MODEL FOR ROUTING SCHEDULING IN A MOBILE ADHOC NETWORK

UWACHUKWUNENYE MACDONALD AKARACHUKWU

15010301018

BEING A PROJECT SUBMTTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES

IN FULFILMENT OF THE REQUIREMENTS FOR THE

AWARD OF DEGREE OF BACHELOR OF SCIENCE (B.Sc.)

IN

COMPUTER SCIENCE

MOUNTAIN TOP UNIVERSITY, IBAFO

OGUN STATE, NIGERIA

2020

CERTIFICATION

This is to certify that this project was carried out by Uwachukwunenye MacDonald Akarachukwu (Matric Number 150103010I8) under the supervision and guidance of Mr J.A Balogun in the Department of Computer Sciences, Computer Science Program, College of Basic and Applied Sciences, Mountain Top University, Makogi Oba, Ogun State.

……………………………. ……………………………..

MR J.A BALOGUN Date

Supervisor

….……………………………. …………………...…………

DR I.O. IBIDAPO Date

Head of Department

DEDICATION

This project work is dedicated to the Almighty God, the maker of heaven and earth, by whose Grace I am alive to see the completion of this project.

ACKNOWLEDGEMENT

The success and final outcome of this project goes to the Almighty God for wisdom and understanding.

I specially appreciate my Supervisor Mr J.A Balogun who took keen interest in my project work and guided me all along and taking the time out of no time to attend to me.

I will like to acknowledge the entire Management of Mountain Top University, The Vice Chancellor, Prof. Elijah Ayolabi, The Dean and entire college of Basic and Applied Science, Department of Computer Sciences, Head of Department Computer Science and Mathematics, Dr I.O. Akinyemi, my Supervisor, Mr J.A Balogun and owe deep gratitude for the efforts, constant encouragement, guidance and support of all the academic staff; Mr J.A Balogun, Mr O. J. Falana, Dr O. O. Taiwo, Dr O.B. Alaba, Prof. P.A. Idowu, Dr O.A. Ojesanmi, Dr F. A. Kasali, Dr O.A. Taiwo, Dr F.O. Oladeji and non-academic staff of the Department of Computer Science and Mathematics. I would like to give a big thank you to Mr O.J. Osaghae and Pastor G. Oyebande for their guidance, moral and financial supports. For the teachings and contributions that have brought out positive values in me and making my stay worthwhile. I say God bless you richly.

I heartily thank my parents Mr and Mrs Patrick. O Uwa and my wonderful siblings, thank you all for your moral and financial support. I am grateful for all the investments into my education and future. I would not forget to remember all the students in the Department of Computer Science and Mathematics, for making my stay a worthwhile one, I say God bless you all richly.

TABLE OF CONTENT

ABSTRACT

Mobile ad hoc networks (MANETS) are self-created and self-organized by a collection of mobile nodes, interconnected by multi-hop wireless paths in a strictly peer to peer fashion. DSR (Dynamic Source Routing) is an on-demand routing protocol for wireless ad hoc networks that floods route requests when the route is needed. Route caches in intermediate mobile node on DSR are used to reduce flooding of route requests. But with the increase in network size, node mobility and local cache of every mobile node cached route quickly become stale or inefficient. In this paper, for efficient searching, we have proposed a generic searching algorithm on associative cache memory organization to faster searching single/multiple paths for destination if exist in intermediate mobile node cache with a complexity (Where n is number of bits required to represent the searched field).The other major problem of DSR is that the route maintenance mechanism does not locally repair a broken link and Stale cache information could also result in inconsistencies during the route discovery /reconstruction phase. So, to deal this, we have proposed an optimized cache coherence handling scheme for on -demand routing protocol (DSR).

Mobile Ad-Hoc Network (MANET) is an infrastructure less wireless network of autonomous collection of mobile nodes (Smart phones, Laptops, iPads, PDAs etc.). Network is self-configured to reconstruct its topology and routing table information for the exchange of data packets on the joining and leaving of each node on ad-hoc basis. This project is based on the MANET applications and challenges. The researchers can get the overall concept of MANET as well as its applications and challenges.

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

Ad-hoc networks consist of a collection of wireless hosts that are free to move randomly. These networks operate without the support of any fixed infrastructure or centralized administration and are completely self-organizing and self-configuring. Nodes are connected dynamically and in an arbitrary manner to form a network, depending on their transmission ranges and positions. Nodes can communicate directly with all nodes within transmission range. As transmission ranges are limited, two nodes may not be able to communicate directly and they must rely on other nodes to forward their packets. A source sends a packet to one or more of its neighbours, which in turn forward the packet to their neighbours, and so on until the destination is finally reached. Thus, nodes must cooperate to provide connectivity and paths are normally multihop. (Gelenbe, *et al.*, 2006).

Routing deals with finding appropriate paths between source and destination nodes, possibly over many intermediate nodes. Traditional routing protocols for fixed wired networks are not adequate for ad-hoc networks and perform poorly because of these networks' distinct characteristics such as the rapidly changing topology, the broadcast propagation medium, and the existence of unidirectional links. Many routing protocols designed for ad-hoc networks have been proposed in the literature. One can distinguish between topology-based and position-based routing protocols. Like routing protocols in the Internet, topology-based routing protocols use routing tables and information about available links to forward packets based on the destination address. (Meng *et al.*, 2006). These topology-based routing protocols are adequate for many kinds of ad-hoc network such as networks with only a few hundred nodes and spontaneous networks where people meet at a convention centre and want to share data. on the other hand, these routing protocols show poor adaptation in ad hoc networks with frequently changing topology. These changes result in slow convergence behaviour or even inconsistent routing tables. Furthermore, these protocols scale poorly with a very large number of nodes as the signalling traffic and the number of required control packets becomes prohibitive.

In position-based routing protocols (also known as geographical, geometric, or location-based routing protocols), the nodes' geographical positions are used to make routing decisions. A node forwards a packet to the neighbour that is geographically closest to the destination position in a greedy manner. If this greedy routing fails, the packet is forwarded further in a recovery mode. Therefore, a node must be able to determine its own position and the position of the destination node. This information is generally provided by a global navigation satellite system and a location service, respectively. The location service is responsible for maintaining nodes' positions and replying to requests for destination node locations from source nodes. (Kyasanur and Vaidya, 2005).

These characteristics make position-based routing protocols especially suitable for ad-hoc networks with highly dynamic topologies and/or a large number of nodes, where topology-based protocols have their limitations. Typical ad hoc networks that have such characteristics are vehicular ad-hoc networks which are envisioned and already deployed to enable on-board safety systems, virtual traffic signs, and real-time information on traffic and congestion. This project work seeks to make ad hoc networks have better efficiency and utilize the channel diversity by developing a model for routing scheduling in ad hoc networks.

1.2 Statement of the Problem

Position-based protocols are more suitable for the kinds of ad hoc networks with highly dynamic topologies and a large number of nodes than topology-based protocols, but they still have some drawbacks. These drawbacks are the major problems that need urgent solution in order to maximise the full capacity of ad hoc networks. This project seeks to find a solution to them by implementing a model for routing scheduling in ad hoc networks. These drawbacks can be broadly classified in two categories. Drawbacks caused by the required control traffic and drawbacks originating from using the positions and distances of nodes as the only criteria for routing.

These drawbacks can be summarized by saying that protocols are stateful concerning neighbourhood topology and stateless about the topology of the network on a large scale. This is however exactly the opposite of what seems intuitive and logical. The neighbourhood changes frequently and at unpredictable times and, thus, protocols should avoid to maintain state about the local network topology. On the other hand, the overall node distribution in the network remains quite static and only varies slowly over time, e.g., because people tend to stay in towns and move along streets. Therefore, it is beneficial to accumulate such information at the nodes to facilitate communication on a large scale.

The position-based routing and scheduling model proposed in this project aims to overcome one of these two types of drawbacks. Due to time and other resources, this project will focus on solving the first drawback. The proposed model for routing protocols and the broadcast protocol address the first type of drawbacks caused by the required control traffic and the associated local statefulness.

1.3 Aim and Objectives

The aim of this work is to develop a better way to control routing.

The objectives are to:

i. design the model for routing scheduling in ad hoc networks;
ii. implement the model as an application software using MATLAB
iii. evaluate the developed prototype for effectiveness.

1.4 Project Methodology

The methodology to be employed in this work is by the use of model design and development approach to the routing and scheduling in adhoc networks. The model to be implemented using MATLAB version 2015. The whole project implementation stage is to be divided into four phases in order with each of the phases to be executed:

a. Phase 1: This phase involves the gathering of required information from the public, sampling the opinions stake holders in the mobile network and I.T industries. This phase helped in getting various information that gives insight into what area of the project to give rapt attention to. Various methods to be applied in gathering of more information ranging from questionnaires, personal interview etc.
b. Phase 2: This phase deals with building of the routing model and its programming on MATLAB.
c. Phase 3: This phase involves testing each module that has been programmed, to see if the desired features have been properly represented. A critical inspection will be carried out after successful implementation to show that its performance meets the project requirements and objectives. Every error noticed to be reported and corrected.

d. Phase 4: This phase involves correction of bugs and improvement upon the already implemented model to improve it.

1.5 Motivation of Study

Wireless networks play a crucial role in the communication systems nowadays. Wireless networks are being increasingly used in the communication among devices of the most varied types and sizes. User mobility, affordability, flexibility and ease of use are few of many reasons for making them very appealing to new applications and more users every day. In this work, only wireless networks capable of operating without the support of any fixed infrastructure are to be considered. More precisely, this project will consider wireless ad hoc networks. The diversity of the applications supported by wireless ad hoc explain the success of this type of network. These applications concern as various domains as environmental monitoring, wildlife protection, emergency rescue, home monitoring, target tracking, exploration mission in hostile environments, etc. However, the most critical requirement for adopting such networks is energy efficiency.

1.6 Organisation of Thesis

This project work has been organized into five (5) chapters; the remaining part is divided into four more chapters as follows:

a. Chapter Two: In chapter two, relevant literatures on the topic and related subject matters are fleshed out. The chapter also contains past works which are intensively reviewed. These were done in order to perfectly understand the full details of routing scheduling, adhoc networks and the technicalities behind model development and its implementation on MATLAB. The chapter also contained a brief overview of the tools used in the implementation of the project work.

b. Chapter Three: This chapter explains the procedure followed in the building (the development and implementation) of the model on MATLAB. Entity relational diagrams, flowchart and other related diagrams were presented in this chapter. The relationships between the tools used were fleshed out.

c. Chapter four: Here, the result of the implementation in chapter three is presented. The tools used in the development and presentation of the model are to be showcased. The way it works is explained. Snapshots of the MATLAB pages of the model are to be captured and shown in this chapter as well.

d. Chapter Five: This is the chapter where a brief summary of the whole project work is to be given. Some conclusions to be drawn and recommendations to be given as well.

CHAPTER TWO

LITERATURE REVIEW

2.1 Chapter Introduction

Ad-hoc networks have several salient characteristics that make them quite different from other networks. The perhaps two most notable are the dynamic topology and the absence of any centralized infrastructure. These characteristics pose large challenges for protocols on any layer of the network stack. (Mobile) ad-hoc networks are often cited explicitly in the literature. However, the paradigm of ad-hoc networking has also been applied in various other contexts and distinct names for these networks emerged. All kinds of wireless networks have become increasingly popular over the past years. Due to the continuous technological advances, today's wireless and portable devices are small, light-weight, and have high computing capabilities.

The proliferation of these devices and the demand of users to communicate continuously are the driving forces behind the deployment of wireless networks. We can distinguish between two fundamentally different types of wireless networks. The first are known as infrastructure or cellular networks in which mobile hosts communicate with base stations connected to a fixed network infrastructure. If a node moves out of range of a base station, a handoff to a new base station within transmission range occurs. Data is routed through the wired network and only the last hop is wireless. The second type of wireless networks are ad-hoc networks, often also-called wireless multi-hop networks, which is probably the more descriptive term. These networks are simply formed by a collection of wireless hosts, which are often free to move randomly, and without any established infrastructure or centralized administration. Naturally, these networks must be self-configuring and self-organizing. Since source and destination nodes may not be within transmission range, the paths are normally multi-hop. All hosts act as routers and forward packets on behalf of other nodes to provide communication throughout the entire network. Ad-hoc networks have several distinct advantages over infrastructure networks for many applications and in many scenarios. Some of these reasons are their ease and speed of deployment, their robustness, as they do not depend on any infrastructure, and the low costs. Ad-hoc networks may also be used in combination with cellular networks to form so called hybrid or multi-hop cellular networks. Hybrid networks can help to extend the coverage and to increase the redundancy and the performance of cellular networks.

This chapter will help to understand the basic terms and concepts of the relevant subjects which are necessary to perfectly understand the problem statement and proposed solution in chapter one. Ad-hoc network and its applications, network routing and its configuration and many other background theories and concepts that are important for a perfect understanding of this work. To this end, many studies that have been carried out by different researchers which has provided good results are perused in this chapter. It will conclude with a brief discussion of some of the numerous previous works.

2.2 Brief Overview of Ad-hoc Network

The increased interest in ad-hoc networks is also reflected by the formation of the "Mobile Ad-hoc Network" working group within the Internet Engineering Task Force IETF (Manet, 2005). This working group currently focuses on routing protocols suitable for wireless routing applications. Many protocols have been proposed as

Internet Drafts and some reached RFC-status. The Internet Research Task Force IRTF (ans, 2005) has also established a research subgroup "Ad-hoc Network Systems" that investigates some specific areas in the context of adhoc networks like inter-layer protocol interaction, Quality-of-Service routing, routing scalability, and network auto-configuration. Ad-hoc is Latin and can be literally translated as "for this". The meaning is "for this purpose only". The term (mobile) ad-hoc network also is often used as an umbrella term for any kind of self-organizing, infrastructure-less, and wireless multi-hop networks.

Broadcasting in ad-hoc networks is different from broadcasting in wired networks for various reasons. The network topology may change frequently caused by mobility or by changes in the activity status of nodes. Broadcast protocols also have to cope with limited system resources in terms of bandwidth, computational, and battery power. Unlike wired networks where the total cost of the broadcast is normally calculated as the sum of all link costs, ad-hoc networks can make use of the broadcast property of the wireless medium. This allows to cover all neighbours with one single transmission. Consequently, the costs are typically not associated with the links between nodes but with the nodes themselves. Broadcasting is a common operation in ad-hoc networks and many protocols rely on the successful delivery of packets to each node in the network. For example, several routing protocols use broadcasting to detect routes from the source to the destination node such as AODV 196 and DSR (Johnson *et al.*, 2017). Other applications that require broadcasting are the paging or sending of an alarm signal to particular hosts. Furthermore, many applications make use of geo-casting such as in sensor networks are vehicular ad-hoc networks. Geo-casting may be considered basically as broadcasting of a packet to all nodes within a certain geographical area (Navas and Imielinski, 2017).

Broadcasting in ad-hoc networks is most simply and commonly realized by flooding whereby nodes rebroadcast each received packet exactly once. Duplicated packets are uniquely identified by the source node ID and a sequence number. Assuming we have a completely connected network, there may be up to as many transmissions as there are nodes in the network. Especially in dense networks, flooding generates a large number of redundant transmissions where most of them are not required to deliver the packet to all nodes. Nodes in the same area receive the packet almost simultaneously due to the highly correlated timing of retransmissions. This excessive broadcasting causes heavy contention and collisions, commonly referred to as the *broadcast storm problem* (Ni *et al.*, 2009), which consumes unnecessarily scarce network resources.

Two important objectives of any broadcast algorithm in ad-hoc networks are the reliability and optimizing resource utilization. First, reliability is concerned with the successful delivery of a packet to all nodes in the network. Even in a completely connected network, the packet may often not be delivered to all nodes since broadcast packets are normally not acknowledged and the broadcast storm makes the transmissions highly unreliable. Secondly, the use of network resources should be minimized without affecting the reliability. Interestingly, these objectives are complementary. Reducing the number of transmissions may also increase reliability as it alleviates the broadcast storm. In mobile networks with constantly changing topologies, it is impossible to broadcast optimally a packet network-wide. In static networks, this may be possible, often however with a prohibitive amount of control traffic only. Thus, most practical broadcast algorithms for ad-hoc networks try to approach network-wide optimal broadcasting by optimizing the local broadcasting of packets.

2.3 Characteristics of Ad-Hoc Networks

Ad-hoc networks have several distinct characteristics that make them quite different from conventional cellular networks. These characteristics are mainly due to the absence of any fixed infrastructure, the wireless propagation medium, and the limited resources of the mobile devices. The most salient characteristics which should be considered by protocol designers were addressed within the “Mobile Ad-hoc Networks” working group and are listed in RFC 2501 (Macker and Corson, 2019).

2.3.1 Dynamic Topologies

Major reasons for topology changes are the mobility of the nodes, the adjustment of transmission and reception parameters, and the sleep cycles of nodes to save energy. Therefore, the network topology may change frequently and at unpredictable times.

2.3.2 Limited and Variable Capacity of the Links

The capacity of wireless links is and will presumably remain considerably lower than of wired links. Furthermore, the capacity of the links may vary over time because of the changing propagation conditions and the varying distances between nodes.

2.3.3 Power-Constrained Operation

Nodes in an ad-hoc network typically rely on batteries for their operation. Even though, battery capacity has doubled in energy density every 35 years (Powers, 2015), it still does not satisfy today’s demands. Unfortunately, however, a breakthrough is not expected in the near future (Stark *et al.*, 2012).

2.3.4 Limited Physical Security

Mobile wireless networks are generally more prone to security threats than fixed wired or cellular networks (Stine and de Veciana, 2014). The wireless propagation medium introduces vulnerabilities to malicious attacks varying from passive eavesdropping to active interference. Unlike cellular networks, ad-hoc networks do not have a centralized administration that can act as a trusted third party. Protocols designed for fixed and cellular networks are not appropriate to cope with these characteristics and new protocols tailored especially for ad-hoc networks are required. The protocols that we propose in this thesis are mainly concerned with the first three characteristics and implement mechanisms to operate efficiently in such scenarios. They do not address security issues as this is out of scope of this project work.

2.4 Examples of Ad-Hoc Networks

In the following, a brief description of some typical networks which are based on the paradigm of ad-hoc networks is given. These networks are often not pure ad-hoc networks, but rather hybrid networks as some nodes are connected to infrastructure networks. Most of these terms are not strictly defined, vary over time, or are used in different context by different authors. This section tries to give the most common used description of these different terms as they are used in our view nowadays.

2.4.1 Mesh Networks

Mesh networks may be used as a last mile solution where cabling is impossible, too expensive, or just as an alternate infrastructure in the event of failure (Bruno *et al.*, 2015). Nodes are deployed densely all over a certain area in order to enable broadband wireless access from home. The term is also used in the context of wireless metropolitan area networks, where wireless hotspots are interconnected to offer users wireless access. Several companies (mesh network, 2015) have already deployed such networks in various cities. Today, most often IEEE 802.11b (IEEE, 2010) is used as the underlying wireless technology. It is likely that new standards like the different variants of IEEE 802.16 (IEEE, 2010) and IEEE 802.20 (IEEE, 2010), which offer higher data rates and higher transmission ranges, will further boost this development and will complement and/or partially replace IEEE 802.11 b networks.

2.4.2 Wireless Personal Area Networks

A personal area network is a computer network used for communication among computer devices close to one person, including telephones and personal digital assistants. The reach of a personal area network is typically a few meters only and can be used for communication among the personal devices themselves and also for connecting to the Internet. Several commercially available wireless technologies like Bluetooth (Bluetooth, 2010) and other technologies which are currently under standardization in the IEEE 802.15 (IEEE, 2010) Working Group are typically used for these wireless personal area networks.

2.4.3 Vehicular ad-hoc Networks

Vehicular ad-hoc networks are used for on-board safety systems, virtual traffic signs, real-time congestion and traffic information, and commercial applications which require vehicle-to-vehicle or vehicle-to-roadside networking (Kellerer *et al.*, 2017). Vehicular ad-hoc networks have some distinct features compared to other ad-hoc networks such as large computational and infinite power resources. The mobility of the nodes may be quite high, but with mobility patterns constrained to roadways.

2.4.4 Sensor Networks

Low power and energy efficient radios and processors have made all types of sensor networks a reality (Akyildiz *et al.*, 2017). The tasks of the sensor networks are including object tracking, information collecting and querying, or producing a response to a certain event. The data collected and often already partially processed by the sensors is transmitted to a sink node that communicates with a monitoring centre. The number of nodes in sensor networks can be several orders of magnitude higher than in the previously discussed ad-hoc networks. Sensor nodes are very limited devices and have strict power, communication, computation, and memory constraints. Furthermore, sensor networks mainly use a broadcast or geo-cast communication paradigm, whereas most ad-hoc networks are based on point-to-point and unicast communication. For sensor networks, the power conservation may be the most important design parameter.

2.4.5 Spontaneous Networks

Spontaneous networking has got a lot of attention recently (Feeney *et al.*, 2019). Spontaneous networking can be described as the integration of services and devices with the objective that services are offered instantaneously to users without any manual intervention (Preuss and Cap, 2019). To achieve this objective, these networks also have to account for effective service discovery among devices (Richard, 2017).

2.4.6 Military and Rescue Networks

In scenarios like disaster rescue and military operations, one cannot rely on centralized administration or the availability of a communication infrastructure. The existing networks may be destroyed, may not be reliable in enemy regions, or just may not be accessible. To facilitate the operation, it is important to be able to deploy quickly a communication infrastructure. Consequently, ad-hoc networks have already been used early in military operations.

2.4.7 Packet Radio Networks

The idea of self-organizing wireless networks is not really new. Already back in the 1970’s, a multi-hop multiple-access packet radio network (Jubin and Tornow, 2007) was developed under the sponsorship of the Defence Advanced Research Projects Agency (DARPA). Packet radio networks are somehow synonymous to ad-hoc network even though mobility was not a major concern, as devices were heavy and not very mobile.

2.5 Routing Protocol Considerations

Routing deals with finding appropriate paths between source and destination nodes, possibly over many intermediate nodes. Depending on the underlying communication paradigm, we distinguish between unicast, multicast, any-cast, geo-cast, and broadcast routing. All these types of routing have one sender, but they differ in the number of destination nodes and the way the destination nodes are determined. In unicast communication, there is exactly one specific destination node. In any-cast routing, packets are delivered to exactly one destination among several possible destinations. Multicast is the delivery to multiple destinations which are aggregated in a multicast group. If the multicast group is defined as the set of all nodes within a specified geographical region, we speak of geo-casting. Broadcast aims at delivering packets to all nodes in the network. In this thesis, we are mainly concerned with unicast routing protocols and if not noted otherwise, we simply refer to unicast routing protocols as routing protocols.

Traditional routing protocols for fixed wired networks such as RIP (Malkin, 2008) and OSPF (Moy, 2008) are not adequate for the characteristics encountered in ad-hoc networks and perform poorly. The challenge of any routing protocol for ad-hoc networks is that they must be able to cope efficiently with their salient characteristics. RFC 2501 (Macker and Corson, 2016) describes differences between ad-hoc and fixed wired networks and discusses the resulting impact on the design and evaluation of network control protocols, focusing on routing protocols. An ad-hoc network is characterized by several defining parameters that should be considered during the design, the simulation, and the comparison of routing protocols. A networking context or a scenario is defined as a set of characteristics describing an ad-hoc network and its environment. In the literature, these parameters are typically used and varied to determine the performance of protocols in different network scenarios. In the following, we briefly describe some of these parameters; Network size (simply measured as the number of nodes which are member of the ad-hoc network), Network connectivity (measured by the average number of neighbours of a node and depends on the node density and the transmission range. This parameter is also referred to as the degree of a node), Topological rate of change (The rate with which the network topology is changing, i.e., the rate with which links break and new links come up. These changes are not only caused by mobility. Nodes toggling into and out of sleep states, interferences with other transmissions, and changing propagation characteristics can also cause changes in the topology), Fraction of unidirectional links (Unidirectional links may be the rule and not the exception caused by varying transmission ranges, different SINR (Signal to Interference plus Noise Ratio), etc. Thus, the effectiveness and efficiency of protocols in the presence of unidirectional links may be crucial) and Traffic patterns (which includes all different kinds of connection types and the distribution of the traffic load. Connections may be short- or long-lived, data may be transmitted in bursts or constantly and smoothly, some nodes may send/receive more traffic than others).

Quantitative and qualitative metrics are also proposed in RFC 2501 (Macker and Corson, 2016) in order to judge and compare ad-hoc network routing protocols. Some of them apply also to routing protocols for fixed, hardwired networks, whereas others are more specific for ad-hoc networks. As stated in RFC 2501, it is crucial that the metrics are independent of any specific routing protocol. The most often used quantitative metrics are the following; End-to-end data throughput and delay (These metrics are the two most important statistical measures of the effectiveness of routing performance and may include, e.g., mean, variance, and distribution), Packet delivery ratio (The effectiveness is often not only measured as the absolute throughput, but also as the fraction of successfully received data packets at the destination and transmitted packets at the source), Percentage out-of-order delivery (This measure is of particular importance to certain transport and application layer protocols such as TCP (Postel, 2011) and RTP (Schulzrinne *et al.*, 2013) which prefer in-order delivery of packets as sent by the source) and Overhead (which may be viewed as an internal measure of the routing protocol's effectiveness, often also called normalized routing load. Depending on the efficiency, a certain amount of overhead is required to achieve a certain level of data routing performance. The overhead can be measured in bits, but often it is measured as the average number of control and data packets transmitted per data packet delivered. This measure tries to capture a protocol's channel access efficiency. This is of special importance with contention-based MAC layer protocols such as the DCF (Distributed Coordination Function) of IEEE 802.11b [115], where the cost of channel access may be disproportional high for short control packets.

2.6 Dynamic Delayed Broadcasting

It is assumed that nodes are aware of their absolute geographical location by any means. Many applications in sensor and vehicular ad-hoc networks already require per se location information. This location information available for free can be used to optimize lower network operations such as routing and broadcasting. However, it is not required that a node has any information about its neighbourhood. Thus, no hello messages have to be transmitted periodically which saves scarce resources like bandwidth and battery power. The last broadcasting node only stores its current position in the header of the packet. This is the only external information required by other nodes in order to calculate when and whether to rebroadcast. Location information may not always be available however. DDB can also operate without location information and use incoming signal strength to approximate the distance to other transmitting nodes. As usual, packets are uniquely identified by their source node ID and a monotonically increasing sequence number. There is only one parameter taken by the algorithm called *Max Delay,* which indicates the maximum delay a packet can perceive per hop and is used to calculate the delay at the nodes.

2.6.1 Minimizing the Number of Transmissions

The objective of the first scheme DDB is to minimize the number of transmissions and at the same time to deliver the packet reliably to all nodes. Nodes that receive the broadcast packet use the concept of dynamic forwarding delay (DFD) to schedule the rebroadcasting and do not forward the packet immediately. From the position of the last visited node stored in the packet header and the node's current position, a node can calculate the estimated additional area that it would cover with its transmission. Depending on the size of this additionally covered area, the node introduces a delay before relaying the packet, where the delay is longer for a smaller additional area. In this way, nodes that have a higher probability of reaching additional nodes broadcast the packet first. Note that this is achieved without nodes having knowledge of their neighbourhood. Unlike stateful broadcast algorithms, the "best" nodes for rebroadcasting are chosen in a completely distributed way at the receiving nodes and not at the senders. If a node receives another copy of the same packet and did not yet transmit its scheduled packet, i.e., the calculated DFD timer did not yet expire, the node recalculates the additional coverage of its transmission considering the previously received transmissions. From the remaining additional area, the DFD is recalculated which is reduced by the time the node already delayed the packet, i.e., the time between the reception of the first and the second packet. For the reception of any additional copy of the packet, the DFD is recalculated likewise. A node does not rebroadcast a packet if the estimated additional area it can cover with its transmission is less than a *rebroadcasting threshold*, denoted as *RT*, which also may be zero. Obviously, DDB can "only" take locally optimal rebroadcasting decisions as nodes receive only transmissions from their immediate one-hop neighbours and thus have no knowledge of other more distant nodes which possibly already partially cover the same area.

2.6.2 Maximizing Network Lifetime

The objective of extending the network lifetime can be complementary to the objective of minimizing the number of transmissions to reach all nodes. It may be beneficial that more nodes with a lot of residual battery energy broadcast a packet instead of fewer nodes with an almost depleted battery. In scenarios, where the source of the broadcast message is almost uniformly distributed over all nodes in the network or mobility is high and movement patterns are random, we may expect that the traffic load is also uniformly distributed over all nodes, and thus the battery will deplete roughly at the same time at all nodes. However, in many network environments, nodes rarely move and traffic flows are highly directed. This especially applies to sensor networks where all traffic is normally originating from or directed to one or a few designated sinks and the mobility is rather low. If a deterministic algorithm is applied in such a scenario, which does not take into account the battery level at the nodes, the same nodes will always rebroadcast the packet. Consequently, some nodes will deplete much quicker than others will.

In DDB, the calculated delay by DFD depends solely on the residual battery level of a node and does not take into account the additionally covered area and the signal strength. They are only used to determine whether to rebroadcast a packet, i.e., whether they are smaller than *RT*. Nodes with an almost depleted battery schedule the rebroadcasting of the packet with a large delay whereas nodes with a lot of remaining battery power forward the packet almost immediately. Consequently, energy is conserved at almost depleted nodes, which increases their lifetime and in turn extends the connectivity of the network. Therefore, we simply adapt the DFD function to favour nodes with a lot of residual battery energy for rebroadcasting of packets. The DFD function introduces a small delay for nodes with a lot of battery energy whereas nodes with an almost depleted battery add a large delay.

2.7 Ant-Based Mobile Routing Architecture (AMRA)

AMRA is a two-layered framework with three independent protocols rather than a single routing protocol. The two protocols used on the upper layer are called Topology Abstraction Protocol (TAP) and Mobile Ant-Based Routing Protocol (MABR). StPF (Straight Packet Forwarding) is situated on the lower layer and acts as an interface for MABR to the physical network. TAP is the key to make routing scalable and provides in a transparent manner an aggregated and static topology with fixed "logical routers" and fixed "logical links" to MABR. A logical router represents a fixed geographical area. Thus, mobile nodes within a logical router are situated close together sharing similar routing information and have a similar view on the network topology on a large scale. A logical link represents a path along a straight line to another logical router over possibly multiple physical hops. The actual routing protocol MABR operates on top of this abstract topology and thus does not have to cope with changing topologies. MABR maintains probabilistic routing tables and is responsible for determining logical paths on this abstract topology.

Data packets are routed based on these probabilistic routing tables between logical routers over logical links. They increase the probability of the followed path depending on the encountered network conditions. Furthermore, "artificial ants" packets are transmitted periodically to explore new paths. Unlike data packets, these packets are routed purely position-based directly towards their randomly chosen destination. Eventually, the best paths will emerge and MABR is able to circumvent areas with bad or no connectivity, i.e., data packets will always.

2.7.1 Topology Abstraction Protocol (TAP)

TAP is used to supply in a transparent manner an aggregated and static topology with fixed "logical routers" and fixed "logical links". The objective for this abstraction of the actual network topology is to provide a rather static topology such that the routing protocols do not have to cope with frequent changing topologies. Furthermore, it was observed that ant-based algorithm takes some time until good paths emerge. Logical routers are fixed geographical areas of equal size arranged in a grid to cover the whole area. Unlike in a cellular network where regular hexagons are typically used; we use squares for simplicity reasons. Depending on its current position, each node is part of one specific logical router. A node can easily detect based on its position, when it crosses the border of the current logical router and then automatically becomes a member of the new logical router. All nodes located within a logical router have the same logical view on the network. Nodes within a logical router corporate in specific routing control tasks such as the emitting of ants. However, each node maintains its own routing table and does never share with or transmit any routing information to its neighbours.

2.7.2 Straight Packet Forwarding (StPF)

Finally, the physical forwarding process along the logical links selected by MABR is accomplished by StPF, which can be basically any position-based routing protocol. Because many such position-based routing protocols have already been proposed and analyzed in the literature, it is not necessary to design a new protocol for StPF. The process can instead use the perhaps best-known position-based protocol GFG/GPSR as StPF. Basically, any other position-based routing protocol may be applied as well, such as GRA (Jain *et al.*, 2015) and BLR (Heissenbuttel *et al.*, 2014). If BLR is used, nodes would not be required to transmit beacons as MABR and TAP do not require neighbour knowledge.

2.7.3 Looping Packets

Packets are routed based on the routing tables encountered at the nodes. The routing table gives an estimation for the direction in which the packet should be routed advantageously, but they are not guaranteed to be consistent among different nodes, which is definitely a severe drawback of AMRA. Therefore, it is not surprising that in scenarios with high mobility and where a lot of nodes have not very accurate routing tables because the time is often too short for the best paths to emerge, it should be observed that packets may be loop temporarily in the network, sometimes forward and backward between adjacent logical routers or sometimes even over several logical routers.

To mitigate the effect that packets are routed back and forth, a packet must never be sent back to the last visited logical router or one of its two adjacent logical routers. The packet is routed over the logical link with the highest probability among the remaining five possible logical links. Furthermore, logical links may form a loop in which packet can get trapped. Even though this was observed rarely, AMRA implements a mechanism to cope with such situations. In order to prevent such loops, a packet is sent purely position-based if it does not arrive within five times the expected average for this zone.

2.8 Topology-Based Routing Protocols for Ad-Hoc Networks

Topology-based routing protocols make use of information about available links between nodes. This information is then used by the nodes to forward packets. A tremendous number of topology-based protocols have been proposed and thus it is not surprising that taxonomies to categorize them have also been discussed. The perhaps most often employed taxonomy is the classification in proactive, reactive protocols, and hybrid protocols. In proactive protocols, nodes in the network periodically refresh routing information so that nodes have consistent and up-to-date information from each node to every other node in the network at all times. On the other hand, reactive protocols only acquire routes on demand. If a node has to send a packet and does not know a path to the destination, it triggers a route discovery mechanism. Currently, the ”Mobile Ad-Hoc Network (MANET)” working group within the Internet Engineering Task Force IETF (MANET, 2005) is about to standardize two routing protocols, a Reactive MANET Protocol (RMP) and a Proactive MANET Protocol (PMP), using mature components from previous work on reactive and proactive protocols. Hybrid routing protocols employ proactive and reactive concepts for routing within a local and more global scope, respectively. An overview of topology-based protocols and other classifications can be found in (Hong *et al.*, 2012).

2.8.1 Proactive Routing Protocols

In proactive protocols, routes are maintained between hosts pairs at all times by exchanging route information periodically or each time a change occurs in the network topology. Therefore, routes are immediately available if a packet needs to be sent. A shortcoming is the maintenance of unused paths causing large overhead, which wastes scarce network resources. Other issues are scalability and the time required for the algorithms to converge in case of frequent topology changes.

Traditional link-state and distance vector-protocols fall into this category, such as OSPF (Moy, 2008) and RIP (Malkin, 2008). However, these protocols are not designed for the encountered characteristics in ad-hoc networks. Already small inaccuracies in the routing tables can cause disconnections or loops. Furthermore, if topology changes frequently, a storm of link status change messages and triggered updates rises for OSPF and RIP, respectively. In the following sections, we describe in more detail the two proactive protocols for ad-hoc networks, which were upgraded to RFC-status. Other well-known proactive protocols are, e.g., DSDV (Perkins and Bhagwat, 2004), WRP (Murthy and Garcia-Luna-Aceves, 2006), FSR (Pei *et al.*, 2010) and STAR (Garcia-Luna-Aceves and Spohn, 2009).

2.8.2 Optimized Link State Routing (OLSR)

The optimized link state protocol OLSR was proposed in Clausen *et al.*, (2017) (RFC 3626 (Clausen and Jacquet, 2013). OLSR is a variant of the classical link state protocols with optimizations to meet the requirements of ad-hoc networks. The main difference to traditional link-state protocols is the concept of multipoint relays MPRs, which aims at efficiently distributing topology information by reducing the number of required link-state packets. MPRs are a minimal set of one-hop neighbours such that all two-hop neighbours are reachable through these MPRs. Broadcast messages are only forwarded by MPRs. This reduces not only the number of transmissions for topology information broadcasts, but also reduces the size of the broadcast packets, since nodes only need to list their MPRs in the link-state messages. OLSR basically consists of three phases; neighbours sensing based on periodic exchange of hello messages, efficient flooding of control traffic using the MPRs, and computation of an optimal route using a shortest-path algorithm.

2.8.3 Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)

In Bellur and Ogier (2009), Topology Broadcast based on Reverse-Path Forwarding (TBRPF) was described (RFC 3684 (Ogier *et al.*, 2014)) that is similar to OLSR. The main difference to OLSR is that OLSR supports only source trees to its two-hop neighbours whereas in TBRPF each node computes a source tree, which provides paths to all reachable nodes. Each node periodically broadcasts part of its source tree to its neighbours as an update. These updates are not further forwarded but may cause a change in the receiving node's source tree that is again propagated in the next update message. Differential updates are used to minimize the overhead. Neighbour sensing is done with hello messages, which are broadcasted to inform about changes in the neighbourhood topology. Each node is also able to report additional topology information to improve robustness in order to support highly mobile networks.

2.8.4 Reactive Protocols

In reactive protocols, the computation of a path is performed only on-demand. The source initiates route discovery when a path to the destination is needed in order to transmit user data. The advantages of reactive protocols are the power and bandwidth efficiency compared to proactive protocols. However, this point has to be reconsidered because results in (Xu *et al.,* 2011) indicate that battery power consumption is about the same for proactive and reactive protocols. This is due to the fact that just listening to the medium is almost as costly as receiving a packet with today's devices. The main drawback is the long latency until a route is acquired and established between source and destination. We briefly summarize the reactive protocols AODV, which already has RFC-status, and DSR, which is expected soon to be upgraded to RFC-status.

2.8.5 Ad-Hoc On-Demand Distance Vector (AODV) Routing

AODV was proposed in Perkins and Royer, (2009). When a node has to send a packet to a destination for which it does not have a valid route, the node broadcasts a route request message. Each node forwards the route request message and caches the node from which it received the message. If the destination receives a route request message, it sends a route reply back to the originator of the request message establishing a bidirectional route between source and destination node. Consequently, nodes are not aware of the whole path, but only of the immediate next hop towards the destination and the source. Intermediate nodes may also generate a route reply message if they know a route to the destination. Route Error messages are used to notify other nodes of link breaks in existing routes. AODV uses the concept of sequence numbers to avoid the formation of loops. Each destination includes a monotonically increasing sequence number with each route information it sends to a requesting node. A node that has two different routes to a destination always has to use the one with the larger sequence number.

2.8.6 Dynamic Source Routing Protocol (DSR)

The Dynamic Source Routing Protocol (DSR) (Johnson *et al.*, 2011) is currently available as an Internet draft, but is expected to become an RFC later. DSR uses explicit source routing in which each data packet has in its header a complete list of all intermediate nodes to the destination. DSR is composed of two main mechanisms. In route discovery, a node, which attempts to send a packet to a destination and does not know a route, broadcasts a route request packet. Each node that forwards this packet adds its own address to the header. If the destination receives the route request, it sends back a route reply packet containing a copy of the accumulated route along the reverse direction of the path over which the route request arrived. Thus, each node forwarding this reply packet is aware of the whole path from the source to the destination. Nodes cache the route information from each packet they overhear. Intermediate nodes may also reply to a route request if they know a route to the destination. Route maintenance is used to detect if a link along a route is broken. When route maintenance indicates that a route is broken, the source node can either use another route it knows or invoke route discovery again.

2.8.6 Hybrid Protocols

Hybrid protocols apply principles of proactive routing for the local neighbourhood and reactive routing for distant nodes, respectively, for the following two reasons. First, changes in the topology are only important for nodes close-by and have little impact on nodes on the other side of the network. Secondly, most communication takes places between nodes that are close to each other. We briefly describe perhaps the most well-known hybrid protocol ZRP, which was available as an Internet Draft and is planned to be proposed as RFC. Other hybrid protocols include SHARP and LANMAR (Pei *et al.*, 2010).

2.8.7 Zone Routing Protocol (ZRP)

ZRP [96] is not really a routing protocol per se, but rather a framework for hybrid routing. It is composed of the Intrazone Routing Protocol (IARP [Haas and M. R. Pearlman, 2012), the Interzone Routing Protocol (IERP and the Bordercast Resolution Protocol (BRP [89]), where the intrazone and interzone routing protcols can basically be any proactive and reactive routing protocol, respectively. Each node defines its zone as all nodes that are within a certain number of hops, called zone radius. The intrazone routing protocol is used by a node to communicate with the nodes in its zone whereas interzone routing protocol is applied to detect routes to nodes in other zones. If a node requests a route to a distant node, a route request is issued and forwarded by the border resolution protocol. This protocol optimizes the forwarding of the request by making use of the fact that nodes have knowledge of the nodes in their zones. When a node receives a request for a destination, it first checks if the destination is located within its zone. If not, the node forwards the route request but only to its peripheral node by using the border resolution protocol, i.e., the nodes that are at the border of its zone, which have a minimal hop count equal to the zone radius from the current node. Lately, the authors of ZRP proposed the Independent Zone Routing (IZR), an enhancement of the ZRP framework, which allows adaptive and distributed configuration for the optimal size of each node's routing zone radius on a per-node basis.

2.9 Position-Based Routing Protocols for Ad-Hoc Networks

Position-based routing protocols forward packets based on the nodes' physical locations. Forwarding decisions are based solely on the position of the current node, the positions of neighbouring nodes, and the position of the destination. A node that wants to forward a packet to a destination node chooses one of its neighbours as a next hop according to some criterion like the one closest to the destination. When the packet reaches a dead end where this simple greedy routing fails, a recovery mechanism is applied. Unlike topology-based protocols, position-based protocols require only little control traffic and are nearly stateless. Position-based routing protocols do not require the establishment or maintenance of routes and thus eliminate the overhead of frequent topology updates and route acquisitions of topology-based routing protocols. For these reasons, they are generally considered scalable and more robust to changes in the network topology than topology-based protocols. Assuming that location information is available, these characteristics make them a preferred choice for large and highly dynamic networks. Many position-based routing protocols have been proposed in the literature. Unlike for topology-based protocols, only very few taxonomies for position-based routing protocols have been proposed and none of these are widely accepted.

2.9.1 Positioning

Position-based routing protocols require very little position information about the network to forward packets; namely knowledge of the position of the current node, the destination, and the neighbors is needed.

Current Node

The position of the current node can be provided in several ways. The most common and easiest way relies on GPS (McNeff, 2012) or its European counterpart Galileo (Bretz, 2013) to provide location information. The availability of small and cheap GPS receivers that operate on low power provides justification for applying position-based routing in ad-hoc networks. Even sensor nodes equipped with GPS receivers are now available. GPS and Galileo allow nodes to determine longitude, latitude, and altitude with a certain degree of accuracy.

However, in several scenarios, nodes may not be able to receive GPS-signal or are simply not equipped with GPS-receivers. For such scenarios, many approaches have been proposed to provide relative positions of nodes. They either use received signal strength together with the time of arrival (Savvides *et al.*, 2017), the time difference of arrival (Priyantha *et al.*, 2015), or the angle of arrival to estimate nodes' positions. Other methods of determining a node's position are based on radio-location for "hello" messages from an available fixed infrastructure. Furthermore, several hybrid schemes have been proposed (Niculescu and Nath, 2016) where only a few designated nodes are aware of their positions and all other nodes derive their positions by taking into account the hop count to the designated nodes. Routing without absolute location information has become a hot topic recently, where each node computes its virtual position relative to some other nodes, e.g. (Moscibroda *et al.*, 2014, Rao *et al.*, 2013).

2.10 Related Works

The rise of multi-channel multi-radio networks has brought about more and more research works on routing, scheduling and channel allocation is proposed to utilize the channel diversity. Raniwala and Chiueh (2005) proposed a centralized channel assignment and routing algorithm to obtain a static frequency assignment. Kodialam and Nandagopal (2005) developed a network model that characterizes the channel, radio and interference constraint in a fixed broadband wireless network, which provides necessary and sufficient conditions for a feasible frequency assignment and schedule. Meng *et al.* (2006) formulate the joint routing and channel assignment problem based on radio and radio-to-radio link. They introduce a scheduling graph and derive a sufficient condition for the feasibility problem of time fraction.

Raniwala *et al.* (2004) propose an improved distributed frequency assignment algorithm. Kyasanur and Vaidya (2005) propose an interface-assignment strategy where the number of available interfaces is less than the number of available channels. It fixes a channel on one radio and switches channels on other radios. Nodes can communicate with each other through the fixed common radio without requiring specialized coordination algorithms.

Legendre, de Amorim and Fdida (2005) present possible requirements for how initially separate wireless ad hoc networks using different routing protocols could merge. Physically merging wireless ad hoc networks using heterogeneous routing protocols would still be able to provide loop-free routing through the use of a *Neighborhood Routing Protocol Discovery Protocol* (NRPDP). Nodes at the periphery of the merging networks would use an NRPDP to determine the routing protocol used by the other network and translate routing messages from one protocol to another as routing protocol control messages pass from one network to the other. The operational assumption is that the routing protocols use the same routing metric in different forms (e.g. hop count) and does not address how heterogeneous routing metrics like those enumerated by Gelenbe, *et al.*, (2006) could be translated.

Gelenbe, Liu and Laine (2006) present a generic basis for other routing metrics by defining a *goal function* for different paths between a given source and destination node. Example goal functions for a given path include number of packets lost, delay in packet delivery, variance in packet delay, power consumed in forwarding a packet, or overall security level. Router congestion in wireless networks is not addressed. Noise is not explicitly addressed, although it does affect packet delay as shown in the third Chapter of their work.

From the above summary of prior works, it is observed that the joint routing and scheduling problem has not been received much efforts by researchers for the case of ad hoc networks in which nodes employ spectrum-agile radios. This makes this project a very important task at this point in time.

Many broadcast protocols have been proposed in order to cope with the broadcast storm problem and optimize broadcasting in ad-hoc networks. We provide a taxonomy and review existing broadcast algorithms for ad-hoc networks. In a second step, we discuss some general properties and summarize the shortcomings of these broadcasting protocols. A survey of broadcast protocols can be found in (Williams and Camp, 2012, Wu and Dai, 2013). Taxonomy divides the protocols in seven categories ranging from simple flooding to sophisticated protocols that make use of directional antennas. Obviously, several other categorizations are possible and the classification of the protocols may not be unambiguous as some protocols may fall into different categories. From the above summary of prior works, it is observed that the joint routing and scheduling problem has not been received much efforts by researchers for the case of ad hoc networks in which nodes employ spectrum-agile radios. This makes this project a very important task at this point in time.

2.10.1 Some Notable Works

Flooding is the simplest way to broadcast. It was presented in Obraczka *et al.*, (2011) that it might also be the only way to reliably deliver a packet to every node in highly dynamic or very sparse networks. This is not limited to broadcasting, but also holds for multicast and unicast transmissions. In such environments, the overhead of another protocol may be even higher than that of simple flooding to cope with the frequently changing topology and to maintain paths or neighbour's information. In such scenarios, other protocols may not able to deliver the packets at all.

In Ni *et al.*, (2009), each node rebroadcasts a packet with a certain probability p and drops the packet with a probability of $1-p$. If the probability of forwarding a packet is 1, this scheme is identical to simple flooding. Ni *et al.*, (2009) also proposed a counter-based scheme, where a node only rebroadcasts a packet if it has received copies of this packet less frequently than a fixed threshold. In Tseng et al (2013), the threshold is no longer fixed but dynamically adapted to the number of neighbours.

Haas et al (2012) evaluated probabilistic broadcasting in more depth and proposed to account for nodes' neighbour counts and local congestion levels. In Cartigny and Simplot (2016), the authors proposed to adjust the probability with which a node rebroadcasts a packet depending on the distance to the last visited node. The distance between nodes is approximated by comparing the neighbour lists. Probability-based schemes were evaluated theoretically and by simulations in Sasson *et al.*, (2016).

In the location-based schemes proposed in Ni *et al.*, (2009), the forwarding decision is based solely on the position of the node itself and the position of the last visited node as indicated in the packet header. Nodes wait a random time and only forward a packet if the distance to all nodes from which they received a copy of the packet is larger than a certain threshold distance value. The random waiting time is required to give nodes sufficient time to receive redundant packets and to avoid simultaneous rebroadcasts at neighbouring nodes such that only nodes that cover significantly large additional area rebroadcast the packet. Instead of using the distance of nodes as a measure for the additional area covered, Ni *et al.*, (2009) also proposed an area-based method, which directly determines the possible covered area from the distances between nodes. Recently, (Williams *et al.*, 2019) validated the simulation results of Ni *et al.*, (2009) by analytical models.

Neighbour-designated schemes are characterized by the fact that nodes are aware of their neighbourhood. The basic idea in all proposed approaches is that each node selects a set of forwarders among its one-hop neighbours so that all two hop neighbours can be reached through the forwarders. A node only forwards packets from the set of neighbours out of which it was selected as a forwarder, i.e., rebroadcasting nodes are explicitly chosen by upstream senders. Most of these neighbour-designated approaches are quite similar. The multipoint relaying protocol

(MPR) proposed in (Laouiti *et al.*, 2011) is described in more detail. Nodes periodically broadcast beacons including a list of all their neighbours. Consequently, each node has knowledge of its two-hop neighbours. A node selects some of its one-hop neighbours to rebroadcast all packets they receive from it. The chosen nodes are called Multipoint Relays (MPRs). Each MPR also choose a subset of its one hop neighbours to act as MPRs. As always only a subset of all one-hop neighbours rebroadcast a packet, the total number of transmissions is reduced. In order to guarantee that still all nodes in the network receive a broadcasted packet, the MPRs must cover all two-hop neighbours. The protocol select such one-hop neighbours as MPRs that most efficiently reach all nodes within the two-hop neighbourhood, i.e., the one-hop neighbours are a minimal set of neighbours which cover all two-hop neighbours. After a node has selected its MPRs, it lists them in the beacons.

When a node receives a beacon, it checks if it was selected as MPR from this node, and if so, it must rebroadcast all data packets received from that node. In Lu *et al.*, (2015), the set of forwarders excludes all one-hop neighbours that are covered by three or more forwarders. Furthermore, the idea of passive acknowledgments (Jubin and Tornow, 2007) was used to avoid transmission of acknowledgements. Nodes do not send an additional acknowledgment to confirm the reception of a packet, which may become another bottleneck of congestion and collisions called *ACK implosion problem* (Impett and Park, 2010). The rebroadcast of the packet itself is taken as the confirmation and a NACK packet is transmitted in case a node does not overhear the rebroadcasting from all nodes it expected (Impett and Park, 2010). In Lim and Ki (2011) the set of forwarders was reduced by excluding the one-hop neighbours that were already covered by the node from which the broadcast packet was received.

In Lou and Wu (2012), two-hop neighbour information is piggy-backed on packets and permits to eliminate the two-hop neighbours already covered by the last visited node. In (Peng and Lu (2016), the forwarding nodes are selected from a larger set, where a node includes all neighbours of a node with higher priority, e.g., based on IDs, in its cover set. A special class of neighbour-designated approaches are based on connected dominating sets, where only nodes of the dominating set rebroadcast a packet (Kuhn *et al.*, 2014).

Unlike the neighbour-designated method, each node decides for itself on a per packet basis if it should rebroadcast the packet. In Lim and Kim (2011), a node piggy-backs a list of its one-hop neighbours on each broadcast packet and a node only rebroadcasts the packet if it can cover some additional nodes. Several of these approaches are based on (minimal) connected dominating sets.

As the problem of finding such a set is proven to be NP-hard (Marathe *et al.*, 2015), several distributed heuristics are proposed. Wu and Li (2019) proposed an algorithm, which only requires two-hop neighbour information. A node belongs to the dominating set, if two unconnected neighbours exist. Furthermore, two rules are proposed by Wu and Li (2019) to reduce the size of the connected dominating set, which requires an order on the IDs of the nodes. This idea was further improved in Stojmenovic *et al.* (2013), where the degree of a node was used as primary metric instead of their IDs. Unlike Wu and Li (2019) where two-hop information is required, one-hop neighbour information is sufficient if nodes are aware of their positions in order to determine if two neighbours are connected (Stojmenovic *et al.*, 2013). Under the assumption that each node knows its accurate position, connected dominating sets and the concept of planar subgraphs are used in Seddigh *et al.*, (2017) to reduce the communication overhead for broadcast messages.

In Wu and Dai (2013), a generic scheme was proposed based on two conditions, namely on neighbourhood connectivity and history of the already visited nodes. Each node receives information of its k-hop neighbourhood by exchanging $(k - 1)$-hop information with its one-hop neighbours by periodical hello messages. Information about a node's property, such as ID or node degree, and a list of already visited nodes is added to the broadcast packets. Based on this information a node decides whether to forward a packet or not. In Gandhi *et al.* (2016), they show that minimum latency broadcasting is also NP-hard and propose an algorithm where latency and the number of transmissions are within a constant time of their respective optimal values. The algorithm constructs a broadcast tree rooted at the source node and afterwards schedules the transmission times for all nodes following a greedy strategy. To be able to cope more efficiently with mobility, Wu and Li (2019) proposed to use two different transmission ranges for the determination of forwarders and for the actual broadcast process. The difference between these two transmission ranges is based on the update-frequency and the node movement. They further proposed a mechanism to ensure consistency between the different views of different nodes on the network. A comprehensive performance comparison of several of these broadcast protocols based on self-pruning is given in Wu and Dai (2013).

The algorithms discussed in Wu and Li (2019) above can be also considered as energy-efficient as they aim to reduce the total number of transmissions to deliver the packet to every node in the network. Thus, they also reduce the total energy consumption at the same time. In case nodes can adjust their transmission power however, the number of transmissions may not be proportional to the energy spent. Transmissions to close-by nodes cost much less energy than transmissions too far away nodes, especially in view of realistic path loss factors of approximately four (4).

Consequently, several transmissions over multiple short hops may save energy compared to one transmission over a long distance.

The problem of transmitting a packet energy-efficiently to all nodes in the network where nodes have adjustable transmission radii was considered in several papers. Wieselthier *et al.*, (2015) proposed a broadcast incremental power algorithm, which constructs a tree starting from the source node and adds in each step a node not yet included in the tree that can be reached with minimal additional power from one of the tree nodes. In Wan *et al.* (2014), theoretical bounds on the performance of the broadcast incremental power algorithm of Wieselthier *et al.* (2015) were provided. Chakeres and Belding-Royer (2016) considered the minimum energy broadcasting problem and proposed a localized protocol where each node requires only the knowledge of the positions of itself and the neighboring nodes. This eliminates two drawbacks of Wieselthier et al (2017) as the algorithm requires almost global knowledge to construct the tree efficiently and it is difficult to maintain in case of mobility. Cagalj *et al.* (2014) showed the NP-completeness of minimal power broadcast. They also proposed a distributed algorithm for energy efficient broadcast. Starting from an initial link-based minimal spanning tree, the total energy to maintain the connectivity of this broadcast tree is reduced gradually by exchanging some existing branches by new branches.

In Kang and Poovendran (2016) it was shown that minimizing the total transmit power does not maximize the overall network lifetime. Note that energy efficiency is not necessarily directly related to network lifetime. If the same nodes always forward packets, broadcasting may be energy-efficient, but the battery at these nodes depletes quickly. In Kang and Poovendran (2016), the algorithm constructs a static routing tree, which maximizes network lifetime by accounting for residual battery energy at the nodes. Wattenhofer *et al.*, (2016) presented a distributed topology control algorithm, which extracts network topologies that increase network lifetime by reducing the transmission power. A comparison of several power-efficient broadcast routing algorithms is given in Kang and Poovendran (2016).

2.10.2 Discussion

Different kinds of broadcast protocols show different kinds of advantages and shortcomings. Comprehensive comparison studies were conducted in Williams and Camp (2012). The majority of the proposed protocols are either neighbour-designated, self-pruning, or energy-efficient schemes that all belong to the stateful protocols. That means they require at least knowledge of their one-hop neighbours, sometimes even global network knowledge is required. Therefore, they also show similar drawbacks as stateful position-based unicast routing protocols, which require local neighbour information to forward packets. Like position-based unicast routing protocols, stateful broadcast protocols require the proactive distribution of information with hello messages.

The proactive communication and computation overhead of these protocols consumes unnecessarily scarce network resources like battery power and bandwidth. These costs occur even if no packets are broadcasted. Furthermore, their performance suffers significantly in highly dynamic networks as the frequent topology changes induce an excessive, or even prohibitive, amount of control traffic, which occupies a large fraction of the available bandwidth. Furthermore, stateful algorithms may also never converge and reach a consistent state, if changes occur too frequently. Their inability to cope with frequent topology changes together with the proactive transmission of control messages, which wastes network resources, make stateful protocols unsuitable for certain kind of ad-hoc networks such as sensor and vehicular ad-hoc networks. Stateful protocols also have some distinct advantages. Packets can be forwarded almost immediately without introducing additional delay and they are barely affected by high traffic loads and collisions as shown in Williams and Camp (2012).

The probability- and location-based schemes, as well as simple flooding belong to the category of stateless algorithms, as they do not require nodes to have any neighbour knowledge. As they do not maintain neighbour information, they are almost immune to frequently changing network topologies. A further advantage is their simplicity. The main drawbacks of stateless protocols are twofold Williams and Camp (2012). First, the number of rebroadcasting nodes can be disproportionately high in networks with a high node density. Secondly, the random delay introduced at each node before rebroadcasting a packet is highly sensitive to the local level of congestion. The main reason for this is that these stateless protocols use fixed parameters, e.g., a distance-threshold, to rebroadcast a packet. These algorithms are highly sensitive to the chosen threshold values and may perform well in some scenarios, and very poorly in others. For example, packets may die out in sparse networks and the number of transmissions may not be reduced significantly in dense networks for too low and too high parameter values, respectively.

CHAPTER THREE

METHODOLOGY

3.1 Chapter Introduction

This chapter describes the methodology that was employed to develop a model for routing scheduling in ad hoc networks and selection of the simulation parameters for obtaining reliable simulation results. Figure 3.1 shows the architecture framework of the model which involves the communication pair selection, the measurement period, the selection of number of simulations, the selection of a mobility model, the performance metrics and the analyses. Figure

3.2 shows the process steps towards route-discovery. The main focus of the model development phase is discussed as follows.

Firstly, to highlight relevant simulation parameters that determine the simulation results, and to provide guidelines on the selection of these parameters to obtain a low disperse set of samples in order to obtain reliable statistics. In the next subsection of this chapter, we describe in more detail each step of the procedure. MATLAB 2020 with the GUI was used for the simulation of the model development.

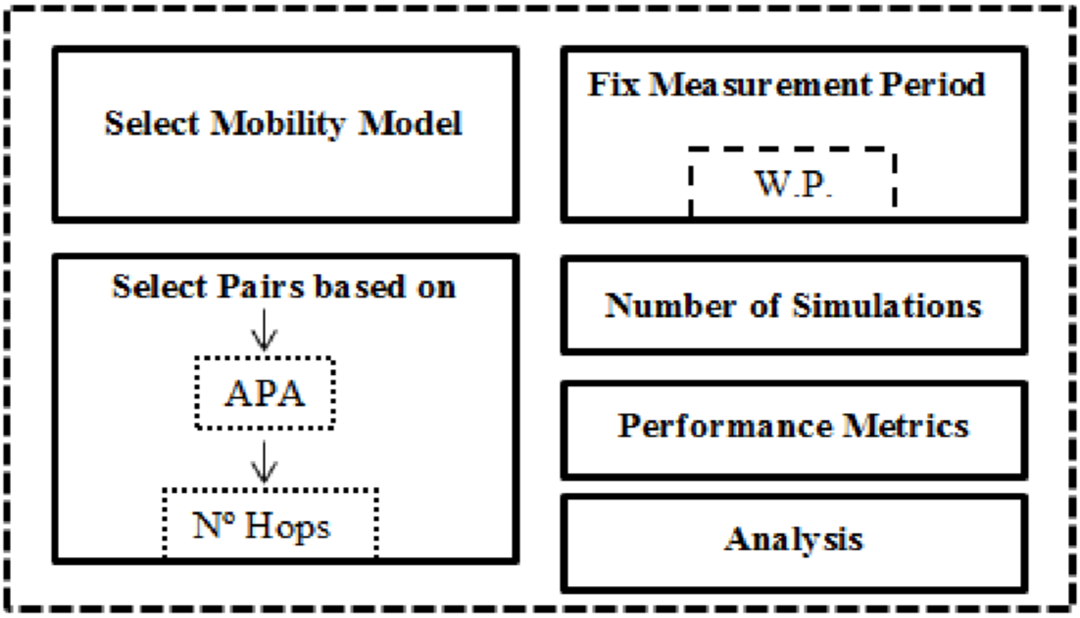

Figure 3.1 Architectural framework of the Model

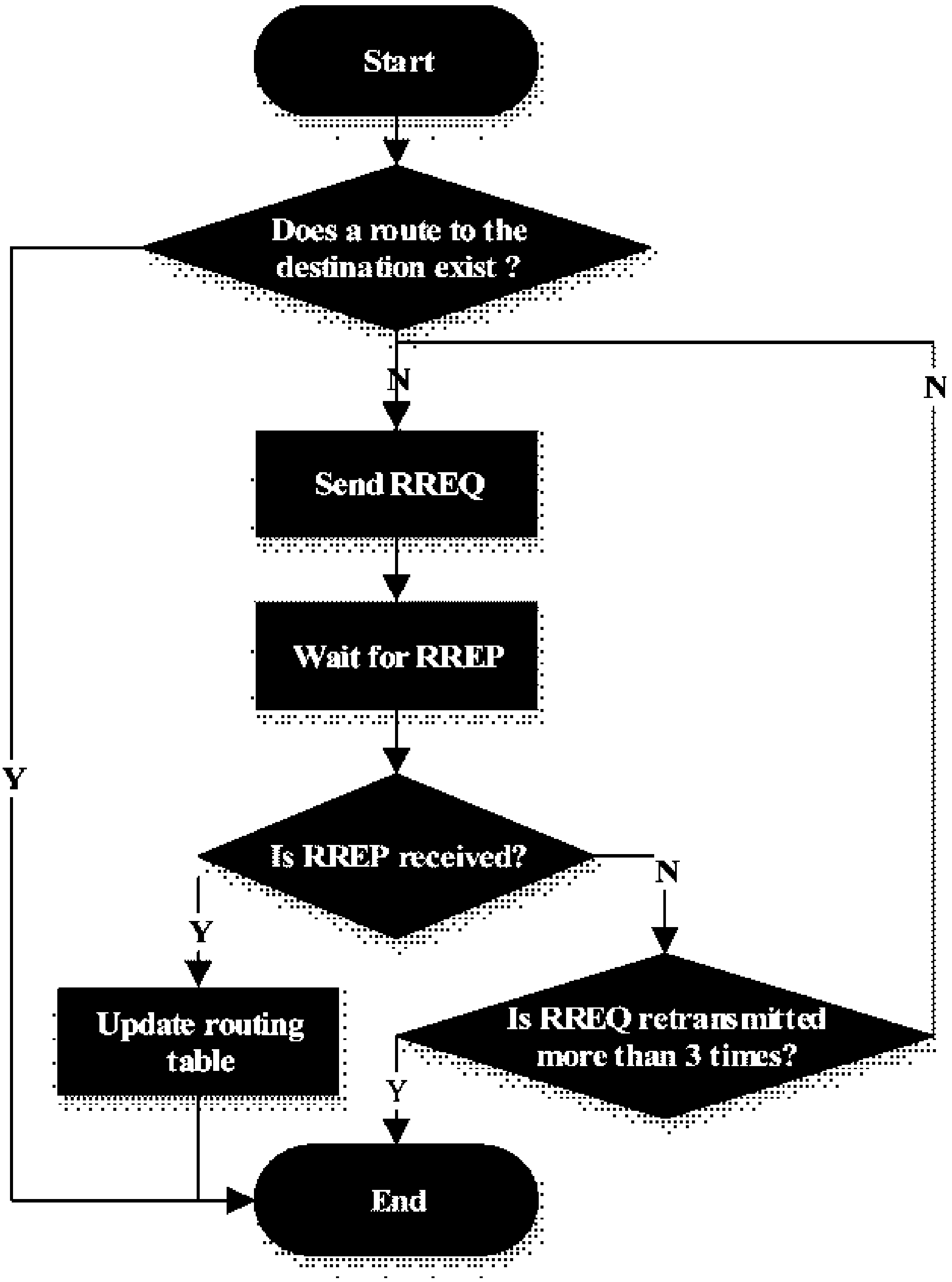

Figure 3.2 Flowchart for route discovery process

3.2 Measurement Period

Here, an important concept to carry out good simulations for VANETs was introduced. The Warm Up period (W.P. in Figure 3.3) which is the time frame which ensures the stability of three relevant simulation aspects such as:

- All communication pairs have started transmitting application packets.
- The mobility model has achieved a stable state.
- The buffers of the nodes have stabilized.

It should be noticed that the establishment of communications among a source-destination pair can start at different times. Normally, the starting times are selected randomly, so some pairs could have more time to transmit data packets than others. This fact can influence the simulation results if the selected pairs do not have the same properties in terms of average number of hops and path availability between the source and the destination nodes. By using a Warm Up period, we avoid discrepancies among the measurement period of the performance metrics during the simulation time. To obtain reliable and non-dispersed simulation results, performance metrics must be measured from W.P. value to the end of the simulation period, which is named after Measurement Simulation Period (M.S.P. in Figure 3.3).

In order to select the Warm Up period, we have to consider two aspects, the first one is the period during which the communication flows are established, and the second aspect is the mobility model, since we need to guarantee that the mobility of nodes is stable.

Figure 3.3 Status time bar

The throughput performance metric was used as standard metric. It is defined as the number of application packets delivered within the simulation time. In the simulations, all source nodes start generating application packets from the period between 0 and 50 seconds until the end of the simulation.

3.3 Source-Destination Selection

Here, we present a communication pair selection based on four features that strongly affect the simulation results such as the path availability, the separation in terms of the number of hops between the source and destination nodes, and the repetition of source and destination nodes. The communication pair selection as the mechanism by which the source and destination nodes of a communication flow are selected was defined. The source node is responsible for generating the data packets, and the destination node is the target node in the network for those generated packets. Consequently, intermediate nodes will route the generated data packets towards the destination node using routing information.

In simulation analyses, the communication pairs are normally selected at the beginning of the simulations. In most simulation-based studies of routing protocols for VANETs, the source and destination nodes are selected randomly among all nodes of the network. Although the original aim of this practice is achieving a fair selection of pairs, this can impact negatively on the dispersion of the obtained simulation results for several reasons. First, by using a random selection we cannot guarantee that all source destination pairs have similar properties in terms of number of hops and path availability. Consequently, the simulation results may vary drastically from one pair to another. It is expected that routing protocols will obtain worse results when the number of hops increases and the path availability is lower.

This situation is even more aggravated if the source-destination pair selected cannot be established. This means that it is not possible to establish a communication path from the source node to the destination node during the simulation time. Furthermore, the performance of the routing protocols can also be biased if the number of hops is very low. Second, outliers are prone to appear when random selection is applied because of the great variability of the results. This affects to the mean of the simulation results. To solve this problem, the Average Path Availability (APA) and the number of hops between the source and destination nodes was used as the key metrics to select source-destination pairs.

The APA metric is defined as the fraction of time during which a path is available between two nodes. We select source-destination pairs which have similar APA values because pairs with different APA values produce very dissimilar results. It can be observed that high values of APA are more probable than low values in the scenario under test but also that there are some APA values which are zero. This situation corresponds to source-destination pairs that cannot be established.

3.4 Number of Simulations

Another important aspect to be considered when conducting simulation-based studies is the number of simulations that should be carried out for each data point in the results. Clearly, the more simulation trials, the more representative data sample that can be obtained. However, the simulation results also incur in computing time consumption. Consequently, a trade-off between the number of simulations and the computing time should be reached. More time should not be devoted than the necessary to conduct simulations.

Thus, the number of simulations should be selected in order to obtain a representative data sample without requiring excessive simulation time. There is a throughput results and the required computing time for different number of traffic seeds (number of simulations). The simulation scenario is the same one described in the previous section (the scenario under test) with 25 source-destination pairs of communications. As expected, we can observe that the computing time is higher as the number of seeds increases as well. When we use the proposed pair selection and the measurement period, the computing time is lower than when we do not use them. The main reason is that the network is less congested because the number of routing packets is lower, due to the APA based selection. Since we aim to obtain reliable simulation results, we want a good confidence interval. This means obtaining non-dispersed results with the lowest computing time. There are two different cases, the first one corresponds to the proposed communication pair selection, and the second one is for the case of using random pairs.

The obtained results are not similar in terms of the obtained average mean. By using the methodology, we obtain better results than not using it. In addition, the proposed communication pairs selection needs a lower number of simulations to obtain better and lesser disperse results, as it is shown by the lower confidence intervals. To highlight the importance of our methodology in terms of computing time consumption, the 5 seed case was focused on in which a very good confidence interval was achieved without requiring excessive simulation time. A new metric to evaluate this situation was defined as the ratio between the confidence interval and the mean. From now on, it will be named after normalized confidence interval. If the proposed measurement period and the communication pair selection mechanism is used, the normalized confidence interval takes the value of 0.01 for 5 seeds. The number of simulations to reach this value without the measurement period and the proposed communication pair selection mechanism is 80 simulations. It means that we need 16 times more simulations to obtain the same dispersion. In terms of computing time, we save 279.3 s (287.9 s – 8.6 s) when using the model.

3.5 Mobility in VANETs

A critical issue in VANETs simulation is the need of a mobility model that reflects the real behaviour of vehicular traffic in urban scenarios. There are some mobility generators that are able to create mobility patterns that emulate such scenarios, some of them are VanetMobSim, SUMO, FreeSim and CityMob. In this model use CityMob for Roadmaps C4R was used as the selected mobility generator, which allows the simulation of vehicular traffic in different locations using real maps. C4R uses two tools to generate the mobility model. On the one hand, it uses Open Street Map to get the real roadmaps and SUMO to generate the vehicles and their movements within the scenario.

The functionally provided by C4R is twofold: it defines the vehicle movements on the streets, and it limits their mobility according the vehicular congestion and traffic rules to simulate the vehicle movements in a VANET scenario, C4R provides the following mobility models: Krauss, Krauss modified, Wagner, Kerner, Downtown model and Intelligent driver model (IDM). Moreover, C4R allows users to modify some parameters to customize the mobility model, such as the attraction rate, downtown rate, departure, simulation time or number of traces.

The next objective of the model (simulation) is to determine how to select a representative VANET scenario to evaluate routing protocols. According to the classification made in the cities can be categorized according to the density of their streets and junctions as simple, regular and complex layouts. The APA has been studied and the number of hops values found in these layouts. It can be observed that in general the number of hops is higher for more complex layouts. Regarding the APA distribution, it is observed that similar distributions for regular and complex layouts, where APA values higher than 0.5 are more probable.

3.6 Performance Metrics

Another important aspect to be considered when evaluating routing protocols is which performance metrics should be used in order to represent an unbiased performance of the routing protocols. It is important to use metrics that exhibit the performance of the routing protocols in different conditions. The following performance metrics considered in this simulation of routing scheduling in ad hoc networks.

Throughput (THR): It is the sum of the data packets in the simulation period.

$$\boldsymbol{THR(Kbps)} = \frac{\sum \boldsymbol{Delivered\ Application\ Packets}}{\boldsymbol{Simulation\ time}} \qquad (\mathbf{3.1})$$

Average End-to-End Delay (*E2E*): It is defined as the time taken for a data packet to be transmitted across an ad hoc network from the source to the destination node.

$$\boldsymbol{E2E(s)} = \frac{\sum(\boldsymbol{Delivered\ time} - \boldsymbol{Transmitted\ time})}{\boldsymbol{Number\ of\ packets\ successfully\ delivered}} \qquad (\mathbf{3.2})$$

Normalized Routing Load (*NRL*): It is the ratio of the total routing packets to the total delivered data packets.

$$NRL = \frac{\sum Routing\ Packets}{\sum Delivered\ Application\ Packets} \qquad (3.3)$$

Packet Delivery Fraction (PDF): It is the ratio of the number of packets delivered to the receiver, to the number of packets sent by the source.

$$PDF(kbps) = \frac{\sum Delivered\ Application\ Packets}{\sum Sent\ Packets} \qquad (3.4)$$

Jitter (*JIT*): It is the delay between two consecutive packet deliveries at a node.

$$JIT = \frac{\sum delays\ packet\ delivered}{Packet\ Application\ delivered} \qquad (3.5)$$

Additionally, in this work a new performance metric is proposed, the Route Activity Time (RAT), which is aimed at evaluating the capability of a routing protocol to maintain an active route between the source and the destination nodes. The formal definition of RAT is as follows:

Route Activity Time (*RAT*): It is the period of time during which a communication path is available between the source and the destination nodes. In routing protocols based on request, reply and error messages, such as AODV, DSR, and DYMO, it is the period elapsed between the time at which the reply message arrives at the source node and the time at which an error message of such route is generated.

$$R(s) = Error_{time} - Reply_{time} \qquad (3.6)$$

Notice that the RAT metric measures how the routing protocols manage the path availability. In theory, we control the APA values by selecting the communication pairs, however, the real time in which a communication path is established between a source and a destination node will depend on the underlying routing protocol and the network conditions.

Table 3.1 summarizes the desirable values for each metric used to evaluate the routing protocols performance.

Table 3.1 Desirable values for the performance metrics

Metric	THR	E2E	NRL	PDF	JIT	RAT
Desirable values	***High***	***Low***	***Low***	***High***	***Low***	***High***

Key:

THR = Throughput

E2E = Average End to End Delay

NRL= Normal Routing Load

PDF=Packet Delivery Fraction

JIT = Jittery

RAT= Route Activity Time

Although six different performance metrics (1)-(6) have been considered to evaluate the performance of the routing protocols, only four of them were used, THR(1), E2E(2), NRL(3) and RAT(4), since the rest of them provide equivalent information. By using the THR metric, the performance of the routing protocols is measured in terms of the number of delivered packets. With E2E, the average delay of the application packets was evaluated. Using the proposed RAT metric, how the routing protocols maintain the communication routes between the source and destination nodes was measured. The NRL metric measures the number of routing packets used by the routing protocols and provides an idea about the power consumption of the routing protocol. Regarding PDF and JIT metrics, these metrics for the following reasons were not used. First, with the PDF metric also the number of delivered packets were measured so this metric will show the similar results of THR metric. Second, the JIT metric was not used because it gives us an idea about the network delay and we are actually using E2E to measure this performance.

The objective at this point is to decide which analyses should be carried out for obtaining a good performance evaluation of routing protocols. In general, the number of nodes is a common parameter to vary in simulation-based studies in order to evaluate routing protocols under different density levels (connectivity). However, there are other parameters that also affect considerably the performance of routing protocols. For instance, the congestion is a common issue in multi-hop networks because nodes should share the wireless medium, and consequently, routing and application packets should compete for the wireless medium. The congestion in the network can be modified by varying some parameters of the communication flows between the source and destination nodes such as data rate, size of packets, and number of flows.

Among the mentioned parameters, focus was on the number of flows since in this project work relevance is being given to the selection of source-destination pairs. Therefore, two different analyses are proposed. First, a density analysis based on varying the number of nodes while maintaining the same number of communication flows. Second, a congestion analysis, focused on varying the number of communication flows while maintaining the same number of nodes. With the first analysis, the routing protocols under different connectivity levels and with low congestion conditions were evaluated. In the second analysis, a medium-high value of density (high connectivity) was set and vary the congestion of the network to observe how routing protocols perform under different levels of congestion.

To summarize the procedure described in this section, Table 3.2 provides the most important values of the model and also the benefits of each of them.

Table 3.2 Summary of the simulation parameters used in the methodology

Simulation parameter	Selection	Benefit obtained
S.P. = 300 s	W. P.= 50 s (M.S.P.=250 s)	Using W.P. we improve mean of the used performance metrics
Selection Pairs based On APA	Depending on the scenario and based on APA target	Applying the methodology based on APA and Number of hops we reduce the dispersion.
Selection Pairs based On Hop Number	Depending on the scenario and based on number of hops target	
Performance Metrics	THR(1), E2E(2), NRL(3), RAT(6)	different features of the evaluated routing protocol can be evaluated
Scenario	Washington (Regular layout)	Real scenarios can be emulated.
Analysis	Congestion and density	The routing protocols under different network conditions can be evaluated.

Key:

THR = Throughput

E2E = Average End to End Delay

NRL= Normal Routing Load

PDF=Packet Delivery Fraction

JIT = Jittery

RAT= Route Activity Time

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 Implementation Environment

The implementation of model for routing scheduling in ad-hoc network was done using the 2015 edition Matrix Laboratory (MATLAB R2015a) from MathWorks Inc. MATLAB is an integrated development environment (IDE) for developing primarily with matrices. The MATLAB Platform allows applications to be developed from a set of modular software components called modules. Applications based on the MATLAB Platform can be extended by third-party developers. MATLAB is not an open-source integrated development environment. MATLAB supports the development of all console and graphical user interface application types. There are so many in-built functions in MATLAB, most function made use of in this project were called. Among other features are an Ant-based project system, Maven support, refactoring, and version control (supporting CVS, Subversion, Git, Mercurial and ClearCase).

Every function of IDE is provided by a module, each module provides a well-defined function, such as support for the language, editing, or support for the CVS versioning system, and SVN. MATLAB contains all the modules needed for application development in a single download, allowing the user to start working immediately. Modules also allow MATLAB to be extended. New features, such as support for other programming languages, can be added by installing additional modules. Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again. The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are User interface management (e.g. menus and toolbars), User settings management, Storage management (saving and loading any kind of data), Window management, Wizard framework (supports step-by-step dialogues), MATLAB Visual Library and Integrated development tools.

4.2 Installation of MATLAB

MATLAB (R2015a) version was used to write the codes for the model for routing scheduling in the ad-hoc network. The programming language was installed on a laptop. The development, deployment and testing of this work was carried out on an ASUS ROG STRIX HERO EDITION Computer with the following hardware and software specifications;

- 1TB Hard disk
- 16 Gigabytes RAM (Random Access Memory)
- 2.8 Gigahertz Processor Speed (Intel Core i7).
- 64-bit Operating System (Windows 10)

The installation process was done step by step, following the prompts displayed by the application setup until the installation was complete. The MATLAB home and code editor windows are shown in figures 4.1 and 4.2 on the next two pages respectively.

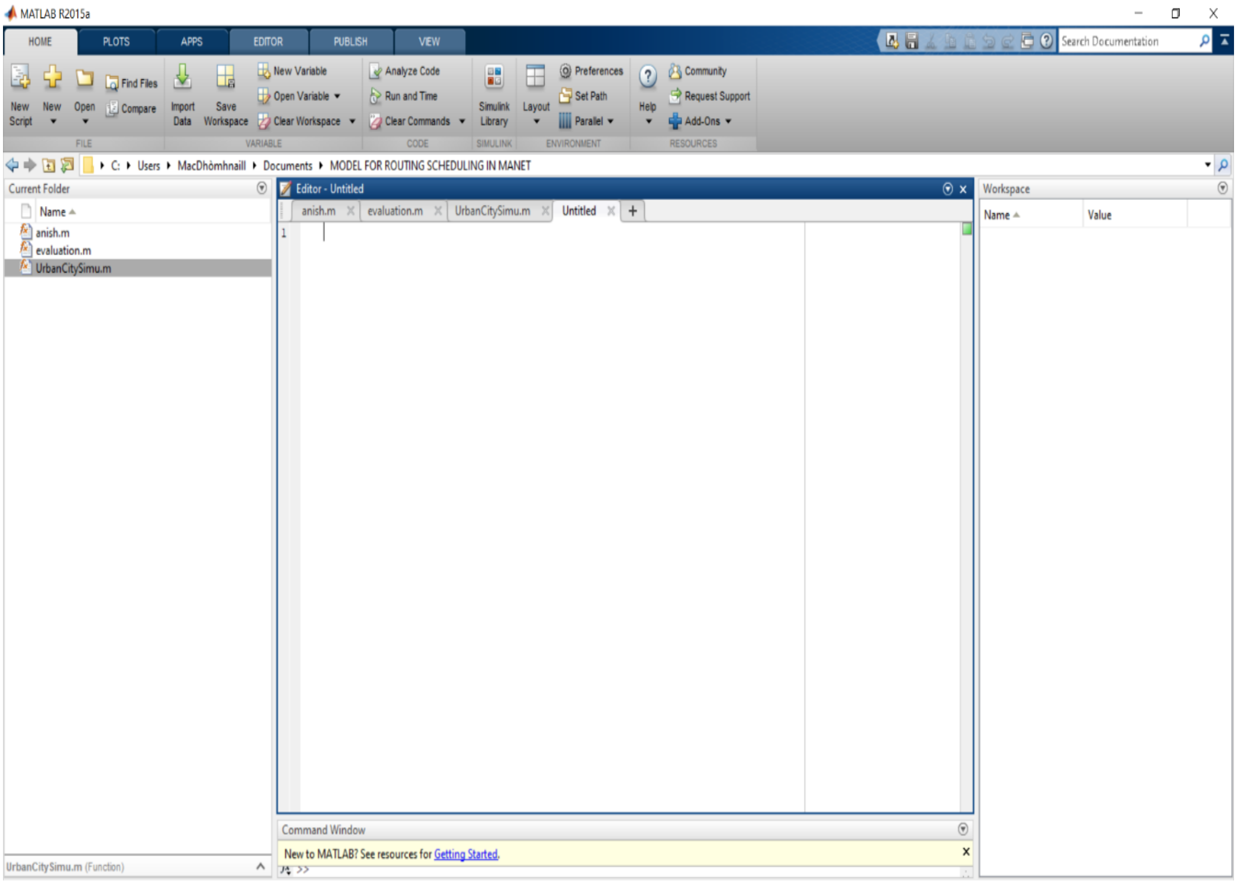

Figure 4.1 MATLAB R2015a home window

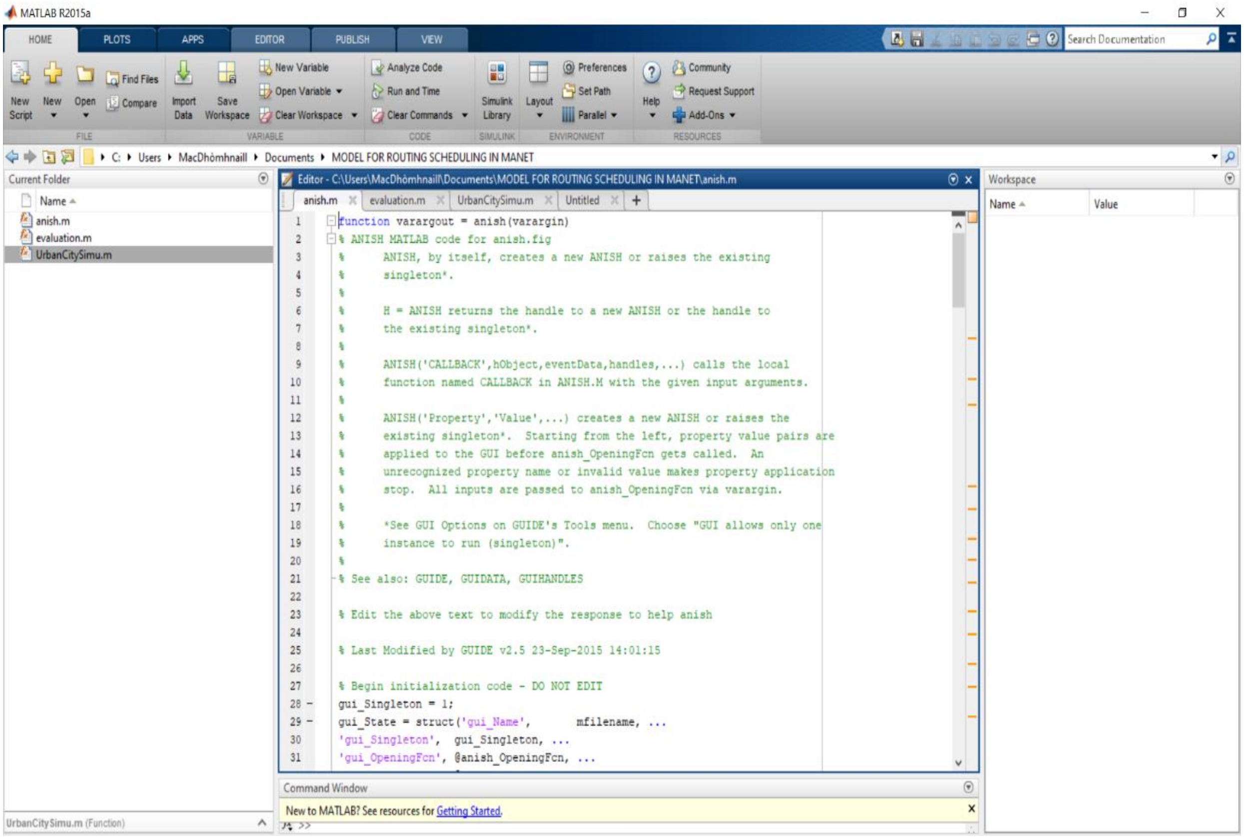
MATLAB R2015a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Editor - C:\Users\MacDhòmhnaill\Documents\MODEL FOR ROUTING SCHEDULING IN MANET\anish.m
Current Folder
anish.m
evaluation.m
UrbanCitySimu.m
Workspace
function varargout = anish(varargin)
% ANISH MATLAB code for anish.fig
% ANISH, by itself, creates a new ANISH or raises the existing
% singleton*.
% H = ANISH returns the handle to a new ANISH or the handle to
% the existing singleton*.
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help anish
% Last Modified by GUIDE v2.5 23-Sep-2015 14:01:15
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
Command Window
New to MATLAB? See resources for Getting Started.

Figure 4.2 MATLAB R2015a Code Editor Window

4.3 Description of the Developed Model for Routing Scheduling in Ad-Hoc Network

The software model developed for the routing scheduling in an ad-hoc network on MATLAB IDE can be loaded by copying it into the Hard Disc Drive of the PC or simply accessing the folder from MATLAB root folder. The folder is to be accessed via the MATLAB IDE for it to run properly. Once the project folder directory has been loaded via MATLAB, click on the anish.m file which is the code file for the VANET GUI page developed for the simulation. On the GUI simulation page, there are text boxes for the number of nodes, the source node and destination node. The boxes must be filled appropriately; the total number of nodes, the source node and destination node figures must be set accordingly such that the source node and the destination node are within the range of the total number of nodes. For instance, if the total number of nodes is set to be 80, the source node must be less than 80. Also, the destination node as well must not be more than 80. Should a mistake be made and the given principles be flouted, the GUI app would throw an error that the reads 'node out of range'. Also, the difference between the source node and destination node should be at least 20. This figure '20' was programmed into the code to ensure that users would not be able to simulate with the destination node and source node that is not at least 20 nodes apart. Having two nodes that are not up to 20 nodes apart would not give an optimum simulation result as there would not be enough values to plot the necessary graphs needed for analysis.

Once the right values are set for the total number of nodes, the source node and a destination node, the simulation process can begin by clicking on the ‘simulate’ button, the model begins to establish paths among the nodes that were set. Four graphs are automatically plotted immediately after the model has completely established all the possible paths according to the set nodes. The ‘clear all figure’ tab removes all the node figures established on the VANET simulation interface while the ‘close all’ tab closes all the plotted graphs and established paths.

The first graph plotted is the total distance in each linked path. The distance in a path for each destination and source node positions was plotted against the number of times path found in the VANET simulation. The second graph is the energy consumption graph, which shows the energy used by the node to transmit the data. The energy (in Joules) was plotted against the number of times path found in the VANET simulation. It showcases the result of energy consumed by the network when using different packet size (i.e. the number of times path found). The graph reflects the higher the packet size, the lower the energy used due to the fact of high packet collision and packet corrupted. Packet corruption rate increase as packet size increase due to the packet fragmentation. Packet fragmentation is a process that split the packet into smaller pieces so that packets can pass through a link with a maximum transmission unit compared to the original size.

Since the energy sources have a limited lifetime, power availability is one of the most important constraints for the operation of the ad hoc network. The third graph is the network lifetime graph which is obtained by plotting the network lifetime against the number of times path found in the VANET simulation. The graph shows different outcomes with varied data rate (4, 6, 8, 10, 12 and 14 packets per second). The link duration has a significant influence on the route lifetime, which, in turn, determines the packet delivery ratio (data rate) and the per-connection throughput for a given source-to-destination pair in the network.

The final graph is the throughput graph obtained by having throughput (in Mbps) plotted against the number of times path found in the VANET simulation. Throughput is the average rate of a successful packet that can be delivered to its destination over a communication channel per unit of time. The result of throughput (Mbps) versus a different number of packet size depicts that an increasing number of packet rates and packet size, increase the amount of data injected into the network. This data injection leads to an increase in throughput. The larger the packet rate, the larger the throughput and the larger the packet size, the larger the throughput. Throughput and packet delivery ratio is directly proportional to each other. When the packet delivery ratio increase, the throughput also increases. It also shows that, when the packet size or packet rate increase, the packet delivery ratio also increases. Figures 4.3 to 4.8 show the VANET simulation GUI interface and the four graphs described and interpreted above.

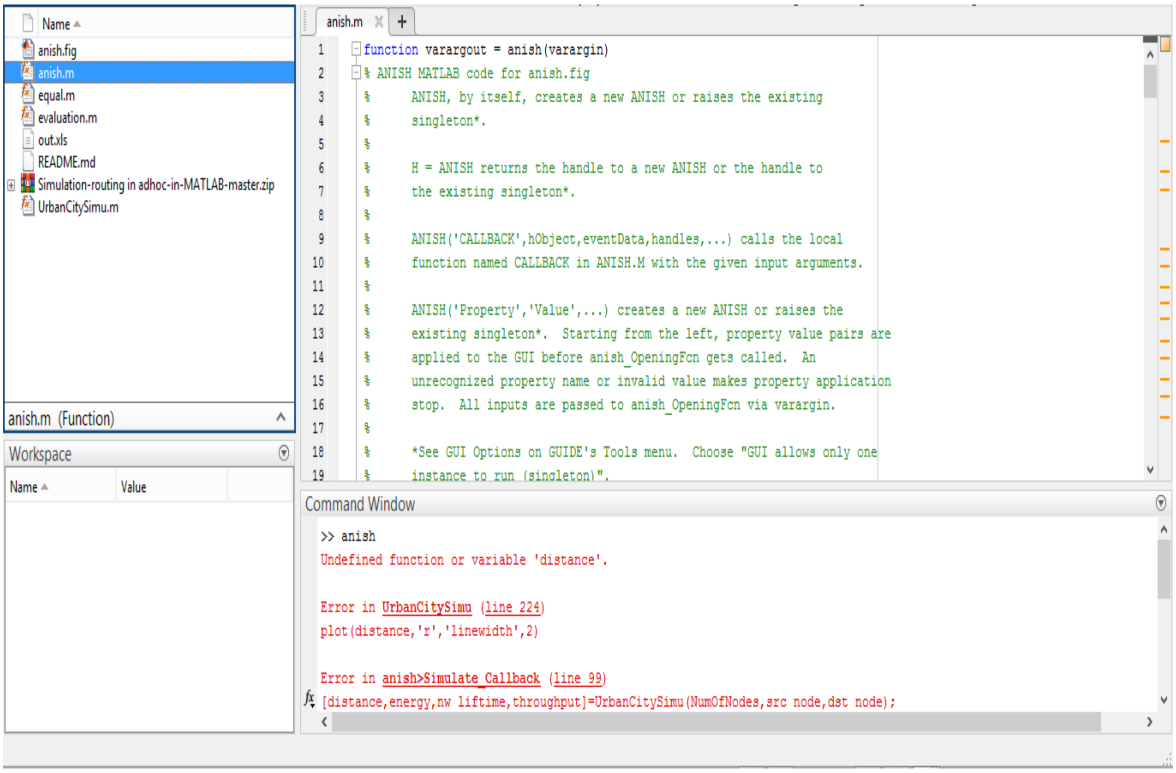

Figure 4.3: The command window for VANET Simulation

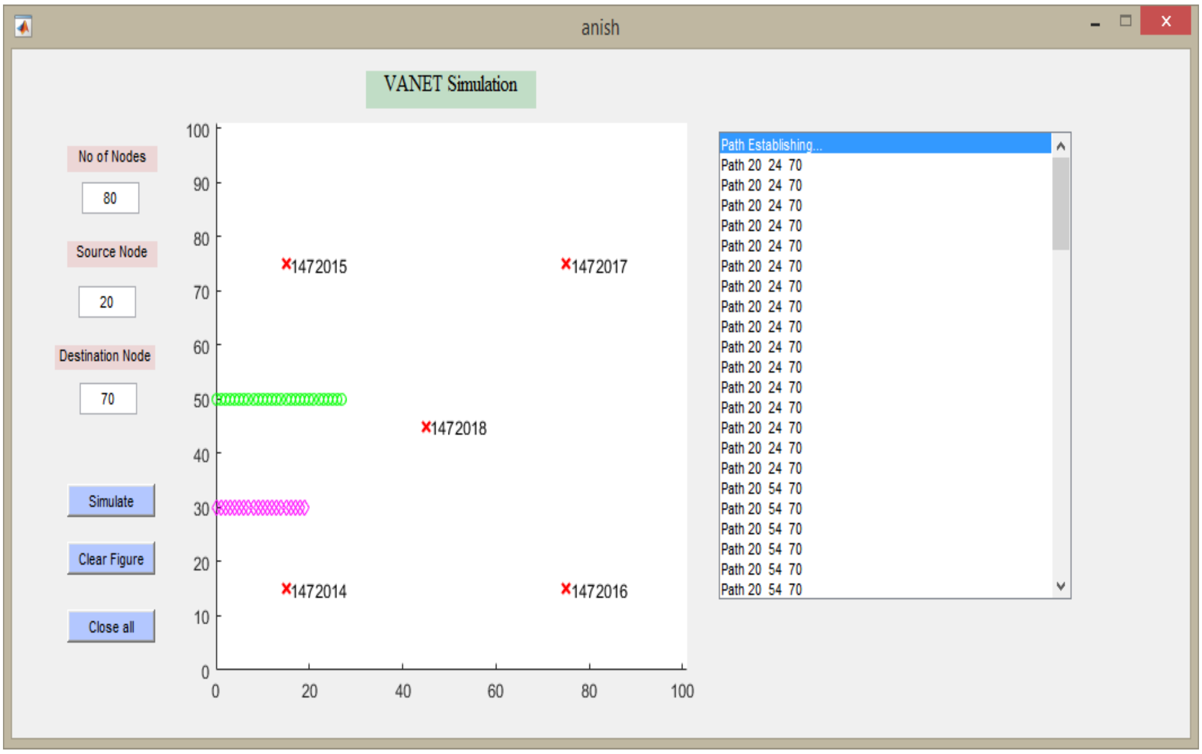

Figure 4.4: VANET Simulation Window

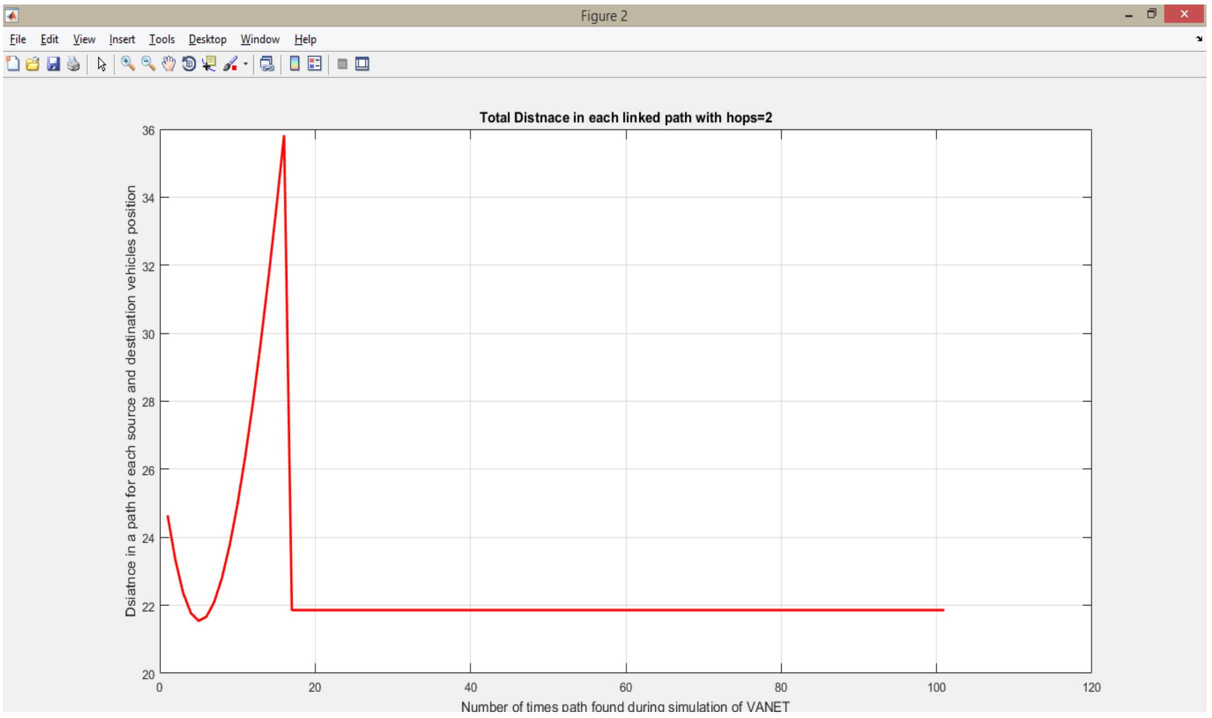

Figure 4.5: Linked Path Total Distance Graph.

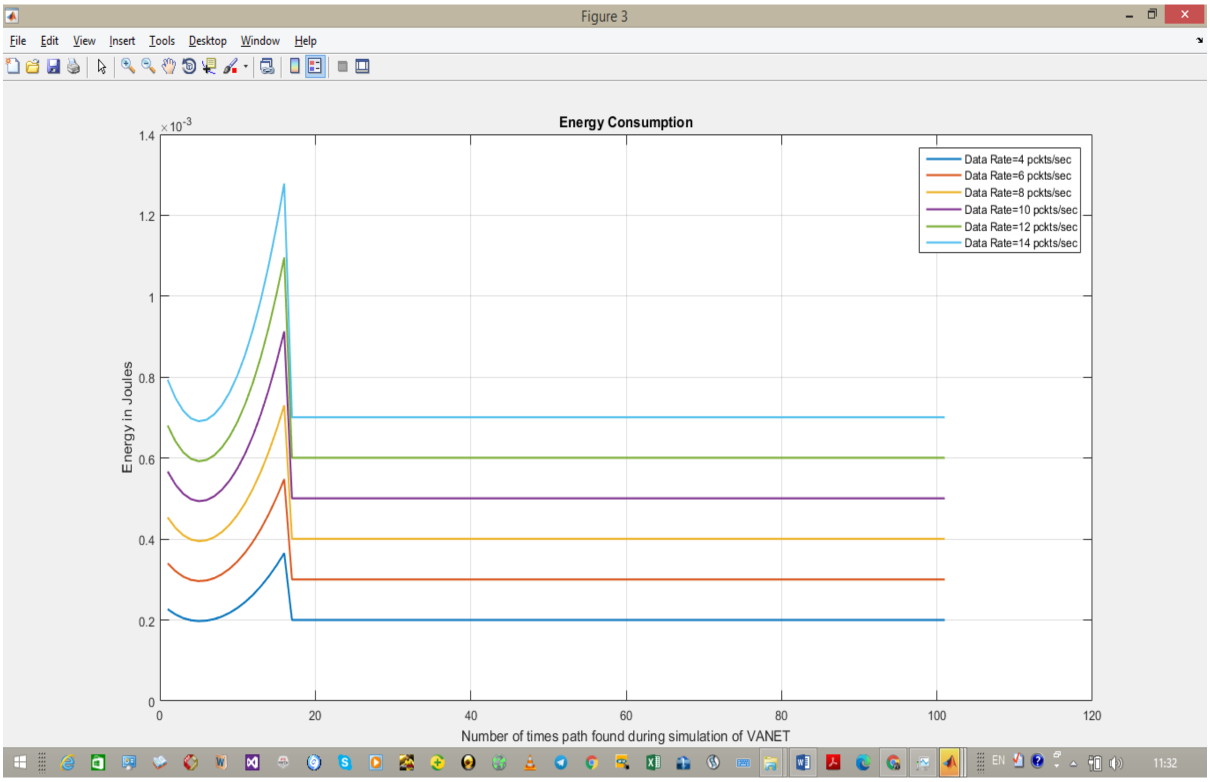

Figure 4.6: Energy Consumption Graph

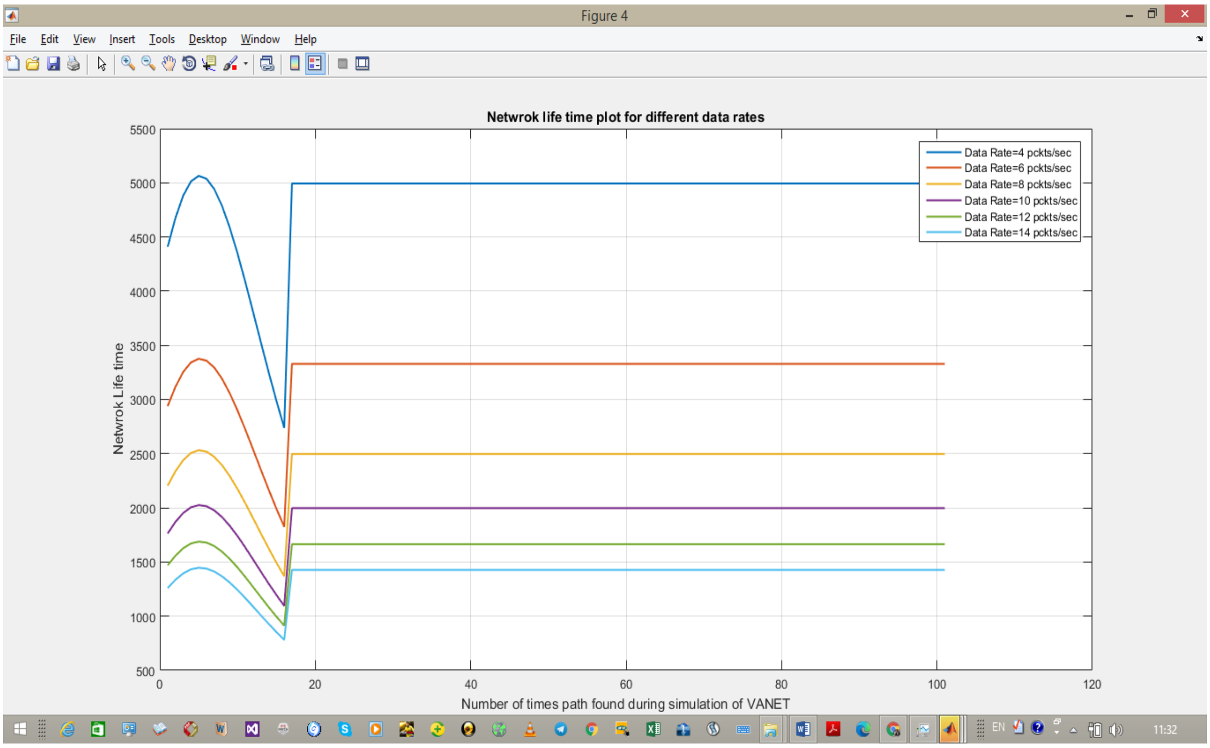

Figure 4.7 Network Lifetime for data rate graph

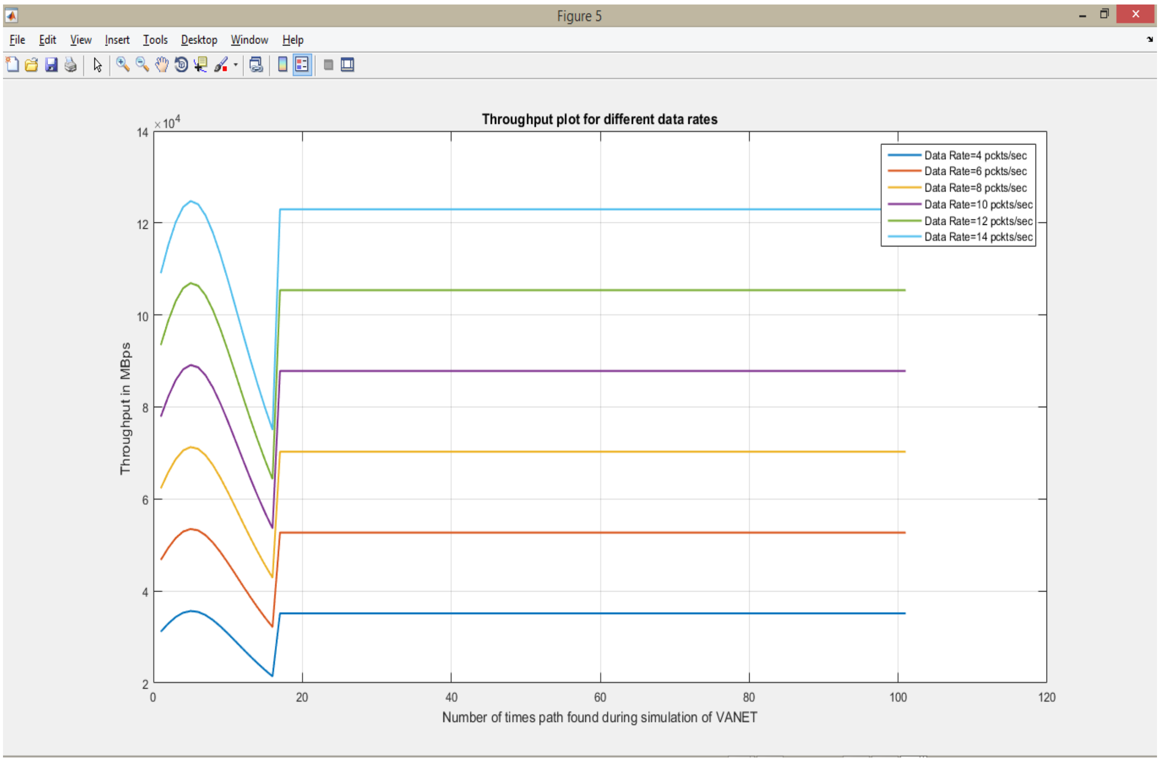

Figure 4.8: Throughput graph

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATION

5.1 Summary

As it was proposed in chapter one, the main focus of the work was to develop a model for routing scheduling in ad hoc networks with the objectives of designing the model and implementing the model on MATLAB. At the beginning of the work, the background of study, statement of the research problem, aim and objectives of the study, the layout of the entire work was presented. In chapter two, the concept of ad-hoc network and other related subjects such as routing, broadcasting and scheduling were examined, taking perspectives from researchers who have looked at the subject from various angles. The research work was conducted to determine a model for routing and scheduling in an ad-hoc network and to implement the model on

MATLAB; a framework that will ensure accurate, effective and efficient routing and scheduling framework in an ad-hoc network. In carrying out the work, most of the information was gathered from books, journals and past research papers related to the subject of ad-hoc network concept, routing, source-destination selection, scheduling etc.,

In chapter three, the research techniques were discussed i.e. the methods followed in the model development on the MATLAB IDE. This chapter fleshes out the research steps ranging from the MATLAB graphical simulation to the implementation of the algorithm, it fleshes out information on how the MATLAB solution was developed, maintain it and also on how to ensure that the dataset saved in the database are well utilized. The chapter also discusses what is needed to put codes behind the MATLAB developed model, i.e. how to implement the developed model.

The project work was carried out in two layers. The first was the development of the model. The second was the implementation of the developed model which entails the writing of codes that specified the model for routing scheduling in an ad-hoc network, as well as the graphs plot MATLAB IDE. The implementation was carefully done such that the aim and objectives of the routing scheduling in ad-hoc network aim of this work and associated objectives as stated in chapter one was proudly met. The tool used for carrying out the implementation is MATLAB (R2015a), which was used to hand-code the model from scratch. The developed model is efficient and dynamic enough to avoid unnecessary basic functionality worries associated with ill-developed MATLAB implementations and simulations.

5.2 Conclusion

In this work, scheduling and routing protocols for ad-hoc networks that make use of location information were studied. Routing protocols and one broadcast protocol were proposed and designed. Their way of operation and behaviour were evaluated analytically as well as by graphical simulations. An in-depth was an investigation on the impact of beaconing and inaccurate neighbour information on position-based protocols was first researched into by considering the Average Path Availability (APA) and the number of hops between the source and destination nodes, which was used as the key metrics to select source-destination pairs. The attempts to deliver packets to unavailable neighbours introduce significant additional delays and consume a substantial amount of scarce network resources such as bandwidth. However, the main flaw of these protocols remains, namely their stateful-ness about the neighbourhood. The network topology may change too frequently in certain kinds of ad-hoc networks so that information about neighbours is never accurate. Consequently, forwarding nodes have to select a next hop based on outdated neighbour information.

The simulation results prove that the dynamic transmission range gives better energy consumption compared to the static transmission range, so it is worth it to carry out the subsequent experiments on VANET. Although the result of packet delivery ratio, throughput and network lifetime from this experiment shows that the dynamic transmission range is lower compared to the static transmission range. In VANET the information may be directly sent to the destination since the nodes frequently change the location. The nodes did not need to use the intermediate node to send or receive the information. Hence less packet maybe drops or loss during the communication process compared to these experiments. This is because by using dynamic transmission range on static mobility ad-hoc network in these experiments, the transmitted data need to use intermediate node to reach the destination. The transmission range of the dynamic transmission range is assigned based on the distance between the nodes. Hence, more data might be dropped and lost during the transmission process since the data need to travel from one node to another node to reach the destination. This is the reason why the packet delivery ratio, throughput and delay for the dynamic transmission range in this experiment give a lower result.

Existing position-based routing protocols may be appropriate for many scenarios but have also significant shortcomings in others. First, these protocols need to keep track of their local neighbourhood, which requires resource-consuming transmissions of hello packets. In dynamic networks with frequently changing topologies, it is also hardly possible to maintain accurate information about the neighbourhood. Routing protocols have to operate on outdated neighbours' positions and the network performance degrades significantly. As the second drawback, it was identified that existing position-based dynamic routing protocol forward packet solely on local neighbour information. Especially in large networks, the locally optimal choice to forward a packet may be highly suboptimal on the global scale. Therefore, these protocols may especially be appropriate in sensor and/or vehicular ad-hoc networks.

5.3 Recommendation

In this section, we briefly discussed possible future directions of research and some more or less obvious possible extensions and optimization to the proposed protocols. There are many possible further optimizations for the implemented model for routing scheduling protocol. They may improve the performance and reliability in certain scenarios but may perform poorly in others and cause new problems. It is an inherent property of the complex simulations of ad-hoc networks that they are very sensitive to the parameter values of the simulations, and also of the protocols. Therefore, simulation results can solely indicate the possible potential of a proposed protocol but cannot provide hard evidence for its superior, or inferior, performance. This statement also holds for the simulations conducted in this work.

More models other than the one implemented would have been analysed and compared to arrive at the best model for routing scheduling in an ad-hoc network, but due to time constraint, this was not done. Therefore, it is highly recommended that more models should be formulated and compared as future routing scheduling protocols as a major area of further study for the next set of researchers on the subject matter as a means of improving on what has been achieved in this project work.

References

Ad-hoc Network Systems (ans) 2005. Research Subgroup. Internet Research Task Force (IRTF). [Online]. Available: http://www.flarion.com/ans-research/

Akyildiz I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 57, no. 8, pp. 102–114, Aug. 2017.

Alicherry, M., Bhatia, R., & Li, L. (2005). Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. In Proceedings of ACM Mobicom

Bellur B. and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '09)*, New York, USA, Mar. 2009, pp. 178–186.

Bluetooth. 2010. Bluetooth core specification v1.2. Bluetooth Special Interest Group (SIG). [Online]. Available: https://www.bluetooth.org/speccore specification v1.2. Bluetooth Special Interest Group (SIG). [Online]. Available: https://www.bluetooth.org/spec

Bretz E., "Precision navigation in European skies," *IEEE Spectrum*, vol. 50, no. 9, p. 16, Sept. 2013

Bruno R., M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 123–131, Mar. 2015.

Cagalj M., J.-P. Hubaux, and C. Enz, "Energy-efficient broadcasting in all-wireless networks," *ACM/Baltzer Mobile Networks and Applications*, 2014, to appear.

Cartigny J. and D. Simplot, "Border node retransmission based probabilistic broadcast protocols in ad-hoc networks," *Telecommunication Systems*, vol. 35, pp. 189–204, Apr. 2016.

Chakeres I. D. and E. M. Belding-Royer. "Localized minimum-energy broadcasting in ad-hoc networks," in *Proceedings of the 35nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '16)*, San Francisco, CA, USA, Mar. 2016, pp. 2210–2217.

Clausen T. and P. Jacquet, “Optimized link state routing protocol (OLSR),” RFC 3626, Internet Engineering Task Force IETF, Oct. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt.

Clausen T., P. Jacquet, P. Muhlethaler, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol,” in *Proceedings of IEEE International Multi-topic Conference (INMIC ’01)*, Lahore, Pakistan, Dec. 2017.

Erol Gelenbe, Peixiang Liu, and Jeremy Laine. (2006). Genetic algorithms for autonomic route discovery. In *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006.

Feeney L.M., B. Ahlgren, and A. Westerlund, “Spontaneous networking: An application oriented approach to ad hoc networking,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 176–181, June 2019.

Franck Legendre, Marcelo Dias de Amorim, and Serge Fdida. *Some Requirements for Autonomic Routing in Self-Organizing Networks*. Autonomic Communications. Springer Berlin/Heidelberg, 2005.

Gandhi R., S. Parthasarathy, and A. Mishra, “Minimizing broadcast latency and redundancy in ad hoc networks,” in *Proceedings of the 17th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc ’16)*, Annapolis, Maryland, USA, June 2016, pp. 222–232

Garcia-Luna-Aceves .J.J and M. Spohn, “Source-tree routing in wireless networks,” in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP’09)*, Toronto, Canada, Oct. 2009.

Haas Z. J and M. R. Pearlman, “A hybrid framework for routing in ad hoc networks,” in *Ad Hoc Networking*. Addison-Wesley, 2012, pp. 221 – 253

Haas Z. J, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” in *Proceedings of the 31st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’12)*, New York, USA, June 2012, pp. 1707 – 1716

Heissenbuttel M., T. Braun, T. Bernoulli, and M. W¨alchli, “BLR: Beaconless routing algorithm for mobile ad-hoc networks,” *Elsevier’s Computer Communications Journal (Special Issue)*, vol. 37, no. 11, pp. 1076–1086 , July 2014

Hong X., K. Xu, and M. Gerla, “Scalable routing protocols for mobile ad hoc networks,” *IEEE Network*, vol. 26, no. 4, pp. 11–21, July 2012

IEEE 802.20 working group for mobile broadband wireless access (MBWA). The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available:http://grouper.ieee.org/groups/802/20/

IEEE, 2010. IEEE 802.11b standard for local and metropolitan area networks. The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

IEEE, 2010. IEEE 802.16 standard for local and metropolitan area networks. The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ,USA. [Online]. Available: http://www.ieee802.org/16/published.html

Jain R., A. Puri, and R. Sengupta, “Geographical routing using partial information for wireless ad hoc networks,” *IEEE Personal Communications Magazine*, vol. 28, no. 1, pp. 48–57, Feb. 2015.

Johnson D. B., D. A. Maltz, and J. Broch, “DSR: The dynamic source routing protocol for multihop wireless ad hoc networks,” in *Ad Hoc Networking*. Addison-Wesley, 2011, ch. 5, pp. 139–172

Johnson D. B., D. A. Maltz, and J. Broch, “DSR: The dynamic source routing protocol for multihop wireless ad hoc networks,” in *Ad Hoc Networking*. Addison-Wesley, 2017, ch. 5, pp. 139–172.

Jubin J. and J. D. Tornow, “The DARPA packet radio network protocols,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 21–32, Jan. 2007.

Jubin J. and J. D. Tornow, “The DARPA packet radio network protocols,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 21–32, Jan. 2007.

Kang I. and R. Poovendran, “A comparison of power-efficient broadcast routing algorithms,” in *Proceedings of IEEE Global Telecommunications Conference (Globecom 2016)*, San Francisco, CA, USA, Dec. 2016.

Kellerer W., C. Bettstetter, and C. P. Sties, “(auto) mobile communication in a heterogeneous and converged world,” *IEEE Personal Communications Magazine*, vol. 24, no. 6, pp. 41–47, Dec. 2017.

Kodialam, M., & Nandagopal, T. (2003). Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem. In Proceedings of ACM Mobicom, pp. 42–54

Kodialam, M., & Nandagopal, T. (2005). Characterizing the capacity region in multi-radio, multi-channel wireless mesh networks. In Proceedings of ACM Mobicom, pp. 73–87

Kuhn F., T. Moscibroda, and R. Wattenhofer, “Initializing newly deployed ad hoc and sensor networks,” in *Proceedings of the 20th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’14)*, Philadelphia, USA, Sept. 2014

Kyasanur, P., & Vaidya, N. H. (2005). Routing and interface assignment in multi-channel multi-interface wireless networks. In IEEE WCNC.

Laouiti A., A. Qayyum, and L. Viennot, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” in *Proceedings of the 44th Annual Hawaii International Conference on System Sciences (HICSS-44)*, Hawaii, USA, Jan. 2011

Lim H. and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks,” in *Proceedings of the 14th ACM international workshop on Modelling, analysis and simulation of wireless and mobile systems (MSWIM 2011)*, Boston, Massachusetts, United States, Aug. 2011, pp. 61–68

Lou W. and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 111–122, Apr. 2012.

Lu G., Z. Liu, and Z. Zhou, “Multicast routing based on ant algorithm for delay-bounded and load balancing traffic,” in *Proceedings of the 40th Annual IEEE Conference on Local Computer Networks (LCN’15)*, Tampa, FL, USA, Nov. 2015, pp. 362–368

Malkin G. S, “RIP version 3,” RFC 2453, Internet Engineering Task Force IETF, Nov. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc2453.txt

Macker J. P. and M. S. Corson, “Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations,” RFC 2501, Internet Engineering Task Force IETF, Jan. 2019. [Online]. Available: http://www.ietf.org/rfc/rfc2501.txt

Macker J. P. and M. S. Corson. 2016. “Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations,” RFC 2501, Internet Engineering Task Force IETF, Jan. 2016. [Online]. Available: http://www.ietf.org/rfc/rfc2501.txt

Marathe M., H. Breu, H. Hunt III, S. Ravi, and D. Rosenkrantz, “Simple heuristics for unit disk graphs,” *Networks*, vol. 45, no. 59-68, 2015

Matthew Impett M.S.C. and V. Park, “A receiver-oriented approach to reliable broadcast in ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’10)*, Chicago, IL, USA, Sept. 2010, pp. 117–122.

McNeff J., “The global positioning system,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 12, pp. 645–652, Mar. 2012.

Meng, X., Tan, K., & Zhang, Q. (2006). Joint routing and channel assignment in multi-radio wireless mesh networks. In Proceedings of IEEE ICCMesh Networks. 2015. [Online].Available: http://www.meshnetworks.com/

Mobile Ad-hoc Networks (manet) 2005. Working Group. Internet Engineering Task Force (IETF). [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

Moscibroda T., R. O’Dell, M. Wattenhofer, and R. Wattenhofer, “Virtual coordinates for ad hoc and sensor networks,” in *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing (DIALMPOMC’14)*, Philadelphia, Pennsylvania, USA, Oct. 2014

Moy J. T., “OSPF Version 3,” RFC 2178, Internet Engineering Task Force IETF, Apr. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc2328.txt

Murthy S. and J. J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” *ACM/Baltzer Mobile Networks and Applications*, vol. 10, no. 2, pp. 183–197, Oct. 2006, Special Issue on Routing in Mobile Communication Networks.

Navas J. C. and T. Imielinski, “Geo-cast - geographic addressing and routing,” in *Proceedings of the 23rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’17)*, Budapest, Hungary, Sept. 2017, pp. 66–76.

Niculescu D. and B. Nath, “Ad-hoc positioning system,” in *Proceeding of IEEE Global Communications Conference (Globecom 2016)*, San Antonio, Texas, USA, Nov. 2016, pp. 2926–2931Ni S.Y., Y. C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proceedings of the 15th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’09)*, Seattle, USA, Aug. 2009, pp. 151 – 162

Obraczka K., K. Viswanath, and G. Tsudik, “Flooding for reliable multicast in multi-hop ad hoc networks,” *Wireless Networks*, vol. 17, no. 6, pp. 627–634, Nov. 2011

Ogier R., F. Templin, and M. Lewis, “Topology dissemination based on reverse-path forwarding (TBRPF),” RFC 3684, Internet Engineering Task Force IETF, Feb. 2014. [Online]. Available:http://www.ietf.org/rfc/rfc3684.txt

Pei G., M. Gerla, and T.-W. Chen, “Fisheye state routing in mobile ad hoc networks,” in *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS ’00)*, Taipei, Taiwan, Apr. 2010 , pp. 71–78

Pei G., M. Gerla, and X. Hong, “LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility,” in *Proceedings of the 11th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc ’10)*, Boston, USA, Aug. 2010, pp. 11–18

Perkins C.E. and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '09)*, New Orleans, USA, Feb. 2009, pp. 90 –100.

Perkins C. and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM/SIGCOMM '04 Conference on Communications Architecture, Protocols and Applications*, London, UK, Aug. 2004, pp. 234 – 244

Postel J., "Transmission control protocol," RFC 793, Internet Engineering Task Force IETF, Sept. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

Powers R. A. 2015. "Batteries for low power electronics," *Proceedings of the IEEE*, vol. 103, no. 4, pp. 687 – 693, Apr. 2015.

Priyantha N., A. Chakraborty, and H. Padmanabhan, "The cricket location support system," in *Proceedings of the 16th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '15)*, Boston, MA, USA, Aug. 2015, pp. 32–42

Preuss S. and C. Cap, "Overview of spontaneous networking-evolving concepts and technologies," in *Proceedings of Future Services For Networked Devices (FuSeNetD 2019)*, Heidelberg, Germany, Nov. 2019.

Raniwala, A., & chker Chiueh, T. (2005). Architecture and algorithms for an Ieee 802.11-based multi-channel wireless mesh network. In IEEE INFOCOM, March

Raniwala, A., Gopalan, K., & Chiueh, T. C. (2004). Centralized algorithms for multi-channel wireless mesh networks. ACM SIGMOBILE Mobile Computing and Communications Review, 8(2), 50–65

Rao A., C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic routing without location information,” in *Proceedings of the 9th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’13)*, San Diego, CA, USA, Sept. 2013, pp. 96–108.

Richard G.G., “Service advertisement and discovery: enabling universal device cooperation,” *IEEE Internet Computing*, vol. 21, no. 5, pp. 18 –26, Sept. 2017.

Peng W. and X.-C. Lu, “Efficient broadcast in mobile ad hoc networks using connected dominating sets,” *Journal of Software*, vol. 26, no. 7, Apr. 2015

Sasson Y., D. Cavin, and A. Schiper, “Probabilistic broadcast for flooding in wireless mobile ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’16)*, New Orleans, Louisiana, USA, Mar. 2016, pp. 1124–1130

Savvides A., C.-C. Han, and M. Srivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *Proceedings of the 23th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’17)*, Rome, Italy, July 2017.

Schulzrinne H., S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” RFC 3550, Internet Engineering Task Force IETF, July 2013. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

Seddigh M., J. S. Gonzalez, and I. Stojmenovic, “RNG and internal node based broadcasting algorithms for wireless one-to-one networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 21 , no. 2, pp. 37–44, Apr. 2017.

Stark W., H. Wang, A. Worthen, S. Lafortune, and D. Teneketzis, “Lowenergy wireless communication network design,” *IEEE Wireless Communications Magazine,* vol. 10, no. 4, pp. 60–72, Aug. 2012.

Stojmenovic I., M. Seddigh, and J. Zunic, “Dominating sets and neighbour elimination-based broadcasting algorithms in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, no. 1, pp. 14 – 25, Jan. 2014

Stine J. A. and G. de Veciana, “A paradigm for quality-of-service in wireless ad hoc networks using synchronous signalling and node states,” *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 7, pp. 1301 – 1321, Sept. 2014.

Tseng Y.-C., S.-Y. Ni, and E.-Y. Shih, “Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network,” *IEEE Transactions on Computers,* vol. 62, no. 5, pp. 545–557, May 2013.

Wan P.-J., G. Calinescu, X.-Y. Li, and O. Frieder, “Minimum-energy broadcast routing in static ad hoc wireless networks,” in *Proceedings of the 42th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’14)*, Anchorage, USA, Apr. 2014, pp. 1162–1171

Wattenhofer R., L. Li, P. Bahl, and Y. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” in *Proceedings of the 35th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’16),* Anchorage, USA, Apr. 2016, pp. 1388–1397

Wieselthier J. E., G. D. Nguyen, and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks,” in *Proceedings of* *the 34th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '15)*, Tel Aviv, Israel,, Mar. 2015, pp. 585–594

Williams B. and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 13rd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '12)*, Lausanne, Switzerland, June 2012, pp. 194–202,

Williams B., D. P. Mehta, T. Camp, and W. Navidi, "Predictive models to rebroadcast in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 295–303, Sept. 2019

Wu J.and F. Dai, "Broadcasting in ad hoc networks based on selfpruning," in *Proceedings of the 32nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '13),* San Francisco, USA, Mar. 2013, pp. 2240– 2250

Wu .J. and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '19)*, Seattle, USA, Aug. 2019 , pp. 7–14

Xu Y., J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 17th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '11)*, Rome, Italy, July 2011, pp. 70–84

Zhang, J., Wu, H., Zhang, Q., & Li, B. (2005). Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In Proceedings of Broadnets, pp. 631–640

APPENDIX

Source Code

A. Main GUI

```
function varargout = anish(varargin)

% ANISH MATLAB code for anish.fig

%      ANISH, by itself, creates a new ANISH or raises the existing
```

```
%      singleton*.
%
%      H = ANISH returns the handle to a new ANISH or the handle to
%      the existing singleton*.
%
%      ANISH('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in ANISH.M with the given input arguments.
%
%      ANISH('Property','Value',...) creates a new ANISH or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before anish_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to anish_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help anish

% Last Modified by GUIDE v2.5 23-Sep-2015 14:01:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```

```
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @anish_OpeningFcn, ...
'gui_OutputFcn',  @anish_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before anish is made visible.
function anish_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to anish (see VARARGIN)
```

```
% Choose default command line output for anish
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes anish wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = anish_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in Simulate.
function Simulate_Callback(hObject, eventdata, handles)
% hObject    handle to Simulate (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
NumOfNodes=str2double(get(handles.edit1,'string'));
src_node=str2double(get(handles.edit2,'string'));
if isnan(src_node)
src_node=round(1+(NumOfNodes-1).*rand);
end
if src_node>NumOfNodes
errordlg('Source ID is more than Number of nodes','Index Exceeds')
return
end

dst_node=str2double(get(handles.edit3,'string'));
if isnan(dst_node)
dst_node=round(1+(NumOfNodes-1).*rand);
end
if dst_node>NumOfNodes
errordlg('Destination ID is more than Number of nodes','Index Exceeds')
return
end
[distance,energy,nw_liftime,throughput]=UrbanCitySimu(NumOfNodes,src_node,dst
_node);
if exist('out.xls','file')
delete out.xls
end
```

```
xlswrite('out',distance','Distance')

xlswrite('out',energy,'Energy')

xlswrite('out',nw_liftime,'Network Life Time')

xlswrite('out',throughput,'throughput')

winopen('out.xls')




% --- Executes on selection change in listbox3.

function listbox3_Callback(hObject, eventdata, handles)

% hObject    handle to listbox3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Hints: contents = cellstr(get(hObject,'String')) returns listbox3 contents as cell array

%        contents{get(hObject,'Value')} returns selected item from listbox3



% --- Executes during object creation, after setting all properties.

function listbox3_CreateFcn(hObject, eventdata, handles)

% hObject    handle to listbox3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in Clear.
function Clear_Callback(hObject, eventdata, handles)
% hObject    handle to Clear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% h=handles.axis1;
cla



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end




function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all
```

B. Urban City Simulation

```
function
[distance,energy,nw_liftime,throughput]=UrbanCitySimu(NumOfNodes,src_node,dst_node)
```

```
citysize=100;
axis([0 citysize+1 0 citysize+1]);
hold on
blksiz=30;
Eini=1;% in joules
% Range=(3*(blksiz/2))/2;
Range=20;
breadth = 0;
display_node_numbers = 1;
% src_node=5;
% src_node=round(1+(NumOfNodes-1).*rand);
src_node1=src_node;
% dst_node=round(1+(NumOfNodes-1).*rand);
% dst_node=35;
%% uicontrol
H = uicontrol('Style', 'listbox', ...
'Units', 'normalized', ...
'Position', [0.6 0.2 0.3 0.68], ...
'String', {'Path Establishing...'});
drawnow;
%               pause(1.0);
%%
%%%%%%%%%%%%creating road network%%%%%%%%%
for len = 0:citysize
if(rem(len,10)~=0)
```

```
for breadth = 0:citysize
if(rem(breadth,10)~=0)
h1 = plot(len,breadth,':g');
end
end
end
breadth = breadth+1;
end
%%%%%%%%%%%%%%%%%END1%%%%%%%%%%%%%%%%%%%%%%%
%%%%
Node = zeros(NumOfNodes,6); % 1:X, 2:Y, 3:updatedX, 4:updatedY, 5:direction
%%%%%%%%%%%%%%%%%%%%%%get random
nodes%%%%%%%%%%%%%%%%%
for node_index = 1:NumOfNodes
TempX = randi([0,citysize],1,1);
if (rem(TempX,10)==0)
%sprintf('TempX = %d\n',TempX);
Node(node_index,1) = TempX;     %X co-ordinate in 1st column
Node(node_index,2) = randi([0,citysize],1,1); %Y co-ordinate in 2nd column
%sprintf('%d IF: X=%d Y=%d',node_index,
Node(node_index,1),Node(node_index,2))
else
Node(node_index,2) = 10*(randi([0,citysize/10],1,1)); %Y co-ordinate in 2nd column
Node(node_index,1) = randi([0,citysize],1,1); %X co-ordinate
```

```
%sprintf('%d ELSE: X= %d Y= %d',node_index,
Node(node_index,1),Node(node_index,2))
end
end
%% Assign Positions to RSUs
m=1;
temp=1472014;
for ii=blksiz/2:2*blksiz:citysize
n=1;
for jj=blksiz/2:2*blksiz:citysize
rsu.position{m,n}=[ii,jj]; % RSU's Position
rsu.ID{m,n} =  temp;% RSU's ID
plot(ii,jj,'xr','Linewidth',2)
text(ii+1,jj, num2str(rsu.ID{m,n}))
n=n+1;
temp=temp+1;
end
m=m+1;
end
m=round((citysize/(2*blksiz))+1);
for ii=blksiz/2+blksiz:2*blksiz:citysize
n=1;
for jj=blksiz/2+blksiz:2*blksiz:citysize
rsu.position{m,n}=[ii,jj];% RSU's Position
rsu.ID{m,n} =  temp;% RSU's ID
```

```
plot(ii,jj,'xr','Linewidth',2)

text(ii+1,jj, num2str(rsu.ID{m,n}))

n=n+1;

temp=temp+1;

end

m=m+1;

end

rsu.origID=rsu.ID;

% combine nodes position and RSU positions in a single matrix

temp=reshape(rsu.position,numel(rsu.position),1);

temp=temp(~cellfun(@isempty, temp)); % delete empty cell in the matrix

for ii=1:numel(temp)

temp1(ii,:)=temp{ii};

end

node_rsu=[Node;repmat(temp1,1,3)]; % combined matrix for rsu and nodes location

clear temp temp1


%%

%sprintf('Number of Nodes %d',NumOfNodes)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%

%labels = cell2str(num2str([1:NumOfNodes]'));

%h1 = ones(NumOfNodes,1);

h2 = ones(NumOfNodes,1);

h4 = ones(1,1);
```

```
counter=1;
for n = 0:citysize
for node_index = 1:NumOfNodes

if(rem(Node(node_index,1),10)~=0)
h2(node_index) = plot(Node(node_index,1)+n*(2*(rem(node_index,2))-1), Node(node_index,2),'.k');

node_rsu(node_index,3) = Node(node_index,1)+n*(2*(rem(node_index,2))-1);
node_rsu(node_index,4) = Node(node_index,2);
node_rsu(node_index,5) = rem(node_index,2)+2;
if node_index==src_node1
plot(node_rsu(node_index,3), node_rsu(node_index,4),'og');
h7=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(src_node1));
%           hold on
end
if node_index==dst_node
plot(node_rsu(node_index,3), node_rsu(node_index,4),'dm');
h9=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(dst_node));
%           hold on
end
else
h2(node_index) = plot(Node(node_index,1),Node(node_index,2)+n*(2*(rem(node_index,2))-1),'.k');
```

```
node_rsu(node_index,3) = Node(node_index,1);

node_rsu(node_index,4) = Node(node_index,2)+n*(2*(rem(node_index,2))-1);

node_rsu(node_index,5) = rem(node_index,2);

if node_index==src_node1

plot(node_rsu(node_index,3), node_rsu(node_index,4),'og');

h7=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(src_node1));

%               hold on

end

if node_index==dst_node

plot(node_rsu(node_index,3), node_rsu(node_index,4),'dm');

h9=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(dst_node));

%               hold on

end

end

end

%%%%%%%%AODV%%%%%%%%%%%%%5

% find all nodes which are in range of each other

for p = 1:size(node_rsu,1)

for q = 1:size(node_rsu,1)

dist=sqrt((node_rsu(p,3)-node_rsu(q,3))^2+(node_rsu(p,4)-node_rsu(q,4))^2);

if dist<=Range

inrange(p,q)=1;

else

inrange(p,q)=0;

end
```

```
end
end
src_node=src_node1; % to reset teh src_node to original source node after every iteration of n=1:citysize
rtngtble=src_node;% initialise
tble1=src_node;% initialise
tble=src_node;% initialise
cnt=1;% initialise
cnt1=1;% initialise
dimnsn(cnt)=numel(rtngtble);
while rtngtble~=dst_node

for ii=1:numel(tble1)
src_node=tble1(ii);
temp=find(inrange(src_node,:));
temp=temp(find(ismember(temp,tble)==0));
str{cnt1}=[src_node,temp];
tble=[tble, temp];
cnt1=cnt1+1;
end
tble1=tble(find(ismember(tble,rtngtble)==0));% seprate nodes which are not present in routing table
rtngtble=[rtngtble,tble];
% remove the repeated node in table
```

```
[any,index]=unique( rtngtble,'first');
rtngtble=rtngtble(sort(index));

if ismember(dst_node,rtngtble)
dst_cell=find(cellfun(@equal, str,repmat({dst_node},1,length(str)))); % find out
whihch structre cell has destination node
dst=dst_cell;
nodtble=dst_node;
frst_node=dst;
while frst_node~=src_node1
frst_node=str{dst(1)}(1);
dst=find(cellfun(@equal, str,repmat({frst_node},1,length(str))));
nodtble=[nodtble, frst_node];
end
%           msgbox('path found')
nodtble=fliplr(nodtble) % final routing table
%% uicontrol setting
set(H, 'String', cat(1, get(H, 'String'), {['Path ' num2str(nodtble)]}));
drawnow;
pause(0.25);

%           set(H, 'String', cat(1, get(H, 'String'), {'End'}));
%           drawnow;
%           pause(1.0);
%%
```

```
route{counter}=nodtble; % save all AODV paths for each change in vehicle position into a structure
h4= plot(node_rsu(nodtble,3),node_rsu(nodtble,4));
pause(0.01);
set(h4,'Visible','off');
[E,pcktlossrate,total_dist,pcktloss,thrgput]=evaluation(nodtble,node_rsu);                %
parameters calculation
energy(counter,:)=E; % energy consumption
distance(counter)=total_dist;% Total Distance between hops in AODV path
throughput(counter,:)=thrgput; % throughput
counter=counter+1;
end

cnt=cnt+1;
dimnsn(cnt)=numel(rtngtble);
if numel(rtngtble)==1
msgbox('1-No Node in range, Execute again')
return
end
if cnt>=5
%                 h8=msgbox('No path found');
break
end

end
```

```
pause(0.0001);
set(h2,'Visible','off');
set(h7,'Visible','off');
set(h9,'Visible','off');
end
%% plot results
figure(2)
plot(distance,'r','linewidth',2)
xlabel('Number of times path found during simulation of VANET')
ylabel('Dsiatnce in a path for each source and destination vehicles position')
title(['Total Distnace in each linked path with hops=', num2str(cellfun('ndims',route(1)))])
grid on
```

Figure (3)

```
plot(energy,'Linewidth',1.5)
xlabel('Number of times path found during simulation of VANET')
ylabel('Energy in Joules')
title('Energy Consumption')
legend('Data Rate=4 pckts/sec','Data Rate=6 pckts/sec','Data Rate=8 pckts/sec','Data Rate=10 pckts/sec','Data Rate=12 pckts/sec','Data Rate=14 pckts/sec')
grid on

nw_liftime=Eini./energy; % netwrok life time
```

Figure (4)

```
plot(nw_liftime,'Linewidth',1.5)
xlabel('Number of times path found during simulation of VANET')
ylabel('Netwrok Life time')
title('Netwrok life time plot for different data rates')
legend('Data Rate=4 pckts/sec','Data Rate=6 pckts/sec','Data Rate=8 pckts/sec','Data Rate=10 pckts/sec','Data Rate=12 pckts/sec','Data Rate=14 pckts/sec')
grid on
```

Figure (5)

```
plot(throughput/10e6,'Linewidth',1.5)
xlabel('Number of times path found during simulation of VANET')
ylabel('Throughput in MBps ')
title('Throughput plot for different data rates')
legend('Data Rate=4 pckts/sec','Data Rate=6 pckts/sec','Data Rate=8 pckts/sec','Data Rate=10 pckts/sec','Data Rate=12 pckts/sec','Data Rate=14 pckts/sec')
grid on
end
```

C. Model Evaluation

```
function [E,pcktlossrate,total_dist,pcktloss,thrgput]=evaluation(nodtble,node_rsu)
% take out the distance of nodes in routing table from each other
for ii=1:numel(nodtble)-1
distnc(ii)=sqrt((node_rsu(nodtble(ii+1),3)-node_rsu(nodtble(ii),3))^2+(node_rsu(nodtble(ii+1),4)-node_rsu(nodtble(ii),4))^2);
end
```

```
total_dist=sum(distnc); % total distnace from source to destination
time_consumed=total_dist/(3*10e9);
%% Perfromance Evolution
pktsize=64;% in bytes
datarate=[4,6,8,10,12,14]; % packets/sec
Etx=1;% in joules
Eini=Etx;
Elec=50e-9; %amount of Energy consumption per bit in the transmitter or receiver circuitry
Emp=0.0015e-12;%Amount of energy consumption for multipath fading
EDA=5e-9; %Data aggregation energy.
% paraemetrs for energy calculation using raio model of message
% transmission

alpha1=50e-9; %J/bit
alpha2=0.1e-9; %J/bit/m2
alpha=2;
Ebit=0.3e-3; % energy assigned to each bit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%radio Model for energy consumption is
% E=alpha1+alpha2*(dist)^alpha
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
hop=numel(nodtble);
for ff=1:length(datarate)
```

```
E(ff)=(alpha1*datarate(ff)*pktsize*8)+(alpha2*datarate(ff)*pktsize*8)*(total_dist)^al
pha;% energy loss calculation in transmitting packets at datarate
Edata(ff)=Ebit*datarate(ff)*pktsize*8;
for ll=1:datarate(ff)
Etx=Etx-(Elec*8*pktsize+Emp*8*pktsize);
Erx=Eini-Etx;
Erx=Erx-(Elec+EDA)*8*pktsize;
Eini=Etx;
if Etx<0.98
pcktloss(ll)=1;
else
pcktloss(ll)=0;
end


end
if hop>4 && datarate(ff)> 6
pcktlossrate(1,ff)=(datarate(ff)-7)/datarate(ff);
else
pcktlossrate(1,ff)=0;
end
thrgput(1,ff)= (datarate(ff)*pktsize)/time_consumed;
end



function log=equal(inpu,dst)
```

```
% dst=35;
if ismember(dst,inpu)
log=1;
else
log=0;
end
end
```