conservation in MANET," *Journal of Networks*, vol. 5, no. 6, pp. 708–715, Jun 2010. [Online]. Available: http://dx.doi.org/10.4304/jnw.5.6.708-715.[Accessdate10/12/2013]

[43] D. G. Anand, H. G. Chandrakanth, and M. N. Giriprasad, "Energy Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," *International Journal of Advanced Networking and Applications*, vol. 3, no. 1, pp. 999–1005, Jul./Aug. 2011. [Online]. Available: http://dx.doi.org/10.5121/acij.2011.2204.[Accessdate10/12/2013]

[44] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *the Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Boston, Massachusetts, US: ACM, 6-11 Aug. 2000, pp. 243–254. [Online]. Available: http://doi.acm.org/10.1145/345910.345953.[Accessdate10/12/2013]

[45] S. Yang, C. K. Yeo, and B.-S. Lee, "Toward Reliable Data Delivery for Highly Dynamic Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 111–124, Jan. 2012. [Online]. Available: http://dx.doi.org/10.1109/TMC.2011.55.

[46] R. Alsaqour, M. Shanudin, M. Ismail, and M. Abdelhaq, "Analysis of Mobility Parameters Effect on Position Information Inaccuracy of GPSR Position-Based MANET Routing Protocol," *Journal of Theoretical and Applied Information Technology*, vol. 28, no. 2, pp. 114–120, Jun. 2011. [Online].

Available: www.jatit.org/volumes/research-papers/Vol28No2/8Vol28No2.pdf. [Accessdate10/12/2013]

[47] D. Son, A. Helmy, and B. Krishnamachari, "The Effect of Mobility-Induced Location Errors on Geographic Routing In Mobile Ad Hoc Sensor Networks: Analysis and Improvement Using Mobility Prediction," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 233–245, Jul. 2004. [Online]. Available: http://dx.doi.org/10.1109/TMC.2004.28.[Accessdate10/12/2013]

[48] J. Tsumochi, K. Masayama, H. Uehara, and M. Yokoyama, "Impact of Mobility Metric on Routing Protocols for Mobile Ad Hoc Networks," in *the Proceeding of the IEEE Pacific Rim Conference on Communications Computers and signal Processing (PACRIM)*, University of Victoria, Victoria, BC, Canada, 28-30 Aug. 2003, pp. 322–325. [Online]. Available: http://dx.doi.org/10.1109/PACRIM.2003.1235782.

[49] V. C. Giruka and M. Singhal, "Hello Protocols for Ad-Hoc Networks: Overhead and Accuracy Tradeoffs," in *the 6th IEEE International Symposium on a World of Wireless Mobile Multimedia Networks, (WoWMoM)*, Taormina-Giardini Naxos, 13-16 Jun. 2005, pp. 354–361. [Online]. Available: http://dx.doi.org/10.1109/WOWMOM.2005.50.

[50] F. Ingelrest, N. Mitton, and D. Simplot-Ryl, "A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks," in *the Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. Istanbul, Turkey: IEEE, 24-26 Oct. 2007, pp. 9–14. [Online]. Available: http://dx.doi.org/10.1109/MASCOTS.2007.5

[51] C. Li, L. Zhu, C. Zhao, and H. Lin, "Hello Scheme for Vehicular Ad Hoc Networks: Analysis and Design," *Journal on Wireless Communications and Networking (EURASIP)*, vol. 2013, no. 1, p. 28, Feb. 2013. [Online]. Available: http://jwcn.eurasipjournals.com/content/2013/1/28.

[52] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, "Evaluating the Limitations of and Alternatives in Beaconing," *Ad Hoc Networks*, vol. 5, no. 5, pp. 558–578, Jul. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2006.03.002.[Accessdate10/12/2013]

[53] Q. Chen, S. S. Kanhere, and M. Hassan, "Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 3, pp. 4046–4051, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1109/TMC.2012.20.[Accessdate10/12/2013]

[54] S. Kwon and N. B. Shroff, "Geographic Routing in the Presence of Location Errors," *International Journal of Computer and Telecommunications Networking*, vol. 50, no. 15, pp. 2902–2917, Oct. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.11.008.[Accessdate10/12/2013]

[55] B. Leong, B. Liskov, and R. Morris, "Geographic Routing without Planarization," in *the Proceeding of the 3rd conference on Networked Systems*

Design and Implementation. San Jose, CA: ACM, 8-10 May 2006, pp. 1–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267680.1267705. [Accessdate10/12/2013]

[56] J. Na, D. Soroker, and C.-K. Kim, "Greedy Geographic Routing using Dynamic Potential Field for Wireless Ad Hoc Networks," *IEEE Communications Letters*, vol. 1, no. 3, pp. 243–245, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2007.061612.[Accessdate10/12/2013]

[57] R. H. Khokhar, M. A. Ngadi, M. S. Latiff, K. Z. Ghafoor, S. Ali, "Multi-criteria Receiver Self-Election Scheme for Optimal Packet Forwarding in Vehicular Ad hoc Networks," *International Journal of Computers, Communication and Control*, vol. 7, no. 5, pp. 865–878, Dec. 2012. [Online]. Available: journal.univagora.ro/download/pdf/639.pdf.[Accessdate10/12/2013]

[58] Z. G. Kayhan, A. B. Kamalrulnizam, and H. N. A., "A Novel Delay-and Reliability-Aware Inter-Vehicle Routing Protocol," *Network Protocols and Algorithms*, vol. 2, no. 2, pp. 66–88, Jul. 2010. [Online]. Available: http://dx.doi.org/10.5296/npa.v2i2.427.[Accessdate10/12/2013]

[59] J. Gong, C. Z. Xu, and J. Holle, "Predictive Directional Greedy Routing in Vehicular Ad hoc Networks," in *the Proceeding of the 27th International Conference on Distributed Computing Systems Workshops*. Toronto, Ont.: IEEE, 22-29 Jun. 2007, pp. 2–10. [Online]. Available: http://dx.doi.org/10.1109/ICDCSW.2007.65.[Accessdate10/12/2013]

[60] Z. G. Kayhan, A. B. Kamalrulnizam, S. Shaharuddin, C. L. Kevin, M. M. Mohd, K. Maznah, and M. A. Marina, "Fuzzy Logic-assisted Geographical Routing over Vehicular Ad Hoc Networks," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 7, pp. 1–15, Jul. 2012. [Online]. Available: www.ijicic.org/11-05101-1.pdf.[Accessdate10/12/2013]

[61] D. Liarokapis and A. Shahrabi, "Fuzzy-based Probabilistic Broadcasting in Mobile Ad Hoc Networks," in *Wireless Days (WD)*. Glasgow Caledonian University, Glasgow, UK: IEEE, 10-12 Oct. 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WD.2011.6098185.

[62] R. Saqour, M. Shanudin, and M. Ismail, "Dynamic Beaconing for Ad Hoc Position-based Routing Protocol Using Fuzzy Logic Controller," in *the Proceedings of the International Conference on Electrical Engineering and Informatics (ICEEI)*, Bandung, Indonesia, 17-19 Jun. 2007, pp. 966–969. [Online]. Available: http://research.mercubuana.ac.id/proceeding/H-14.pdf. [Accessdate10/12/2013]

[63] D. Liarokapis and A. Shahrabi, "A Probability-Based Adaptive Scheme for Broadcasting in MANETs," in *the Proceedings of the 6th International Conference on Mobile Technology, Application and Systems*. Nice, France: ACM, 10-13 Sep. 2009, pp. 235–239. [Online]. Available: http://dx.doi.org/10.1145/1710035.1710081.

[64] A. K. Ahmad and N. Mitton, “Adapting Dynamically Neighbourhood Table Entry Lifetime in Wireless Sensor Networks,” in *the Preceding of the 10th International Conference on Wireless Communications and Signal Processing*, Univ. Lille 1, Lille, France, 21-23 Oct. 2010, pp. 1–6. [Online]. Available: {http://dx.doi.org/10.1109/WCSP.2010.5633707}.

[65] D. Chen and P. K. Varshney, “A Survey of Void Handling Techniques for Geographic Routing in Wirless Networks,” *IEEE Communications Surveys & Tutorials*, vol. 9, no. 1, pp. 50–67, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1109/COMST.2007.358971.

[66] J. Na, Y.-J. Kim, and R. Govindan, “Minimizing Recovery Overhead in Geographic Ad Hoc Routing,” *Computer Communications*, vol. 33, no. 11, pp. 1343–1353, Jul. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.03.021.

[67] B. Leong, S. Mitra, and B. Liskov, “Path Vector Face Routing: Geographic Routing with Local Face Information,” in *the Proceeding of the 13th IEEE International Conference on Network Protocols (ICNP ’05)*, Boston, MA, USA, 6-9 Nov. 2005, pp. 147–158. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2005.32.

[68] C. Jayapal and S. Vembu, “Performance Evaluation of Hole Avoidance Techniques in Geographic Forwarding for Mobile Ad Hoc Networks,” *International Journal of Information Technology Convergence and Services (IJITCS)*, vol. 1, no. 4, pp. 21–32, Aug. 2011. [Online]. Available: http://www.docshut.com/hohohohohohohohoi/Ijitcs.html.

[69] W. Kieb, H. Fubler, J. Widmer, and M. Mauve, “Hierarchical Location Service for Mobile Ad-Hoc Networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 4, pp. 47–58, Oct. 2004. [Online]. Available: http://doi.acm.org/10.1145/1052871.1052875.[Accessdate10/12/2013]

[70] M. Ayaida, H. Fouchal, L. Afilal, and Y. Ghamri-Doudane, “A Comparison of Reactive, Grid and Hierarchical Location-Based Services for VANETs,” in *the Proceeding of the IEEE Vehicular Technology Conference (VTC Fall)*. Québec City, Canada: IEEE, 3-6 Sept. 2012, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/VTCFall.2012.6398920.

[71] C. Liu and J. Kaiser, “A Survey of Mobile Ad Hoc network Routing Protocols,” University of Magdeburg, Tech. Rep. 8, Oct. 2005. [Online]. Available: http://vts.uni-ulm.de/docs/2005/5346/vts_5346.pdf.[Accessdate10/12/2013]

[72] N. Gupta and R. Gupta, “Routing Protocols in Mobile Ad-Hoc Networks: an Overview,” in *in the Proceedings of the International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT)*. Chennai: IEEE, 3-5 Dec. 2010, pp. 173–177. [Online]. Available: http://dx.doi.org/10.1109/INTERACT.2010.5706220.

[73] A. Boukerche, M. Z. Ahmad, D. Turgut, and B. Turgut, “A Taxonomy of Routing Protocols for Mobile Ad Hoc Networks,” in *Algorithms and Protocols*

for Wireless and Mobile Ad Hoc Networks. New Jersey: Wiley, 2009, ch. 5, pp. 129–164. [Online]. Available: http://dx.doi.org/10.1002/9780470396384.ch5.

[74] M. G. Rubinstein, I. M. Moraes, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "A Survey on Wireless Ad Hoc Networks," in *Mobile and Wireless Communication Networks (from International Federation for Information Processing (IFIP))*, ser. IFIP Advances in Information and Communication Technology, G. Pujolle, Ed. Springer US, Aug. 2006, vol. 211, ch. 1, pp. 1–33. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-34736-3.[Accessdate10/12/2013]

[75] N. N. Qadri and A. Liotta, "Analysis of Pervasive Mobile Ad Hoc Routing Protocols," in *Computer Communications and Networks, Pervasive Computing*. Springer, 2010, ch. 19, pp. 433–453. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-599-419.[Accessdate10/12/2013]

[76] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *the Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*. London, England UK: ACM, 31 Aug.-2 Sep. Aug. 31 - Sep. 02, 1994, pp. 234–244. [Online]. Available: http://dx.doi.org/10.1145/190809.190336.

[77] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, ser. The Kluwer International Series in Engineering and Computer Science, T. Imielinski and H. Korth, Eds. Springer US, 1996, vol. 353, ch. 5, pp. 153–181. [Online]. Available: http://dx.doi.org/10.1007/978-0-585-29603-6_5.[Accessdate10/12/2013]

[78] G. Kumar, Y. Reddyr, and M.Nagendra, "Current Research Work on Routing Protocols for MANET: A Literature Survey," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 3, pp. 706–713, May. 2010. [Online]. Available: www.enggjournals.com/ijcse/doc/IJCSE10-02-03-82.pdf.

[79] S. A. K. Al-Omari and P. Sumari, "An Overview of Mobile Ad Hoc Networks for the Existing Protocols and Applications," *International Journal on Application of Graph Theory in Wireless Ad Hoc Networks and Sensor Networks (Graph-Hoc)*, vol. 2, no. 1, pp. 87–110, Mar. 2010. [Online]. Available: arxiv.org/pdf/1003.3565.[Accessdate10/12/2013]

[80] S. Parul, K. Arvind, and T. Jawahar, "Performance Analysis of AODV, DSR and DSDV Routing Protocols in Mobile Ad-hoc Network (MANET)," *Journal of Information Systems and Communication*, vol. 3, no. 1, pp. 322–326, 2012. [Online]. Available: http://www.bioinfo.in/contents.php?id=45.[Accessdate10/12/2013]

[81] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," in *the Proceeding of the 21st Annual Joint Conference of*

the IEEE Computer and Communications Societies INFOCOM'02. New York, USA: IEEE, 23-27 Jun. 2002, pp. 1678–1687. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2002.1019421.[Accessdate10/12/2013]

[82] H. Fubler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "A Comparison of Routing Strategies for Vehicular Ad-Hoc Networks," Department of Computer Science, University of Mannheim, Atlanta, Georgia, USA, Tech. Rep. TR-02-003, Jul. 2002. [Online]. Available: http://www.cn.uni-duesseldorf.de/publications/details/Fuessler2002b.html.[Accessdate10/12/2013]

[83] S. Jain and S. Sahu, "Topology vs. Position based Routing Protocols in Mobile Ad hoc Networks: A Survey," *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, no. 3, pp. 1–11, May 2012. [Online]. Available: http://www.ijert.org/browse/may-2012-edition?download=34%.[Accessdate10/12/2013]

[84] P. H. Dana, "Global Positioning System (GPS) Time Dissemination for Real-Time Applications," *Real-Time Systems*, vol. 12, no. 1, pp. 9–40, Jan. 1997. [Online]. Available: http://www.pdana.com/PHDWWW_files/Rtgps.pdf.[Accessdate10/12/2013]

[85] N. Bulusu, J. Heidemann, and J. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Journal*, vol. 7, no. 5, pp. 28–35, Oct. 2000. [Online]. Available: http://dx.doi.org/10.1109/98.878533.[Accessdate10/12/2013]

[86] K. Abrougui, P. Richard Werner Nelem, and A. Boukerche, "Performance Evaluation of Location-Based Service Discovery Protocols for Vehicular Networks," in *the Proceeding of the IEEE International Conference on Communications (ICC)*. Cape Town: IEEE, 23-27 May 2010, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/ICC.2010.5502662.

[87] S. M. Das, H. Pucha, and Y. C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," in *the Proceeding of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. Miami, FL, USA: IEEE, 13-17 Mar. 2005, pp. 1228–1239. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2005.1498349.[Accessdate10/12/2013]

[88] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *the Proceeding of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, Texas, United States, 25-30 Oct. 1998, pp. 76–84. [Online]. Available: http://doi.acm.org/10.1145/288235.288254.[Accessdate10/12/2013]

[89] M. Kasemann, H. Hartenstein, and M. M., "A Reactive Location Service for Mobile Ad Hoc Networks," Department of Computer Science, University of Mannheim, Tech. Rep. TR-14-2002, 2002. [Online]. Available: http://www.cn.uni-duesseldorf.de/publications/details/Kaesemann2002c.html.[Accessdate10/12/2013]

[90] Z. J. Haas and B. Liang, "Ad Hoc Mobility Management with Uniform Quorum Systems," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 228–240, Apr. 1999. [Online]. Available: http://dx.doi.org/10.1109/90.769770. [Accessdate10/12/2013]

[91] L. Jinyang, J. John, S. J. D. C. Douglas, R. K. David, and M. Robert, "A Scalable Location Service for Geographic Ad Hoc Routing," in *the Proceeding of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 6-11 Aug. 2000, pp. 120–130. [Online]. Available: http://doi.acm.org/10.1145/345910.345931.[Accessdate10/12/2013]

[92] C.-C. Hsu and . I. F. I. Chin-Laung Lei., "A Geographic Scheme with Location Update for Ad Hoc Routing," in *the Proceeding of the 4th IEEE International Conference on Systems and Networks Communications*. Porto: IEEE, 20-25 Sept. 2009, pp. 43–48. [Online]. Available: http://dx.doi.org/10.1109/ICSNC.2009.83.

[93] M. Ayaida, M. Barhoumi, H. Fouchal, Y. Ghamri-Doudane, and L. Afilal, "HHLS: A hybrid Routing Technique for VANETs," in *the Proceeding of the IEEE Global Communications Conference (GLOBECOM)*. Anaheim, CA: IEEE, 3-7 Dec. 2012, pp. 44–48. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2012.6503088.

[94] Y.-B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, Jul. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1019106118419. [Accessdate10/12/2013]

[95] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. L. Boudec, "Self Organization in Mobile Ad Hoc Networks: the Approach of TERMINODES," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 166–174, Jun. 2001. [Online]. Available: http://dx.doi.org/10.1109/35.925685. [Accessdate10/12/2013]

[96] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–257, Mar. 1984. [Online]. Available: http://dx.doi.org/10.1109/TCOM.1984.1096061.

[97] R. Nelson and L. Kleinrock, "The Spatial Capacity of A slotted ALOHA Multihop Packet Radio Network with Capture," *IEEE Transactions on Communications*, vol. 32, no. 6, pp. 684–694, Jun. 1984. [Online]. Available: http://dx.doi.org/10.1109/TCOM.1984.1096124.

[98] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," in *the Proceeding of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, 15-18 Aug. 1999, pp. 51–54. [Online]. Available: www.cccg.ca/proceedings/1999/c46.pdf.[Accessdate10/12/2013]

[99] T.-C. Hou and V. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Transactions in Communications*, vol. 34, no. 1, pp. 38–44, Jan. 1986. [Online]. Available: http://dx.doi.org/10.1109/TCOM.1986.1096436.[Accessdate10/12/2013]

[100] B. N. Karp, "Geographic Routing for Wireless Networks," Ph.D Dissertation, Harvard University, Cambridge, Massachusetts, 2000. [Online]. Available: https://www-new.comp.nus.edu.sg/~bleong/geographic/related/karp00geographic.pdf.[Accessdate10/12/2013]

[101] J. Gao and L. Zhang, "Load Balanced Short Path Routing in Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 377–388, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2006.49.[Accessdate10/12/2013]

[102] D. Tran and H. Raghavendra, "Routing with Congestion Awareness and Adaptivity in Mobile Ad Hoc Networks," in *the Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, 13-17 Mar. 2005, pp. 1988–1994. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2005.1424824.

[103] N. Tantubay, D. R. Gautam, and M. K. Dhariwal, "A Review of Power Conservation in Wireless Mobile Adhoc Network (MANET)," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 4, pp. 378–383, Jul. 2011. [Online]. Available: http://www.ijcsi.org/papers/IJCSI-8-4-1-378-383.pdf.[Accessdate10/12/2013]

[104] F. Xue and P. R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," *Wireless Networks Journal*, vol. 10, no. 2, pp. 169–181, Mar. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:WINE.0000013081.09837.c0.[Accessdate10/12/2013]

[105] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An Analysis of the Optimum Node Density for Ad hoc Mobile Networks," in *the Proceedings of the IEEE International Conference In Communications (ICC'01)*. Helsinki: IEEE, 11-14 Jun. 2001, pp. 857–861. [Online]. Available: http://dx.doi.org/10.1109/ICC.2001.937360.[Accessdate10/12/2013]

[106] N. Ali, R. Ahmad, and S. Aljunid, "Link Availability Estimation for Routing Metrics in MANETs: An Overview," in *International Conference on Electronic Design (ICED)*. Penang, Malaysia: IEEE, 1-3 Dec. 2008, pp. 1–3. [Online]. Available: http://dx.doi.org/10.1109/ICED.2008.4786686.

[107] Z. Cheng and W. B. Heinzelman, "Discovering Long Lifetime Routes in Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 661–674, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2007.06.001.

[108] E. Y. Hua and Z. Z. Haas, "An Algorithm for Prediction of Link Lifetime in MANET Based on Unscented Kalman Filter," *IEEE Communications Letters*, vol. 13, no. 10, pp. 782–784, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2009.090974.[Accessdate10/12/2013]

[109] M. Shanudin, M. Ismail, and R. Saqour, "Impact of Mobility Metrics on Greedy Ad Hoc Network Routing Protocol and Improvement Using Angular Prediction Model," in *the Proceeding of the 13th IEEE International Conference on Networks, Jointly held with the 7th Malaysia International Conference on Communication*, Kuala Lumpur, Malaysia, 16-18 Nov. 2005, pp. 262–267. [Online]. Available: http://dx.doi.org/10.1109/ICON.2005.1635481.

[110] Y. Kim, J.-J. Lee, and A. Helmy, "Modeling and Analyzing the Impact of Location Inconsistencies on Geographic Routing in Wireless ," *Mobile Computing and Communications Review*, vol. 8, no. 1, pp. 48–60, Jan. 2004. [Online]. Available: http://doi.acm.org/10.1145/980159.980168. [Accessdate10/12/2013]

[111] R.-H. Cheng and C. Huang, "Efficient Prediction-Based Location Updating and Destination Searching Mechanisms for Geographic Routing in Mobile Ad Hoc Networks," *Journal Of Information Science And Engineering*, vol. 28, no. 1, pp. 115–129, Aug. 2012. [Online]. Available: http://www.iis.sinica.edu.tw/page/jise/2012/201201_08.html,[Accessdate10/12/2013]

[112] G. T. Toussaint, "The Relative Neighbourhood Graph of a Finite Planar Set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980. [Online]. Available: http://dx.doi.org/10.1016/0031-3203(80)90066-7.[Accessdate10/12/2013]

[113] R. K. Gabriel and R. R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Biology*, vol. 18, no. 3, pp. 259–278, Sep. 1969. [Online]. Available: http://dx.doi.org/10.2307/2412323.

[114] K. Jaffres-Runser, C. Comaniciu, and J.-M. Gorce, "A Multiobjective Optimization Framework for Routing in Wireless Ad Hoc Networks," Dept. of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New-Jersey, USA, Research Report RR-7180, Jan. 2010. [Online]. Available: http://hal.inria.fr/inria-00449010/PDF/RR-7180.pdf.[Accessdate10/12/2013]

[115] M. Heissenbuttel and T. B., "Optimizing Neighbor Table Accuracy of Position-Based Routing Algorithms," in *the Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, Miami, FL USA, 13-17 Mar. 2005, pp. 143–152. [Online]. Available: citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.1625.[Accessdate10/12/2013]

[116] S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *Network Working Group RFC 2501*, Jan. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2501.txt.

[117] K. M. Reineck, "Evaluation and Comparison of Network Simulation Tools," Master Thesis, University of Applied Sciences Bonn-Rhein-Sieg Department of Computer Science, Aug. 2008. [Online]. Available: http://www.projektiv.net63.net/doc/Reineck_-_Evaluation_and_Comparison_of_Network_Simulation_Tools.pdf.[Accessdate10/12/2013]

[118] S. R. Das, R. Castañeda, and J. Yan, "Simulation-Based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications (MONET) Journal*, vol. 5, no. 3, pp. 179–189, Sep. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1019108612308. [Accessdate10/12/2013]

[119] R. E. Shannon, "Design and Analysis of Simulation Experiments," in *the Preceding of the 10th conference on Winter simulation WSC*, ser. WSC '78, vol. 1. Piscataway, NJ, USA: IEEE Press, 1978, pp. 55–61. [Online]. Available: http://dl.acm.org/citation.cfm?id=800288.811210.

[120] O. Al-Momani, "Dynamic Redundancy Forward Error Correction Mechanism for the Enhancement of Internet-based Video Streaming," Ph.D. Thesis, Universiti Utara Malaysia, 2010. [Online]. Available: http://etd.uum.edu.my/2523/.[Accessdate10/12/2013]

[121] O. Ghazali, "Scaleable and Smooth TCP-friendly Receiver-based Layered Multicast Protocol," Ph.D. Thesis, Universiti Utara Malaysia, 2008. [Online]. Available: http://etd.uum.edu.my/1291/.[Accessdate10/12/2013]

[122] R. G. Sargent, "Validation and Verification of Simulation Models," in *the Preceding of the 2011 Winter Simulation Conference (WSC)*, Phoenix, Ariz, USA, 11-14 Dec. 2011, pp. 183–198. [Online]. Available: http://dx.doi.org/10.1109/WSC.2011.6147750.[Accessdate10/12/2013]

[123] O. Balci, "Principles of Simulation Model Validation, Verification, and Testing," *Transactions of the Society for Computer Simulation International Journal*, vol. 14, no. 1, pp. 3–12, Mar. 1997. [Online]. Available: http://dl.acm.org/citation.cfm?id=264096.264099.

[124] U. Atta, K. Rehman, M. Sardar, and O. Mazliza, "A Performance Comparison of Network Simulators for Wireless Networks," *arXiv preprint arXiv:1307.4129*, pp. 1–6, Jul. 2013. [Online]. Available: http://arxiv.org/ftp/arxiv/papers/1307/1307.4129.pdf.

[125] *The Network Simulator - ns-2*. [Online]. Available: http://www.isi.edu/nsnam/ns/.[Accessdate10/12/2013]

[126] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 2nd ed., July 2008. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-71760-9. [Accessdate10/12/2013]

[127] *MATLAB Primer R2012b*. [Online]. Available: www.mathworks.com/help/pdf_doc/matlab/getstart.pdf.[Accessdate10/12/2013]

[128] *The Network Simulator MatLap: Documentation*. [Online]. Available: http://wireless-matlab.sourceforge.net/.[Accessdate10/12/2013]

[129] E. Natsheh, A. B. Jantan, S. Khatun, and S. Shamala, "A Survey on Fuzzy Reasoning Applications for Routing Protocols in Wireless Ad Hoc Networks," *International Journal of Business Data Communications and*

Networking (IJBDCN), vol. 4, no. 2, pp. 22–37, 2008. [Online]. Available: http://dx.doi.org/10.4018/jbdcn.2008040102.

[130] L. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965. [Online]. Available: http://dx.doi.org/10.1016/S0019-9958(65)90241-X.[Accessdate10/12/2013]

[131] E. Cox, "Fuzzy Fundamentals," *IEEE Spectrum Magazine*, vol. 29, no. 10, pp. 58–61, Oct. 1992. [Online]. Available: http://dx.doi.org/10.1109/6.158640.

[132] V. B. Robinson, "A perspective on the fundamentals of fuzzy sets and their use in geographic information systems," *Transactions in GIS*, vol. 7, no. 1, pp. 3–30, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1111/1467-9671.00127

[133] E. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with A Fuzzy Logic Controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, Jan. 1975. [Online]. Available: http://dx.doi.org/10.1016/S0020-7373(75)80002-2.

[134] J. M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995. [Online]. Available: http://dx.doi.org/10.1109/5.364485.[Accessdate10/12/2013]

[135] J. Bas and A. Neira, "A Fuzzy Logic System for Interference Rejection in Code Division Multiple Access," in *the Preceding of the 12th IEEE international conference on fuzzy system (FUZZ)*, MO, USA, 25 - 28 May 2003, p. 996–1001. [Online]. Available: http://dx.doi.org/10.1109/FUZZ.2003.1206567.

[136] E. Natsheh, A. B. Jantan, S. Khatun, and S. Shamala, "Adaptive Optimizing of Hello Messages in Wireless Ad-Hoc Networks," *International Arab Journal Information Technology*, vol. 4, no. 1, pp. 191–200, Jan. 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/iajit/iajit4.html#NatshehJKS07a. [Accessdate10/12/2013]

[137] A. Agarwal and S. Das, "Dead Reckoning in Mobile Ad Hoc Networks," in *the Proceeding of the IEEE Wireless Communications and Networking (WCNC'03)*. New Orleans, LA, USA: IEEE, 20-20 Mar. 2003, pp. 1838–1843. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2003.1200666.

[138] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Boder Effects," *ACM Mobile Computing and Communications Review (SIGMOBILE)*, vol. 5, no. 3, pp. 55–67, Jul. 2001. [Online]. Available: http://doi.acm.org/10.1145/584051.584056.

[139] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications & Mobile Computing (WCMC): Special Issue On Mobile Ad Hoc Networking: Research, Trends And Applications*, vol. 2, no. 5, pp. 483–502, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1002/wcm.72.[Accessdate10/12/2013]

[140] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1109/TNET.2002.801403.[Accessdate10/12/2013]

[141] F. Bai, N. Sadagopan, and A. Helmy, "The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks," *AdHoc Networks Journal*, vol. 1, no. 1, pp. 383–403, Nov. 2003. [Online]. Available: http://dx.doi.org/10.1016/S1570-8705(03)00040-4.[Accessdate10/12/2013]

[142] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," in *the Preceding of the 22ed Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, San Fransisco, California, USA, 30 Mar.-3 Apr. 2003, pp. 1312–1321. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2003.1208967.[Accessdate10/12/2013]

[143] W. Navidi and T. Camp, "Stationary Distributions for the Random Waypoint Mobility Model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 99–108, Jan.-Mar. 2004. [Online]. Available: http://dx.doi.org/10.1109/TMC.2004.1261820.[Accessdate10/12/2013]

[144] J.-Y. Le Boudec and M. Vojnovic, "Perfect Simulation and Stationarity of A Class of Mobility Models," in *the Preceding of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, 13-17 Mar. 2005, pp. 2743 – 2754. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2005.1498557.[Accessdate10/12/2013]

[145] G. Mohimani, F. Ashtiani, A. Javanmard, and M. Hamdi, "Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1998 – 2007, May 2009. [Online]. Available: http://dx.doi.org/10.1109/TVT.2008.2004266.[Accessdate10/12/2013]

[146] L. Zhuoqun, S. Lingfen, and E. Ifeachor, "Range-based Mobility Estimations in MANETs with Application to Link Availability Prediction," in *the Proceedings of the IEEE International Conference on Communications*. Glasgow: IEEE, 24-28 Jun. 2007, pp. 3376–3382. [Online]. Available: http://dx.doi.org/10.1109/ICC.2007.559.

[147] D. Veljan, "The 2500-year-old Pythagorean Theorem," *Mathematics Magazine*, vol. 73, no. 4, pp. 259–272, Oct. 2000. [Online]. Available: www.jstor.org/pss/2690973.[Accessdate10/12/2013]

[148] S. Bai, Z. Huang, and J. Jung, "Beacon-based Cooperative Forwarding Scheme for Safety-related Inter-vehicle Communications," in *the Proceedings of the 2010 International Conference on Computational Science and Its Applications - Volume Part IV*, ser. ICCSA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 520–534. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12189-0_45.

[149] D. Triantafyllidou and K. Al Agha, "Evaluation of TCP Performance in MANETs using an Optimized Scalable Simulation Model," in *the Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '07)*, Istanbul, 24-26 Oct. 2007, pp. 31 – 37. [Online]. Available: http://dx.doi.org/10.1109/MASCOTS.2007.25.

[150] D. Perkins, H. Hughes, and C. Owen, "Factors Affecting the Performance of Ad Hoc Networks," in *the proceeding of the IEEE International Conference on Communication (ICC'02)*. New York NY, USA: IEEE, 28 Apr. - 2 May 2002, pp. 2048–2052. [Online]. Available: http://dx.doi.org/10.1109/ICC.2002.997208.[Accessdate10/12/2013]

[151] S. H. Kurkowski, "Credible Mobile Ad Hoc Network Simulation-based Studies," Ph.D. Dissertation, Faculty and the Board of Trustees of the Colorado School of Mines, Golden, CO, USA, 2006. [Online]. Available: http://www.dtic.mil/get-tr-doc/pdf?Location=U2&doc=GetTRDoc.pdf&AD=ADA462915.[Accessdate10/12/2013]

[152] K. Stuart , N. William and C. Tracy, "Discovering Variables that Affect MANET Protocol Performance," in *the Proceeding of the IEEE Global Telecommunications Conference (GLOBECOM'07)*. Washington, DC: IEEE, 26-30 Nov. 2007, pp. 1237–1242. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2007.238.

[153] M. Hyland, B. Mullins, R. Baldwin, and M. Temple, "Simulation-based Performance Evaluation of Mobile Ad Hoc Routing Protocols in A Swarm of Unmanned Aerial Vehicles," in *the Preceding of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*. IEEE, 21-23 May 2007, pp. 249–256. [Online]. Available: http://dx.doi.org/10.1109/AINAW.2007.336.

[154] K. Stuart , N. William and C. Tracy, "Constructing MANET Simulation Scenarios that Meet Standards," in *the Preceding of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*. Pisa: IEEE, 8-11 Oct. 2007, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/MOBHOC.2007.4428640.

[155] R. G. Sargent, "Validation and Verification of Simulation Models," in *the Proceedings of the 2008 Winter Simulation Conference*, no. „ Florida, USA, 7-10 Dec. 2008, pp. 157–169. [Online]. Available: http://dl.acm.org/citation.cfm?id=1516744.1516780.[Accessdate10/12/2013]

[156] T. C. M. GROUP, *Wireless and Mobility Extensions to ns-2*, THE CMU MONARCH GROUP, Oct. 1999. [Online]. Available: http://www.monarch.cs.cmu.edu/cmu-ns.html.Accessdate10/12/2013

[157] R. Santos, O. Alvarez, and A. Edwards, "Performance Evaluation of two Location-based Routing Protocols in Vehicular Ad-Hoc Networks," in *the proceeding of the IEEE 62nd Vehicular Technology Conference (VTC'05)*,

vol. 4, Dallas, Texas, USA, 25-28 Sept. 2005, pp. 2287 – 2291. [Online]. Available: http://dx.doi.org/10.1109/VETECF.2005.1558956.

[158] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," in *the Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. Dallas, Texas, USA: ACM, Oct. 1998, pp. 85–97. [Online]. Available: http://doi.acm.org/10.1145/288235.288256.[Accessdate10/12/2013]

[159] G. Almes, S. Kalidindi, A. Morton, and M. Zekauskas, "A One-Way Delay Metric for IPPM (draft-morton-ippm-2679-bis-00)," *Network Working Group RFC 4656*, 2012. [Online]. Available: http://www.hjp.at/doc/rfc/rfc4656.html.

[160] S. Charcranoon, "Measurement Architecture to Obtain Per-hop One-way Packet Loss and Delay in Multi-class Service Networks," USA Patent 7,292,537, Nov., 2007. [Online]. Available: http://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=7292537B2&KC=B2&FT=D.[Accessdate10/12/2013]

[161] *IEEE Computer Society LAN MAN Standards Committee, Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*, The Institute of Electrical and Electronics Engineers, New York, NY, 1997. [Online]. Available: www.cs.uiuc.edu/homes/haiyun/cs598hl/papers/802.11-1999.pdf.[Accessdate10/12/2013]

[162] H. Chen and Y. Li, "Performance Model of IEEE 802.11 DCF with Variable Packet Length," *IEEE Communications Letters*, vol. 8, no. 3, pp. 186–188, Mar. 2004. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2004.823429.

[163] J. Simo Reigadas, A. Martinez-Fernandez, J. Ramos-Lopez, and J. Seoane-Pascual, "Modeling and Optimizing IEEE 802.11 DCF for Long-distance Links," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 881–896, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1109/TMC.2010.27.

[164] D. Kim, C.-K. Toh, J.-C. Cano, and P. Manzoni, "A Bounding Algorithm for the Broadcast Storm Problem in Mobile Ad Hoc Networks," in *the Preceding of the IEEE Wireless Communications and Networking (WCNC)*, vol. 2. IEEE, 16-20 Mar. 2003, pp. 1131–1136. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2003.1200530.

[165] E. M. Royer, S.-J. Lee, and C. E. Perkins, "The Effects of MAC Protocols on Ad Hoc Network Communication," in *the Proceedings of the IEEE Wireless Communications and Networking Confernce (WCNC)*, vol. 2. Santa Barbara, CA, USA: IEEE, 23-28 Sep. 2000, pp. 543–548. [Online]. Available: http://doi.acm.org/10.1109/WCNC.2000.903911,2013.03.14.[Accessdate10/12/2013]

[166] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, "A Survey on Routing Metrics," Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland, Technical Report TIK/262, Feb. 2007. [Online]. Available: http://www.baumann.info/public/tik262.pdf.[Accessdate10/12/2013]

[167] *Normalization Rank: Tutoriall, Kardi Teknomo's Personal Home Page*. [Online]. Available: http://people.revoledu.com/kardi/personal/cv/BriefCV.htm.[Accessdate10/12/2013]

[168] Y. Wang, X.-Y. Li, W.-Z. Song, M. Huang, and T. Dahlberg, "Energy-Efficient Localized Routing in Random Multihop Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1249–1257, Aug. 2011. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2010.198.[Accessdate10/12/2013]

[169] N. Luttenberger and H. Peters, "Node Degree-based Improved Hop Count Weighted Centroid Localization Algorithm," in *the Proceeding of the 17th Conference on Communication in Distributed Systems (GI/ITG)*, vol. 17. Kiel, Germany: Dagstuhl-Leibniz Center for computer science, 8-11 Mar. 2011, pp. 194–199. [Online]. Available: drops.dagstuhl.de/opus/volltexte/2011/2972/.

[170] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 107–123, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1016/S1570-8705(03)00015-5.

[171] A. Mohammad, X. H., M. Islam, and K. Zunnurhain, "Delay Analysis of Wireless Ad Hoc Networks: Single vs. Multiple Radio," in *the Proceedings of the IEEE 35th Conference on Local Computer Networks (LCN)*. Denver, CO: IEEE, 10-14 Oct. 2010, pp. 814–820. [Online]. Available: http://dx.doi.org/10.1109/LCN.2010.5735817.

[172] J. Li, Z. Li, and P. Mohapatra, "Adaptive Per Hop Differentiation for End-to-end Delay Assurance in Multihop Wireless Networks," *Ad Hoc Netw.*, vol. 7, no. 6, pp. 1169–1182, Aug. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2008.10.005.[Accessdate10/12/2013]