VANET BASED EFFICIENT PARKING SYSTEM

MOHD RASAUDDEN BIN ABD RAHMAN

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

STATUS CONFIRMATION FOR MASTER'S THESIS

VANET BASED EFFEICIENT PARKING SYSTEM

ACADEMIC SESSION : 2016/2017

I, **MOHD RASAUDDEN BIN ABD RAHMAN** agree to allow this Master's Project Report to be kept at the Library under the following terms:

1. This Master's Project Report is the property of the Universiti Tun Hussein Onn Malaysia.
2. The library has the right to make copies for educational purposes only.
3. The library is allowed to make copies of this report for educational exchange between higher educational institutions.
4. ** Please Mark (√)

[ ]	CONFIDENTIAL	(Contains information of high security or of great importance to Malaysia as STIPULATED under the OFFICIAL SECRET ACT 1972)
[ ]	RESTRICTED	(Contains restricted information as determined by the organization/ institution where research was conducted)
[√]	FREE ACCESS	

Approved by,

(WRITER'S SIGNATURE)

(SUPERVISOR'S SIGNATURE)

Permanent Address:

PT 18490, LORONG 2 KANAN,
JALAN GONG PASIR,
23000 DUNGUN,
TERENGGANU.

Supervisor's Name

ASSOC. PROF DR. JIWA BIN ABDULLAH

Date: 9/8/2016

Date: 9/8/2016

NOTE:

** If this Master's Thesis is classified as CONFIDENTIAL or RESTRICTED, please attach the letter from the relevant authority/organization stating reasons and duration for such classifications.

This thesis has been examined on date ....26/6/2016....

and is sufficient in fulfilling the scope and quality for the purpose of awarding the Degree of Master.

Examiners:

DR. LUKMAN HANIF BIN MUHAMMAD AUDAH
Faculty of Electrical and Electronic Engineering
Tun Hussein Onn University of Malaysia

DR. KHAIRUN NIDZAM BIN RAMLI
Faculty of Electrical and Electronic Engineering
Tun Hussein Onn University of Malaysia

VANET BASED EFFICIENT PARKING SYSTEM

MOHD RASAUDDEN BIN ABD RAHMAN

A project report submitted in partial

fulfilment of the requirement for the award of the

Degree of Master of Electrical Engineering

Faculty of Electrical & Electronic Engineering

Universiti Tun Hussein Onn Malaysia

JULY 2016

DECLARATION

I hereby declare that the work in this project report is my own except for quotations and summaries which have been duly acknowledged

Student : ....

MOHD RASAUDDEN BIN ABD RAHMAN

Date : 9/8/2016

Supervisor : ....

PROFESOR MADYA DR. JIWA BIN ABDULLAH

ACKNOWLEDGEMENT

I would like to give thanks to Allah S.W.T for giving me the opportunity to implement and execute this project. I would like to express my deepest gratitude to my supervisor, Associate Professor Dr. Jiwa bin Abdullah for providing me with an excellent guide for doing this study. I also would like to thank all of my lecturers and course mates for their inspiration and motivation. Last but not least, I would like to thank my family members especially to my mother and siblings for their support and also to my beloved wife for their care and encouragement for me to complete this project.

ABSTRACT

A recent study discovers that about 30% of the traffic congestion is caused by vehicles cruising around their destination and looking for a vacant park. Therefore, addressing the problems associated with parking in crowded area is significance. One of the solutions to improve the existing guidance parking system with low cost consumption is use wireless ad-hoc network system with infrastructure less. This study presents the outcomes of the performance evaluation of efficient parking system based on Vehicle Ad-hoc Network (VANET). This simulation uses software OMNeT++ and Dynamic MANET On-demand (DYMO) as routing information dissemination protocol. The performance of the network is evaluated on parameters of throughput and packet delivery ratio (PDR) respect to different node density and data packet sizes. This evaluation can determine either this VANET based parking system with DYMO routing protocol is most suitable and efficient to implement in the real time. In this study, observed that the percentage of packet delivery ratio of DYMO is more than 83% even in high density nodes. Therefore DYMO is most suitable routing protocol to use in order implement for this parking system which has large parking lots.

ABSTRAK

Kajian terkini menemui 30% kesesakan lalulintas disebabkan oleh pergerakkan kenderaan untuk mencari parkir di sekitar tempat yang dituju Oleh itu mengutarakan masalah yang berkaitan dengan parkir di kawasan yang sesak adalah sangat relevan. Salah satu penyelesaian untuk meningkatkan sistem parkir berpandu sedia ada dengan kos yang rendah adalah menggunakan sistem rangkaian tanpa wayar sementara tanpa infrastruktur. Kajian ini membentangkan hasil simulasi penilaian prestasi sistem parkir kenderaan berefisien berasaskan *Vehicular Ad-hoc Network (VANET)*. Simulasi ini menggunakan perisian OMNeT++ dan *Dynamic MANET On-Demand (DYMO)* sebagai protokol laluan penghantaran maklumat. Prestasi rangkaian dinilai berdasarkan parameter jumlah penerimaan data per masa dan nisbah penghantaran paket berdasarkan bilangan nod dan saiz paket data yang berbeza. Penilaian ini boleh menentukan sama ada sistem parkir efisien berasaskan VANET ini yang menggunakan protokol laluan DYMO sangat bersesuaian dan efisien untuk dilaksanakan dalam keadaan sebenar. Dalam kajian ini, didapati peratusan nisbah penghantaran paket data bagi DYMO adalah lebih daripada 83% walaupun dalam bilangan nod yang tinggi. Oleh itu, DYMO sangat bersesuaian digunakan sebagai protokol laluan penghantaran untuk sistem parkir ini yang mempunyai bilangan lot parkir yang besar.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

RREQ	-	Route Request
RREP	-	Route Reply
RERR	-	Route Error
MANET	-	Mobile Ad-hoc Network
VANET	-	Vehicular Ad-hoc Network
DYMO	-	Dynamic MANET On-demand
WAVE	-	Wireless Access for the Vehicular Environment
V2V	-	Vehicle–to-Vehicle
V2I	-	Vehicle–to-Infrastructure
OMNeT++	-	Objective Modular Network Testbed in C++
IETF	-	Internet Engineering Task Force

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1. Research Background

Today, parking system plays an important role in the urban transportation scenario. Various schemes have been introduced for the convenience and comfort of city dwellers thus reducing traffic congestion, air pollution and as well as driver frustration. It has been pointed out that about 30% [1] of the traffic congestion is caused by vehicles cruising around their destination looking for a parking space. An effective solution to this problem is by providing guidance for the vehicles to be parked according to the occupancy status of each parking lot. The growing numbers of vehicles are typically equipped with wireless transceiver which means that they are able to communicate with other vehicles thus; forming a special class of wireless network. This is one reason that Vehicle Ad-hoc Network (VANET) system is widely used in the guidance parking system. A VANET system turns every participating car into a wireless router or nodes to connect each other within the transmission range of 100 to 300 meters to create a wireless ad hoc network with a wide range. All vehicles equipped with a wireless transceiver should be able to receive and relay messages to other wireless transceiver in its neighbourhood.

The purpose of this study is to investigate the performance of guidance parking system based on VANET which uses Dynamic MANET On-Demand (DYMO) as a routing protocol. The performance of information dissemination between nodes in this parking topology will be analysed by using discrete simulation software, OMNeT++ to conclude that the method of this parking design is suitable or not if it were to be implemented in real time environment.

1.2 Wireless Ad-hoc Network

Wireless Ad-hoc network is defined as a network which doesn't have pre-existing communication infrastructures which is network created by some nodes that are available. In ad-hoc network, all devices are treated equally with have same status. There are some differences between MANETs and VANETs and will be briefly explain in this chapter. The main differences between MANET and VANET are the special mobility pattern and rapidly changing topology in which the existing MANET routing protocols is not effectively applied into VANETs [2].

1.2.1 Mobile Ad-hoc Network (MANET)

MANET is collection of two or more autonomous nodes which communicate with each other without any centralized administering node or might be call infrastructure less mobile network. In MANET, different participating node moves randomly in order to create ad-hoc wireless network. The routing protocols of MANET are not feasible to be used in the VANET network because of their route instability. Some of the main application of MANET as shown as follows.

i. Personal Area Networking (PAN): A personal area network is a localized a short range network associated with person. This network including mobile devices such as PC, smartphone, printer, switches connected in short range that associated with person.
ii. Crisis Management Application: In case the communication infrastructure is damaged, wireless ad-hoc network created easily in limited time.
iii. Vehicular Ad-hoc Network (VANET): To improve safety in the road, connection between moving vehicle on road is created. [2]

1.2.2 Vehicular Ad-hoc Network (VANET)

A Vehicular Ad-hoc Network (VANET) is a subclass of mobile ad-hoc network (MANETs) with the distinguishing property that the node is the vehicle. In VANET, the nodes are moving on predefined roads and their trails aren't too complicated and this is where the routing protocols have to be modified or changed. VANET addresses all the issues that are related to the communications between vehicles and on-going research with wireless communication and also covers the aspects of Wireless Access for the Vehicular Environment (WAVE). VANETs basically enable infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications as shown in Figure 1.1 [2, 4]

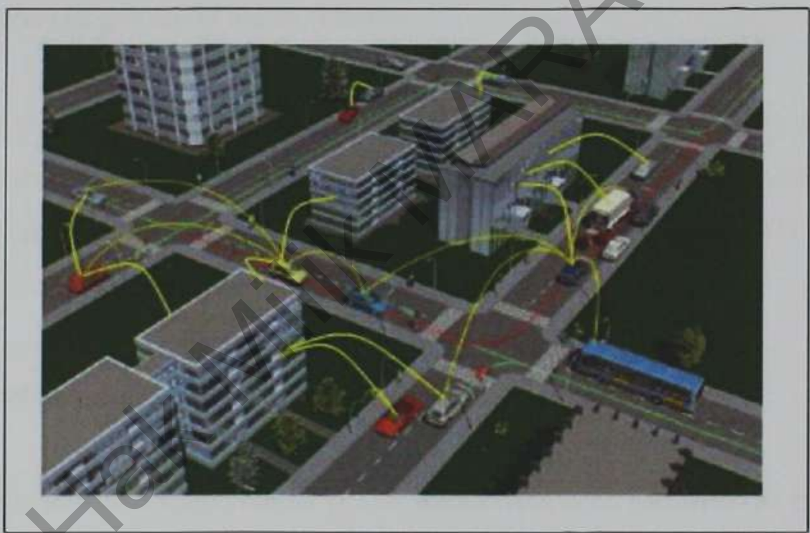

Figure 1.1: VANET topology route [4]

1.3 Problem Statements

Finding an available parking space in a large parking area is time consuming which would incur great economic losses. The guided parking systems in Malaysia were only provided at big shopping complexes that surely consume high capital to implement it. One of the solutions to improve the existing guided parking system with low cost consumption is to use wireless ad-hoc network system. VANET based guided parking system connects two or more vehicles in certain area to disseminate information through connectivity that is established by the related routing protocol.

Each vehicle that participate in network exchange an information such as parking lot number to another vehicles that are searching for vacant space at that time. High mobility of nodes in VANET makes design a challenging issue.

Based on existing guided parking based on VANET, this thesis will focus on the comparison of data respect to different node density and packet size. In this study, Dynamic MANET On-demand (DYMO) used as routing protocol. In a comparative study, the data retrieved and compares the percentage of packet delivery.

1.4 Research Objectives

The objectives of this study are as follows;

1. To investigate a parking management design approaches and their limitation.
2. To simulate the guidance parking system via VANET by using OMNET++.
3. To analyse the performance of wireless network using DYMO as routing protocols for guidance parking system.

1.5 Scopes

In order to realize this study, the scopes of works have been identified as follows:

1. Literature review that covers all researches about current parking management strategy and VANETs routing protocols.
2. Select a real parking topology environment which MyDin Mall Kuala Terengganu in which the area is about 275m x 200m.
3. Study OMNET++ methodology that simulates a different number of nodes to navigate parking spaces.
4. Analysis a total throughput and Packet Data Ratio (PDR) with respect to packet size and different node density (maximum 150 nodes).

1.6 Organisation of the Dissertation

Chapter 1 summarizes the project background and elaborates on the project by briefly description of differences two networks, VANETs and MANETs. In this chapter also describes the problem statement, objectives and scope of the project. The dissertation is introduced in this chapter.

Chapter 2 reviews literatures, including description of existing design of guidance parking system, routing protocols in VANET, DYMO performance and warm introduction of OMNeT++.

Chapter 3 describes the methodology of the dissertation. OMNET++ simulation procedure is briefly described. Modelling of efficient parking system with V2V communication method is briefly explained and also methodology flow chart is presented in this chapter as well. Gantt chart is shown for the actual progress of the state.

Chapter 4 elaborates on results and analysis in OMNET++ simulation of DYMO routing protocols performance. It also describes the findings of the simulation results and analysis process for each of the resulting data.

Finally, Chapter 5 discusses the conclusions of the dissertation, and recommends possible future research.

CHAPTER 2

VANET BASED EFFICENT GUIDED PARKING SYSTEM DESIGN METHODOLOGY REVIEW

2.1 Introduction

In this chapter, a comprehensive review of the existing application and requirement of guided parking system design based on VANET and wireless network technology is discussed. In order to evaluate the performance of each design, the author in [1, 4, 5, 6, 7] introduces a different simulation method either in real time simulation or using by software simulator. Moreover, this chapter provides a review of the guidance to use OMNeT++ simulation and routing protocol in order to disseminate information in VANET.

2.2 The Existing Smart Parking Management Review

Many parking guided system have been developed over the past decade. In this part, several existing parking guided approaches and their limitation briefly explained. Some of these difference parking management strategy and approaches has simulated under realistic traffic and parking condition. [4, 6] provides the guided parking system using hardwired and [1, 5, 7] provides the guided parking system using wireless ad-hoc network (VANET). These parking schemes reviewed and compared with this study.

2.2.1 A Reservation-based Smart Parking System

Wang et al [4] have designed and implemented the prototype of a reservation-based smart parking system (RVSP) that provides a real time parking information broadcasting and reservation service for targeted user by using advanced sensing and mobile communication methods which retrieve by using internet and Wi-Fi. The reservation of parking make from internet and the sensors verify the driver's identity via Bluetooth module. The system hardware for this prototype system which is organized into three main components, the sensor network, the central server and the mobile device connect each other to implement the system.

In order to evaluate the system performance, this parameter introduced as followings; *walking distance* and *traffic volume*. *Walking distance* is defined as the distance from a driver's selected parking space to the destination whereas *traffic volume* is defined as the amount of traffic generated by parking searching. In order to investigate the parking system, simulator software *iRev* is used and the architecture of software as shown in Figure 2.1.

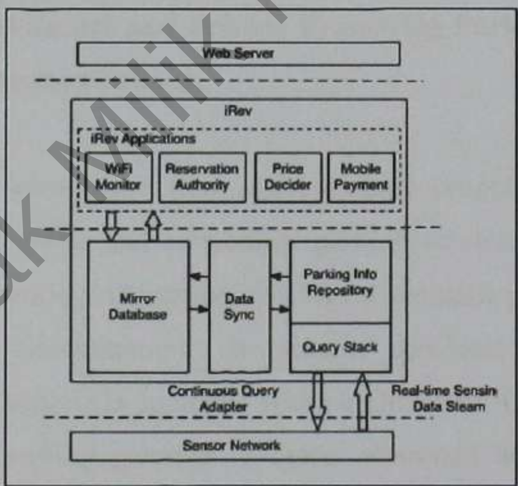

Figure 2.1: iRev simulator architecture [4]

The biggest weakness in this system is a large installation and operational costs in order adequately monitor the parking spaces. More parking lots will increase cost and surely the maintenance cost of each sensor also high. Furthermore each parking must access with internet to communicate with the management system and users and share parking information with other parking lots.

2.2.2 VANET Based Airport Parking System Simulation Using MATLAB

Suresh [5] evaluated a VANET based for Airport Parking System by using MATLAB simulation considered vehicle-to-infrastructure (V2I) unit as an interconnection in the VANET system, which is the unit considered all routes and the wireless communication as a nodes and Road Side Unit (RSU) act as central controller for the region. In order to investigate the performance of this parking system, MATLAB is used to develop algorithm and create models and application.

This study prefers to different environment and use the same networks system, VANET. Vehicles are considered as node that in stationary and mobility movement and provided with built-in transceiver. The communication unit vehicle-to-infrastructure (V2V) is modelled with different simulator, OMNeT++. Compare that MATLAB, most researcher convenient to use OMNeT++ because of friendly use and easy to understand.

2.2.3 An Intelligent Secure and Privacy Preserving Parking Scheme Through Vehicular Communications.

Lu et al [6] introduce a new smart parking proposed based on VANET provides user with accurate and convenient parking services in large parking lots including real time parking navigation, intelligent antitheft protection and friendly parking information dissemination. The scheme considers the flourish stage of VANET where each vehicle is equipped with an On Board Unit (OBU), Road Side Unit (RSU) act as parking guarded to detect abnormal activity at parking lots (antitheft protection services) and trusted authority (TA) which responsible for the registration of both OBUs and the parking lot RSUs. The model of parking lot that suggested as shown in Figure 2.2.

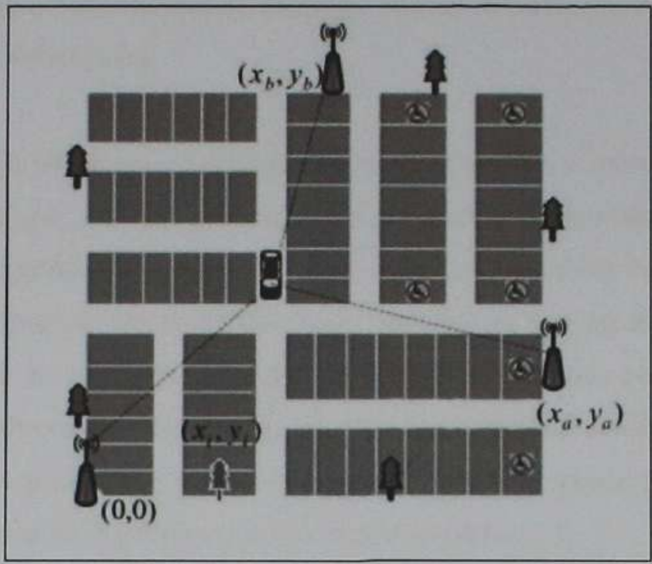

Figure 2.2: Parking lot model [6]

In order to evaluate the system performance, these parameters is introduced as following; *position, reservation, occupancy, pseudo-ID, ticket ID, ticket key, start time* and *last update time.* In order to evaluate the performance parking system, two custom simulators built in Java as followings; *Simulation on Real-Time Parking Navigation* and *Simulation on Parking Information Dissemination*.

This scheme is proved to be efficient and effective but deploying RSUs in large scale requires a large amount of investment. In this study, we also consider the communication method that approaches such as *vehicle to infrastructure* (V2I) which is employment of RSUs to disseminate message in network range. But in order to evaluate the performance of system, the simulation environment that proposed not familiar with some researcher. Otherwise the routing protocol not discussed.

2.2.4 Reaching Available Public Parking Spaces in Urban Environment using Ad-hoc Networking

Verroios et al [7] introduce a guidance parking system that a vehicle can determine the best route to get a vacant parking lot that reported to be available. In order to implement this system, the author use *Time-Varying Travelling Salesman Problem* approaches to compute the possibility route for vehicle to visit all vacant parking spaces known to be available. In order to evaluate the system performance, these criteria is introduced as followings; a) The time needed to arrive each space available, b) the probability to find the spot in question available at the time of arrival, and c) the walking distance to the final destination. [7]

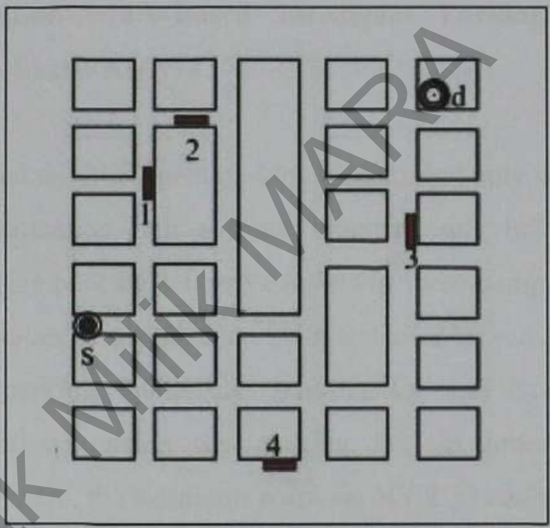

Figure 2.3: Example of parking space [7]

Figure 2.3 shown an example parking that proposed which is *d* represent of the final destination for the driver, *s* represent of the present position of vehicle and the number 1,2,3,4 are represent the vacant space parking are currently available. Communication system such as *vehicle-to-infrastructure (V2I)* and *vehicle to vehicle (V2V)* are considered as an interconnection in the VANET system to help vehicles navigate inside large parking lots.

2.2.5 Smart Parking Service based on Wireless Sensor Networks

Yang et al [8] introduces the smart parking based on Wireless Sensor Network (WSN) that allows vehicle drivers to find the free parking space efficiently. The proposed scheme consists of low cost wireless sensor networks, embedded web server (EWS), central web server (CWS) and mobile phone application (MPA). This scheme is same with [4] which is the biggest problem is the increase of parking lot to cover will increase cost to invest even though this scheme use low cost wireless sensor network. The future design should use low cost design in order to use current technology network such as VANETs.

2.2.6 IPARK: Location-Aware-Based Intelligent Parking Guidance over Infrastructure-Less VANETs

Zhao et al [1] introduced the intelligent parking system that only deployed *Vehicle to Vehicle (V2V)* communication unit without requiring any infrastructure cost as shown in Figure 2.4. This parking scheme consists of three components; parking lot cluster, intermediate nodes and end user in order to achieved the following two goals; a) Real-time parking inside the parking lot and b) Efficient parking information dissemination outside the parking lot. In order to evaluate the performance parking system, the simulator software NS-2.33 environment is used.

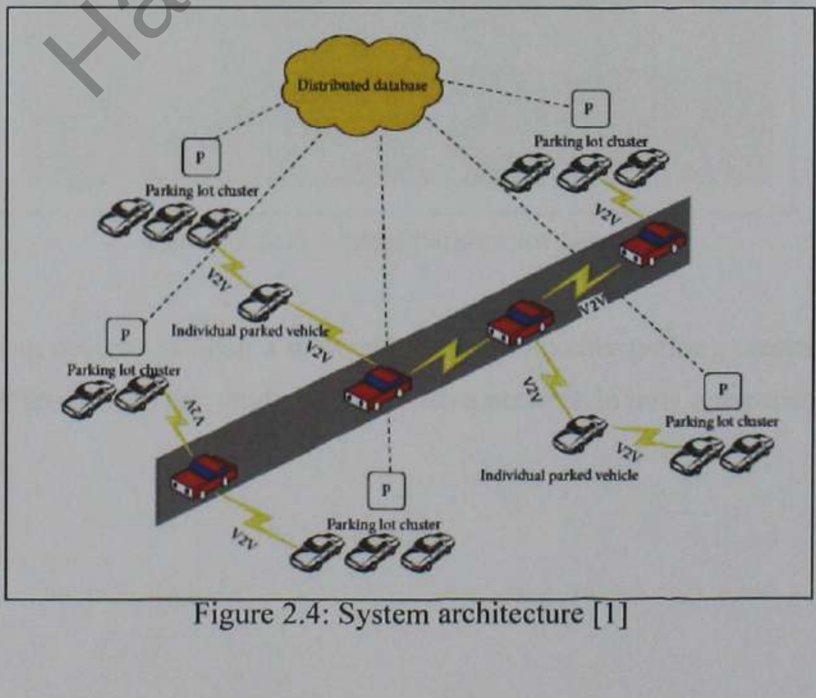

Figure 2.4: System architecture [1]

This scheme of parking system is almost similarly with this study which there is no need requirements of infrastructure investment. Moreover the routing protocol for dissemination information of parking lots between vehicles is not discussed. Otherwise this study uses a different method of environment simulation to evaluate performance of design.

A typical parking lot cluster as shown in Figure 2.5 is similar design with this study which is the first parked vehicle is declared as cluster head (H) and the others parked vehicle is declared as members ($M_1 - M_{23}$). For this study, a cluster head (H) changed to destination host, host[0] and another member's (M) changed to host[1..249] for 250 nodes case and quasi-head (QH) as backup cluster head is not introduced.

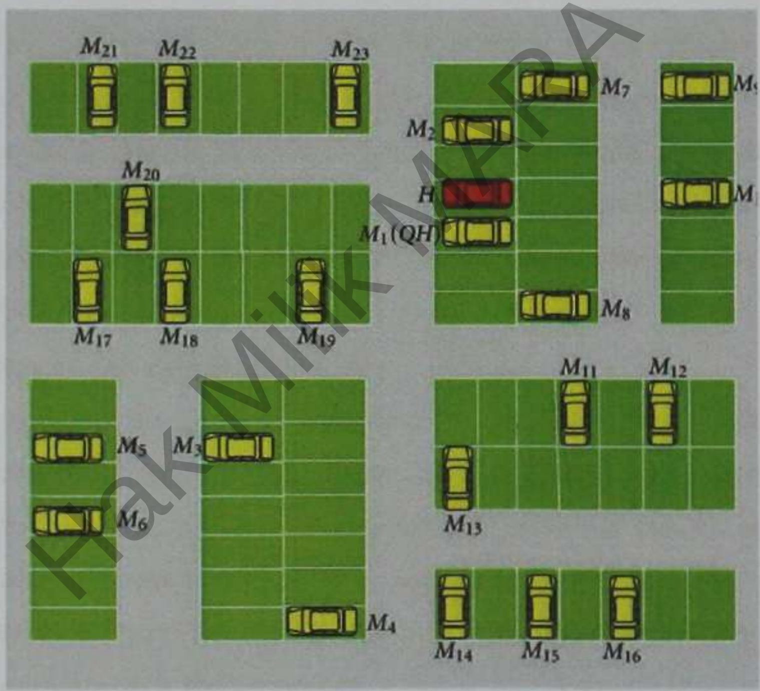

Figure 2.5: A typical parking lot cluster [1]

This parking design establish a networks between another parking cluster in certain area but different with this study just establish a network in only one parking lot area.

2.3 VANETs Overview

Vehicular ad-hoc networks (VANETs) enable vehicles to communicate with each other (V2V) as well as with roadside infrastructure units (V2V). These units provide different services such as driver information system and internet access.

2.3.1 Characteristic of VANETs

There are some special characteristics of VANET as followings;

I. **High dynamic topology.**

VANET have very high dynamic topology. The communication links between node changes very rapidly. Communication between two nodes remains remain very short period. For example if two vehicles moving away from each other with speed of $25ms^{-1}$ with transmission range 250m, the connection present is about 5 second ($250m/ 50ms^{-1}$). For this study, the speed of vehicles is set at $15\text{-}22ms^{-1}$ and communication range at 50m means that the period of communication links between two vehicles is about $1.1\text{-}1.7ms^{-1}$.

II. **Frequent disconnected network.**

To maintain the continuous connectivity, vehicles need another connection nearby immediately. But if failure occurs, vehicles can connect with Road Side Unit (RSU). Frequent disconnected network mainly occurs for low density of vehicles commonly occurs at rural area. For this study, a destination host set as RSU to maintain continuous connectivity each vehicle.

III. **Communication Environment**

The mobility model is commonly different depends on environment which is mobility of vehicles at highways is different with city environment. Mobility modelling of vehicle movement prediction and routing algorithm should adapt to these change. For highways mobility models are very simple because vehicle movement is one dimensional compare than city mobility models. For this study, stationary mobility, linear mobility and random mobility are considered.

IV. Hard delay constrains.

Safety aspect like accident, sudden break and emergency call of VANET application depends upon the deliver time of data. It cannot compromise for data delay in this aspect. Therefore hard delay constrain is more important in VANET than high data rate.

V. Interaction with on-board sensor

The on-board sensor are built-in the vehicle. This sensor is used to find vehicle location, vehicle speed and vehicle movement. This information is used for effective communication between vehicles.

2.3.2 Routing Protocols

A routing protocol is mechanisms that manage information exchange between two communication entities. This management include of establishing a route, decision in forwarding data, route maintenance and failed route recovery. There have some problem that occurs in designing the routing protocols in VANETs. Lin et al [2] considers two problems that occur, the temporary network fragmentation problem and the broadcast storm problem. Rapidly changeable topology influence on the performance of data transmission caused the temporary network fragmentation problem and the broadcast storm problem seriously affects the successful rate message delivery in VANETs [2]. To enhance the safety of drivers and provide the comfortable driving environment, messages for different purposes need to be sent to vehicles through the inter-vehicle communication. Figure 2.6 shows the structure of VANETs routing that surveyed in [2]. As describe previous before, VANETs is subset of MANETs that's mean there are some of routing protocols developed for MANET but these usually do not apply for the VANET because of high mobility nodes and frequently disconnection on network.

From Figure 2.6, VANETs is structured into three broad categories, unicast routing, multicast and geo-cast routing and broadcast routing. Unicast routing is a fundamental operation for vehicle to construct a source-to-destination routing in a VANET as in Figure 2.7(a). Multicast protocol is defined by delivering multicast packets from a single source vehicle to all multicast members by multi-hop communication. Geo-cast routing is to deliver a geo-cast packet to a specific geographic region which means vehicles that located in this specific geographic region should receive and forward the geo-cast packet as shown in Figure 2.7(b). Broadcast protocol is utilized for a source vehicle sends broadcast message to all other vehicles in the network as shown in Figure 2.7(c) [3].

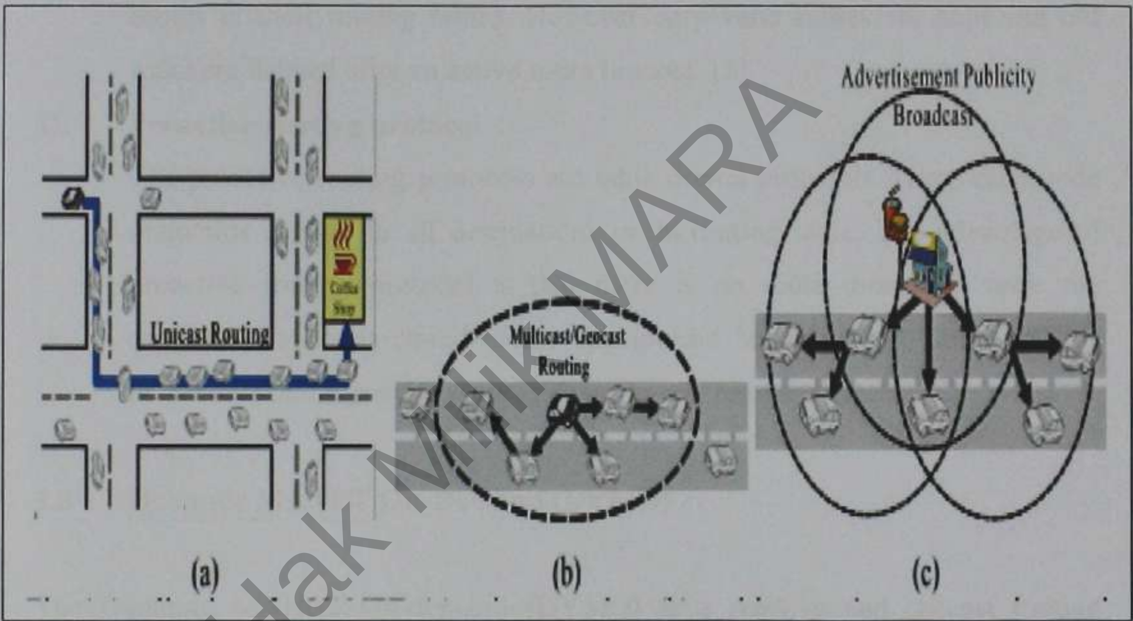

Figure 2.7: Routing protocols of VANETs (a) Unicast routing, (b) multicast and geo-cast routing, (c) broadcast routing [3]

Kumar et al [3] classified into two major categorize of routing protocols in VANETs; topology-based and position-based routing but Wang et al [4] classified the routing protocols to five categorized; topology-based, position-based, cluster based, geo cast and broadcast routing protocols. Topology-based Routing Protocols that related with this study is briefly description as followings.

2.2.1 Topology-based Routing Protocols

These routing protocols use links information that exist in the network to disseminate packet to participating nodes in network. There are divided into two which are Reactive and Proactive routing protocols. The description of both as followings;

I. **Reactive routing protocol**

This protocol discovers a route for communication between two nodes when it need. A source node initiates route discovery by broadcasting route query or request messages into the network. All nodes maintain the discovered routes in their routing tables. However only valid routes are kept and old route are deleted after an active route timeout. [2]

II. **Proactive routing protocol**

The proactive routing protocols are table driven protocols where each node maintains a route to all destinations in its routing table. The advantage of proactive routing protocol is that there is no route discovery since the destination route is stored in the background but the disadvantage of this protocol is that it provides low latency for real time application. [2]

2.3 Dynamic MANET On-Demand (DYMO)

The Dynamic MANET On-demand (DYMO) is a reactive and unicast routing protocol which has been especially designed for MANET. This protocol is a simple and fast routing protocol for multi-hop networks. This protocol has been proposed by Perkins as advancement to the existing od AODV protocol. The basic operations of DYMO are route discovery and route maintenance that elaborated as followings [13]:

I. **Route Discovery**

Route discovery is the process of creating a route to a destination when a node needs a route to it. When a source node wishes to communicate with a destination node, it initiates a Route Request (RREQ) message. In the RREQ message, the source node includes its own address and its sequence number, which is incremented before it is added to the RREQ. The most important part is the address of the destination node. Upon sending the RREQ, the originating node will await the reception of an RREP message from the destination node or target node. If no RREP is received within RREQ waiting time, the node may again try to discover a route by issuing another RREQ. When the RREQ reaches the destination node, an RREP message is created as a response to the RREQ that containing information about destination node [13]. The description above as illustrate in Figure 2.8.

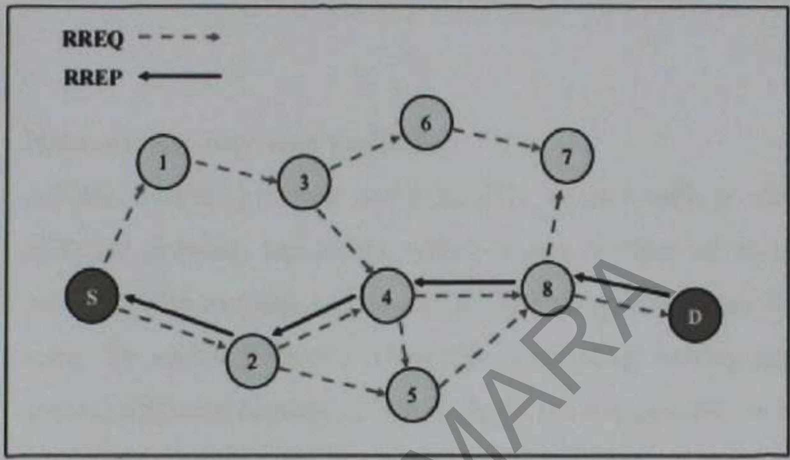

Figure 2.8: DYMO Route discovery process [13]

II. **Route Maintenance**

Route maintenance is the process of responding to changes in topology that happens after a route has initially been created. To maintain paths, nodes continuously monitor the active links and update the Valid Timeout field of entries in its routing table when receiving and sending data packets. If a node receives a data packet for a destination it does not have a valid route for, it must respond with a Route Error (RERR) message. When creating the RERR message, the node makes a list containing the address and sequence number of the unreachable node [13]. A Route Error (RERR) message is generated when a node receives a data packet for the destination for which route is not known or the route is broken. This RERR notifies other nodes about the link failure. The source node reinitiate route discovery quickly as it receives this RERR.

A serious issue for MANETs arises when link failures occurs due to high node mobility. Therefore reactive routing protocols are subjected to an increase in network control overhead.

2.3.1 Routing Protocol Issue

There are some issue have been discuss in previous research briefly describe as followings;

I. **Network topology and mobility.**

Ad-hoc routing protocol must be able to suit with a wide variety of different network topologies with varying number of nodes. The most problem with random topologies is they likely not cover the worst case setup for ad-hoc network. Therefore, evaluating routing protocols using several different topologies will help to outline possible weakness in the ad-hoc routing process. There are there kind of topologies that commonly used such as random topology, grid topology, and linear network. In addition to the different topologies, the density of the network can have a great influence on the routing performance. Consideration of nodes participating and communication range is relevant for this issue. For mobility issue, node mobility also influences the outcome of simulation to a large degree [12]. For this study, the performance of DYMO is evaluating even it is suitable to use in order to implement this parking system in real time. This simulation also consider of mobility and stationary nodes in traffic generation. Mobility nodes is parked vehicle and mobility nodes is vehicle user seeking a vacant parking represent in linear network topology.

II. **Parameter setting and traffic pattern.**

For any protocol evaluation, it is necessary to precisely identify and document the possible ranges of the protocol parameters. These parameters must either be clearly defined by the application scenario or they need to be analysed in different simulation runs. Figure 2.9 shows the protocol parameters that are suggested by the IETF [12].

Parameter	Value
MIN_HOPLIMIT	5 hops
MAX_HOPLIMIT	10 hops
NET_TRAVERSAL_TIME	1000 ms
ROUTE_TIMEOUT	5 s
ROUTE_AGE_MIN_TIMEOUT	1 s
ROUTE_AGE_MAX_TIMEOUT	60 s
ROUTE_NEW_TIMEOUT	5 s
ROUTE_USED_TIMEOUT	5 s
ROUTE_DELETE_TIMEOUT	10 s
RREQ_RATE_LIMIT	$10 s^{-1}$
RREQ_BURST_LIMIT	3 RREQs
RREQ_WAIT_TIME	2 s
RREQ_TRIES	3

Figure 2.9: DYMO module parameter [12]

III. Performance metrics.

A number of performance metrics are relevant for the adequate analysis of network protocols such as the number of packets sent and received or message latencies. Some of them are especially relevant in the domain of wireless ad-hoc routing that influenced by the configured setups (different topologies, network densities, mobility configuration, and traffic pattern) [12]. For this study, performance metric that use in order to evaluate DYMO are packet delivery ratio and throughput respect to different packet sizes and node density.

IV. Simulation control

Two aspects of simulation control need to be considered in the context of automated simulation execution are how to run simulations and when to terminate each run. The simplest approach to running simulations would be to execute all simulations sequentially on one computer. However, simulations may take a long time to complete [12]. For this study, execution of simulation thru OMNeT++ divided to two approaches; first approach is executing all simulation sequentially and the second approach is executing with random generator setting with different seed in command line.

2.3.2 Advantages of DYMO

For this study, selection of routing protocols is critical decision. From previous research about routing protocol of DYMO, there are some advantages of DYMO that influence this study to use this protocol as per followings;

- Rahman et al [13] conclude that the performance of DYMO is best performance than AODV, OLSR, and ZRP in term of packet delivery fraction.
- DYMO is energy efficient when the network is large and shows a high mobility [11].
- The routing table of DYMO is comparatively less memory consuming than AODV even with Path Accumulation feature [11].
- The overhead for the protocol decrease with increased network sizes and high mobility [11].
- Nand et al [15] found that DYMO performs better than both DSR and AODV in terms of ability to search route quickly as it avoids expiring good route by updating route lifetime appropriately.
- Gupta et al [11] observed that DYMO has a high throughput and packet delivery, low average end to end delay but incurs a low routing overhead.

2.4 Traffic Simulation

In the following, the OMNeT++ simulation environment briefly introduced in order to evaluate the performance of DYMO for this study.

2.4.1 OMNeT++ (Objective Modular Network Testbed in C++)

OMNeT++ is a discrete event simulator used for modelling communication networks, multiprocessor and other distributed or parallel system based in C++. This software consists of different modules which communicate with each other using message passing that are written in C++ programming. Simulation models in OMNeT++ are described in NED language. Development of OMNeT++ is started in 1992, since then many people contributed to OMNeT++ with several models. It is primarily used to simulate the communication networks and other distributed systems. It used for academic as well as industrial research purposes. Here are the features of OMNeT++ which makes it different from other simulation environment [15].

i. OMNeT++ is designed to support network simulation on a large scale
ii. Modular structure.
iii. The design of NED (Network Description).
iv. GUI Interface with Graphical Editor.
v. Separation of Model and Experiments.
vi. Simple Module Programming Model.
vii. Design of the Simulation Library.
viii. Parallel Simulation Support.
ix. Real-Time Simulation, Network Emulation.
x. Animation and Tracing Facility.
xi. Visualization of Dynamic Behaviour
xii. Enriched Result Analysis Mechanism.

Simulation of VANET is divided into two parts as followings;

a. Traffic Simulation: Generation of traffic movement, defining the mobility model vehicle and creating traffic movement.
b. Network simulation: Generating inter communicating vehicle, defining communication protocols.

Both the simulation is connected in bi-directional coupling.

CHAPTER 3

SIMULATION OF VANET EFFICIENT PARKING SYSTEM USING DYNAMIC MANET ON-DEMAND (DYMO) AS INFORMATION DISSEMINATION ROUTING PROTOCOL

3.1 Introduction

This chapter includes a review of the research method and the appropriate design used for this study. As stated earlier, one of the objectives of this research is to build the simulation using OMNeT++ toward produce the VANET based Guidance Parking System analysis for real time implementation. The data gathered from the simulation will be interpreted and discussed. The challenge on this study is to create the appropriate nodes mobility in parking topology and to ensure the correct procedures during simulation work in order to get a precise result. This chapter describes the software used in this project, and presents the project's methodology flow chart.

3.2 The System Design

In order to design the efficient parking system based on VANET, a real parking topology of shopping mall at Kuala Terengganu have been selected (MyDin Mall). This parking topology has approximately 329 parking spaces in dimension area of 275m x 200m as shown in Figure 3.1.

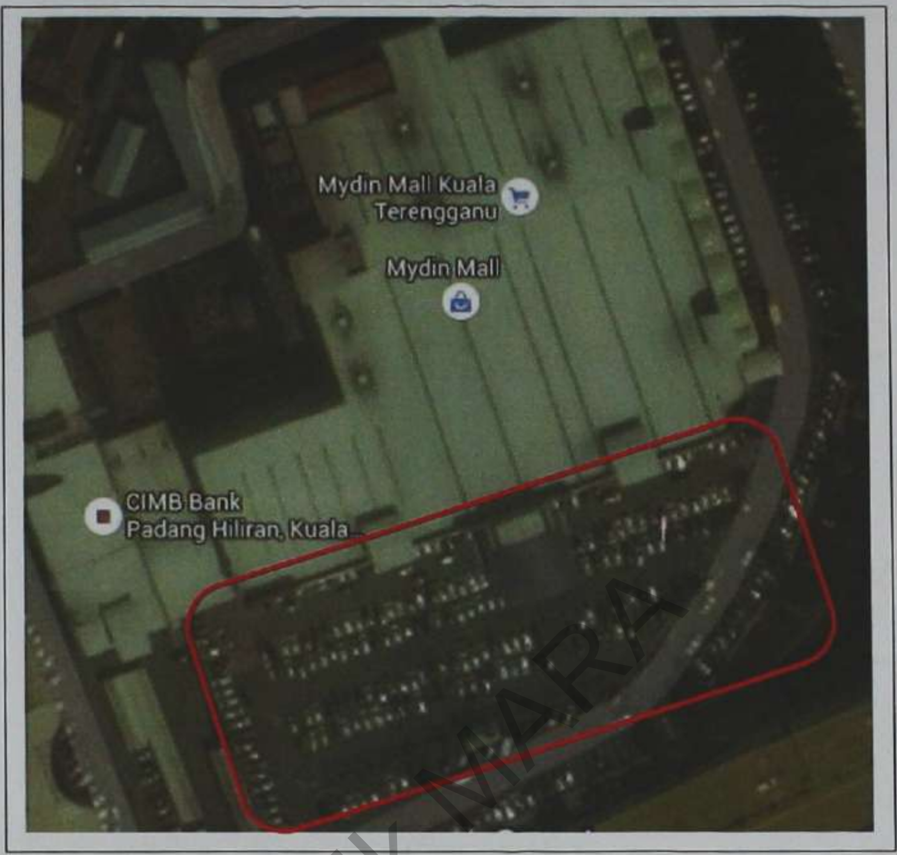

Figure 3.1: Parking lot map used in simulation

For this study, some criteria are considered in order to implement this guidance system in real time. There are as followings;

1. **Assumptions and Design Goals:**
   - Assumes that each vehicle is equipped with transceiver and GPS that common use in the future.[8]
   - Assume that some vehicles user will share their devices during parking. The user sends a parking lot number to destination host in order to update parking vacancy.[8]

2. **Real-Time Parking Navigation:**

   While a vehicle enters a large parking lot system, the destination host should send the current parking vacancy map (a map showing each parking spot as occupied) to help them find an available parking space. The examples of vacancy map that use in this study as shown in Figure 3.2. [8]

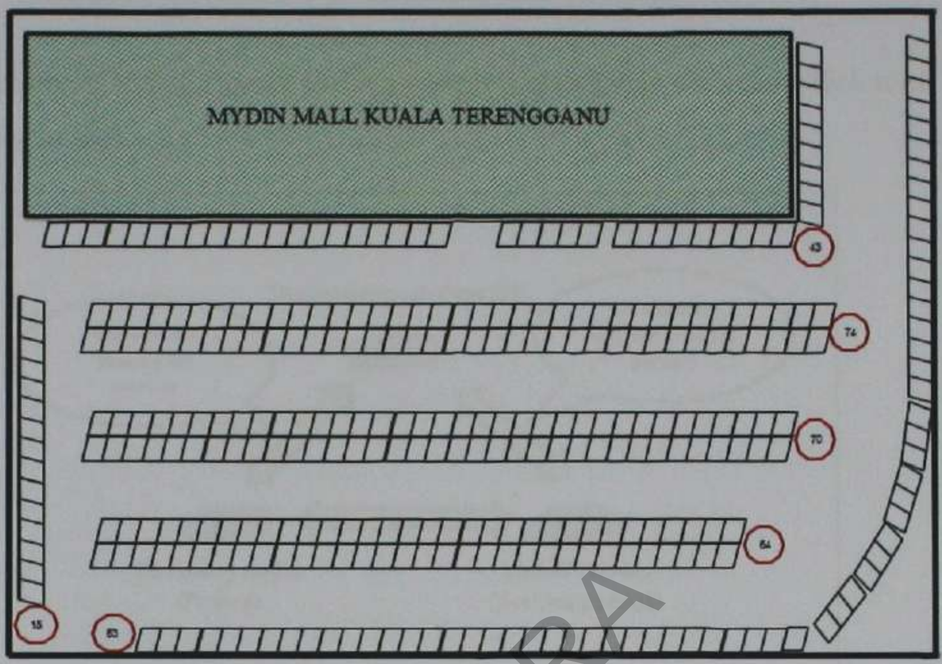

Figure 3.2: Parking vacancy map

3. **Efficient Parking Information Dissemination:**

   While a vehicle user is driving on a road and issue a parking data query request, a destination host should respond to this request and return the query results (the number of parking spaces available on each parking lot) at small delay. [8]

3.3 Architecture

The architecture of VANET based parking system is elaborately designed which is as shown in Figure 3.3.

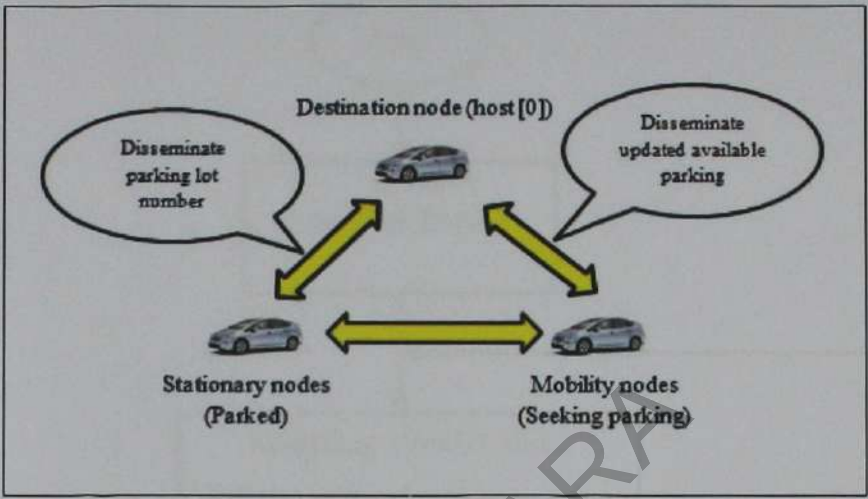

Figure 3.3: Architecture of system

This scheme consists of three components; destination nodes, stationary nodes and mobility nodes (end user). Destination node is composed of stationary nodes (vehicles parked). It work as parking monitoring units and data dissemination units which gather the information necessary for parking guidance and provide services for the vehicles in search of parking spaces. Stationary nodes are the vehicle parked. This node disseminates information of parking number to destination nodes to update vacant parking. Mobility nodes are the vehicle users who are seeking for an unoccupied parking space while entering a large parking lot. This node requests a message to destination node in order to get parking map and updated parking availability. Over this this architecture, destination node monitors the corresponding parking lot and establishes a parking occupancy map in real time based on which it provides navigation services for the incoming vehicles. Moreover, the destination node within a specified area shares their real time parking space availability information periodically which jointly responds to the parking data query request from mobility nodes. There are two advantages are finding for this parking scheme; first, cost saving which is this scheme exploits the benefits of parked vehicles and does not require any infrastructure investment. Second, time is saving which is with the parking availability information distributed; a driver seeking for an available parking lot could get a response within short time. [8]

3.4 Flow Chart of Project Methodology

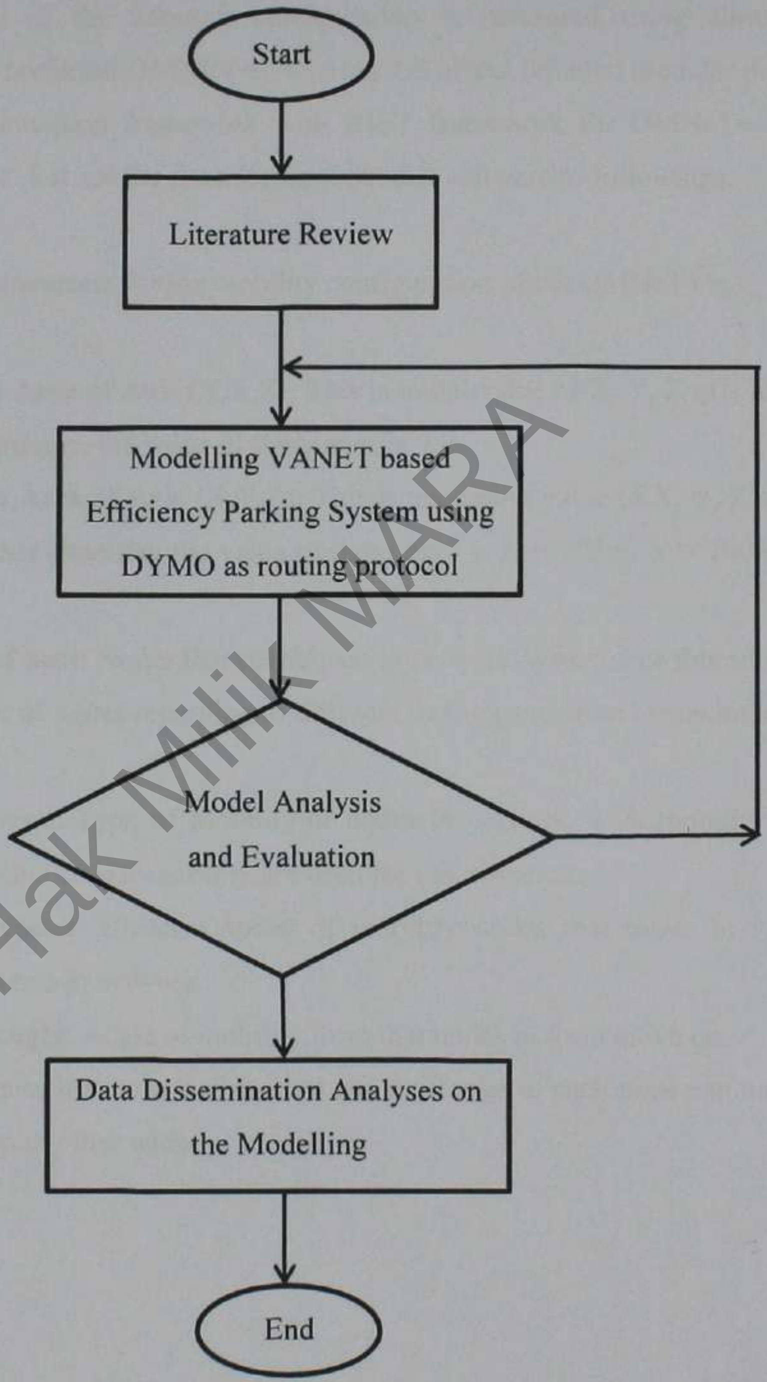

Figure 3.4: Methodology Flow Chart

3.5 Simulation Model

The performance of the network configuration is measured using simulation environment. We preferred OMNET++ Version 4.6 object oriented modular discrete event network simulation framework with INET framework for OMNeT++ with INET 3.2.0 release. Let see the description about this software as followings.

Details of input parameters for the mobility configuration of the OMNeT++:

a. **Minimum Area** of Axis (X,Y,Z): This is initial value of X, Y, Z axis in area. For this situation, the value of these axes is 0m.

b. **Maximum Area** of Axis (X,Y,Z): This is maximum value of X, Y, Z axis in area. For this situation, the value of these axes is X = 275m, Y = 200m, Z = 0m.

c. **Number of host**: Nodes that participate in network system. For this situation, the number of nodes regarding to different traffic generation is maximum 150 nodes.

d. **Mobility type**: Type of mobility of nodes in network. Two mobility types Stationary and Linear mobility are used for this situation.

e. **Mobility speed**: Uniform speed of mobility nodes that move in various mobility forms in network.

f. **Mobility angle**: Angle of mobility form that nodes in form move on.

g. **Radio transmitter communication range**: Range of each node can transmit information to other nodes.

3.5.2 Simulation Procedure

Performance evaluation parameters are set through initialization (INI) and Network Description (NED) files and result of the experiment is collected through answer (ANF) file. These are following steps can be used in constructing the modelling of VANET based Efficient Guidance Parking System

1. Install OMNeT++ 4.6 and import INET 3.2 framework to a new work space as shown in Figure 3.5.

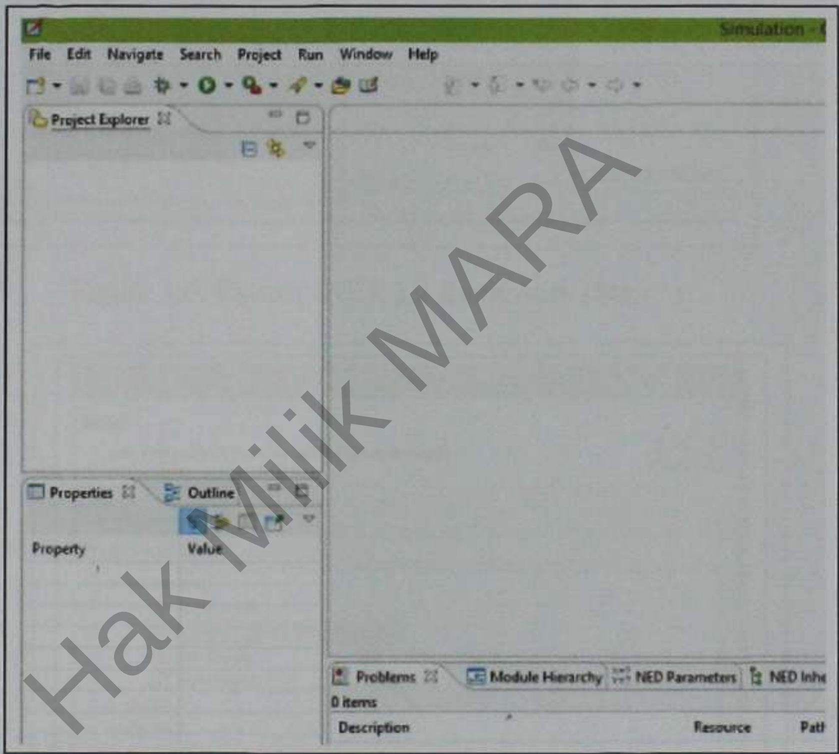

Figure 3.5: OMNeT++ workspace

2. Download INET 3.2 from OMNET++ website and extract the file to a known directory, then import INET 3.2 to the work space as shown in Figure 3.6-3.10.

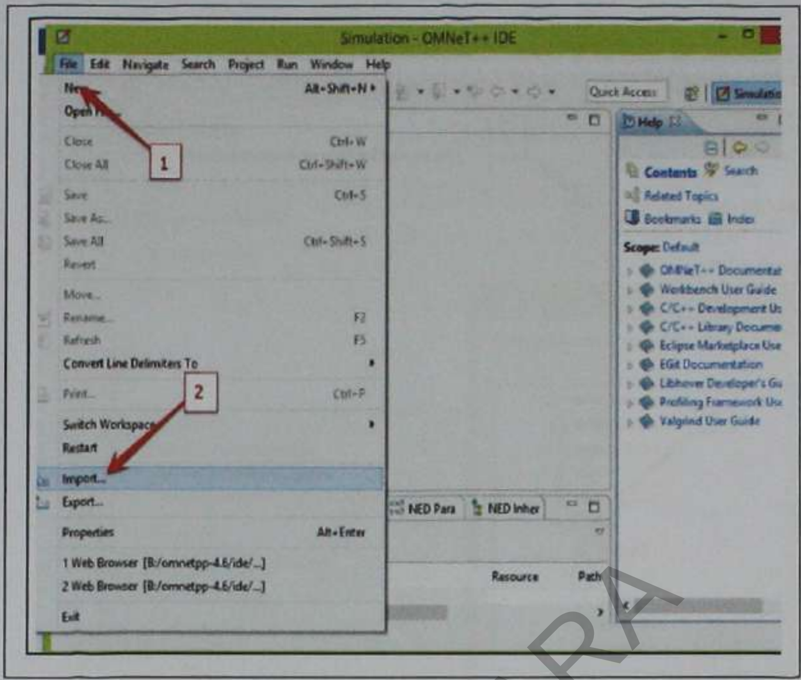

Figure 3.6: Extract INET 3.2 framework (Step 1)

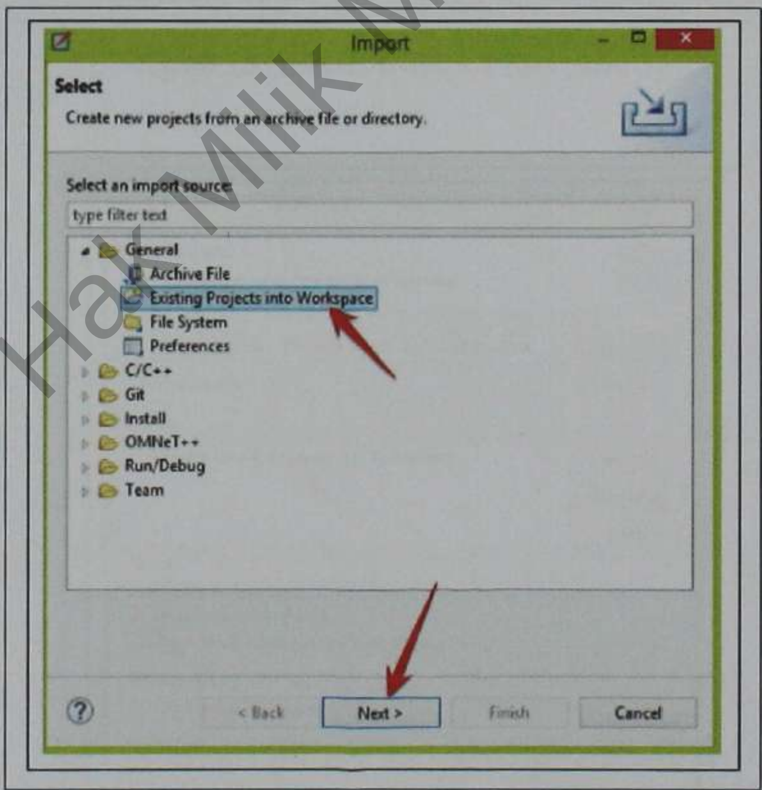

Figure 3.7: Extract INET 3.2 files (Step 2)

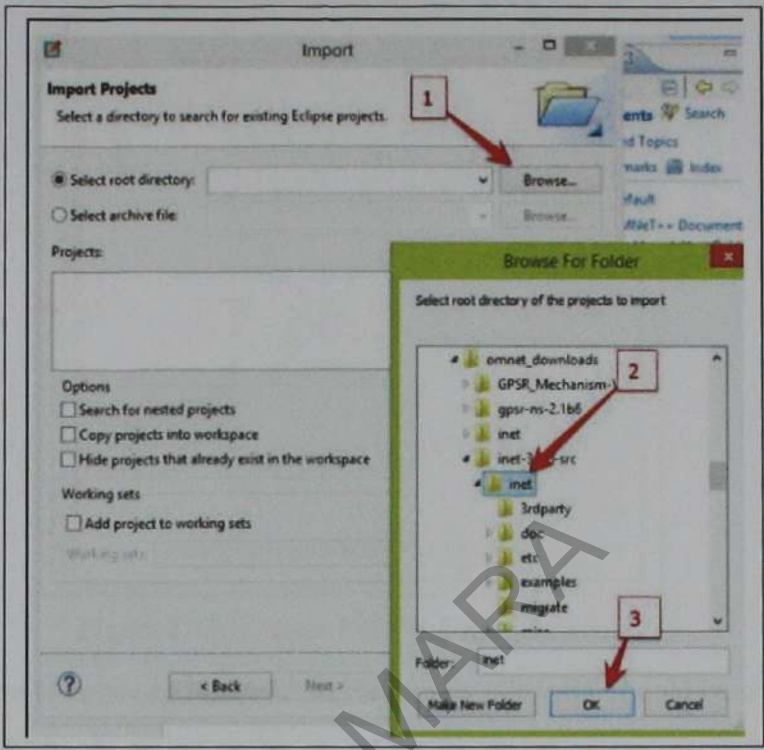

Figure 3.8: Extract INET 3.2 files (Step 3)

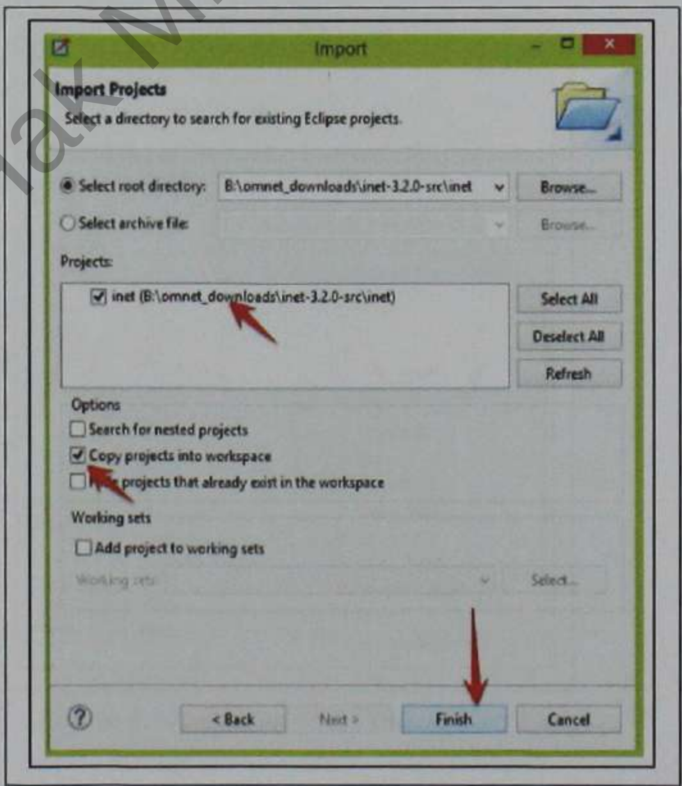

Figure 3.9: Extract INET 3.2 files (Step 4)

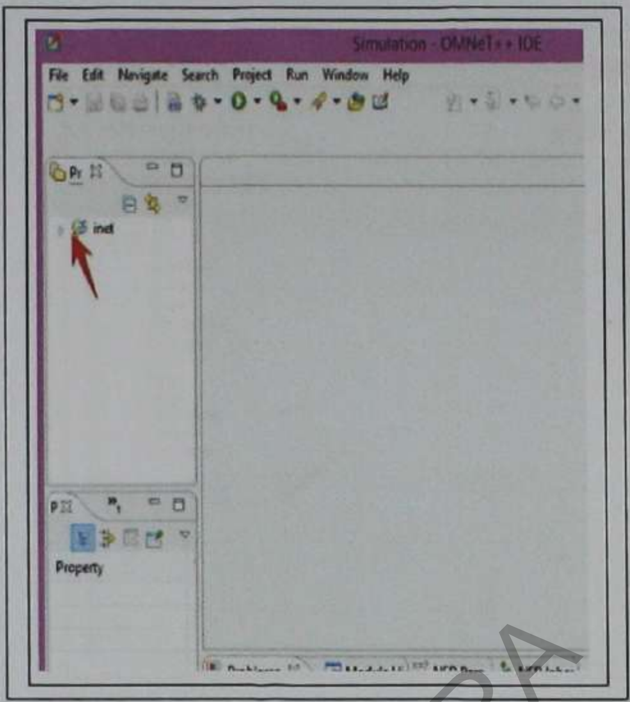

Figure 3.10: Extract INET 3.2 file (Step 5)

3. Once into is imported build the workspace by (Ctrl+B), should be waiting for a while here.

4. Create a new folder under directory (Example: adhocDymo) as shown in Figure 3.11-3.12.

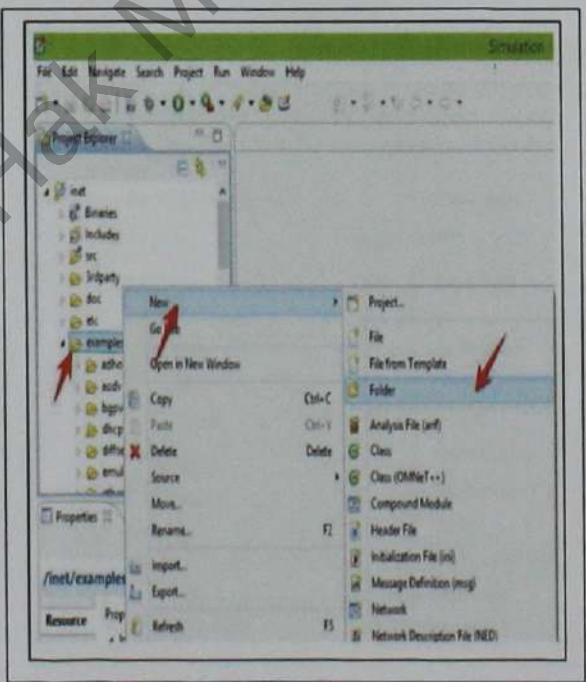

Figure 3.11: Create New Folder (Step 1)

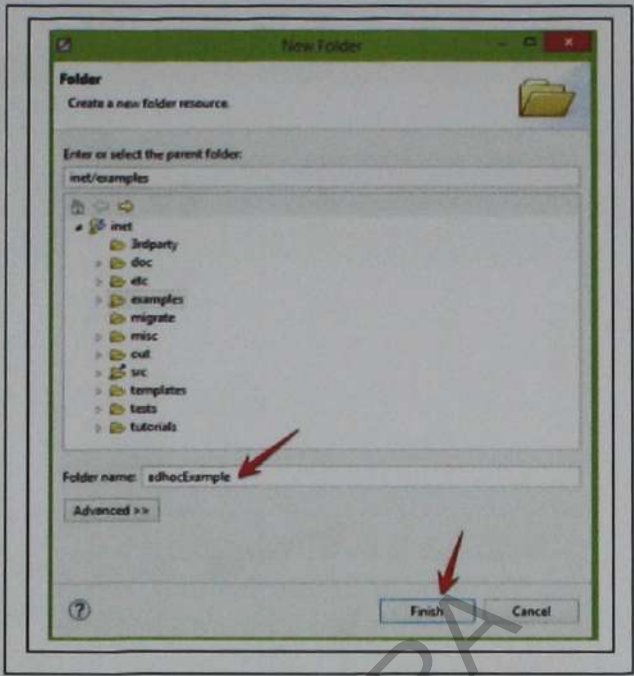

Figure 3.12: Create New Folder (Step 2)

5. Create NED file network. Under the same directory create a new NED file and name it adhocnet.ned as shown in Figure 3.13 – 3.16.

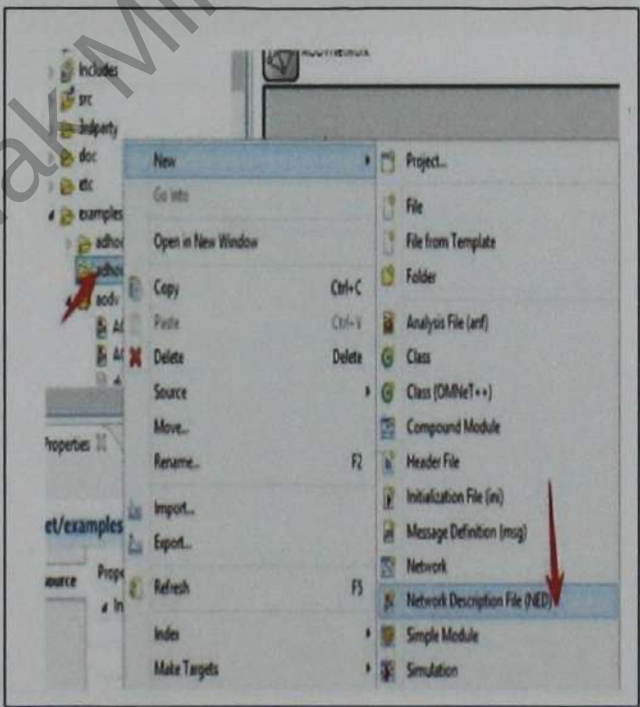

Figure 3.13: Create NED file (Step 1)

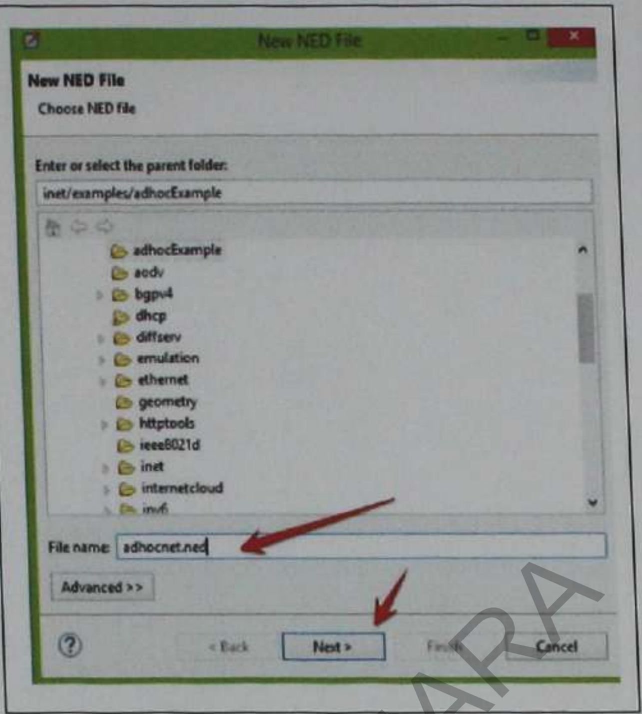

Figure 3.14: Create NED file (Step 2)

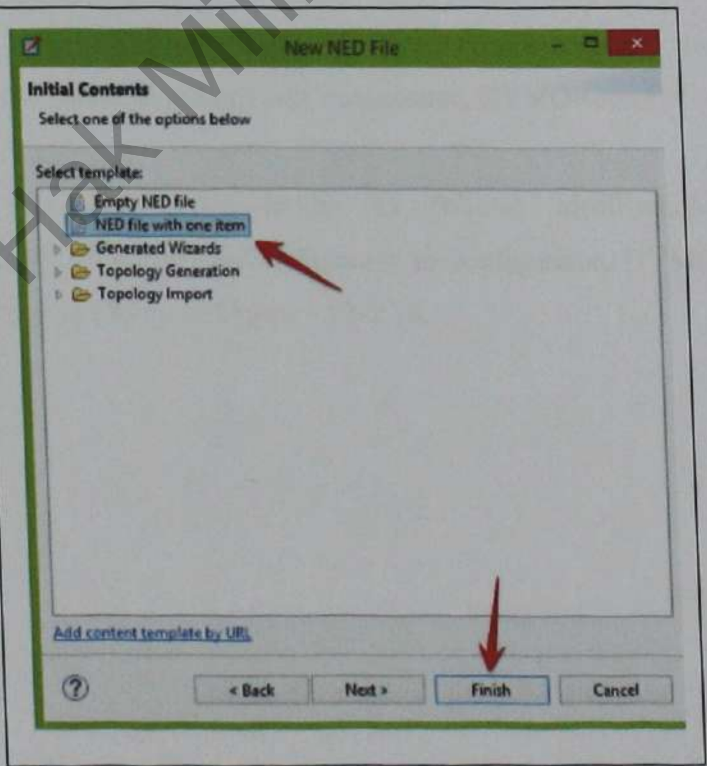

Figure 3.15: Create NED file (Step 3)

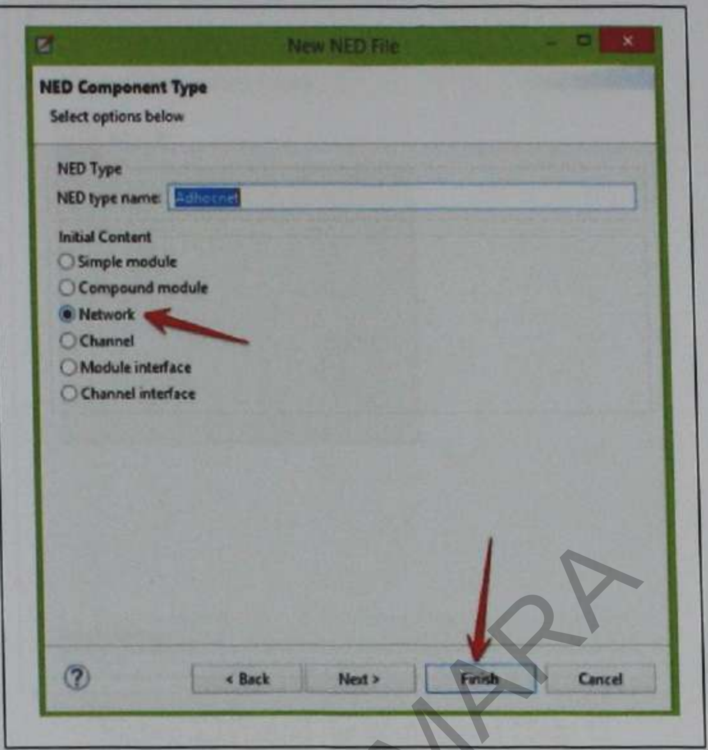

Figure 3.16: Create NED file (Step 4)

6. The next step is to add the followings components to the network; IdealRadioMedium, IPv4NetworkConfigurator, DYMORouter

7. Rename the added components as follow: IdealRadioMedium to radioMedium, IPv4NetworkConfigurator to configurator, DYMO Router to host[n_hosts] as shown in Figure 3.17-3.18.

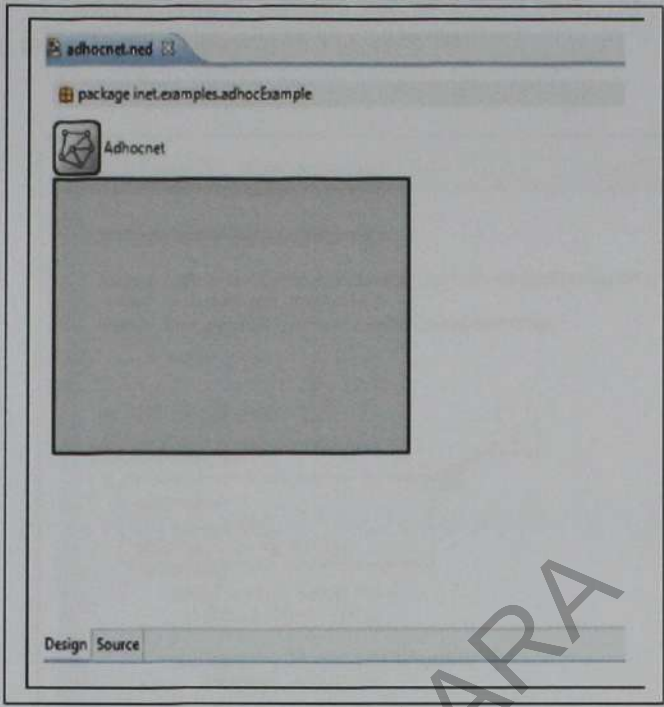

Figure 3.17: Add the component (Step 1)

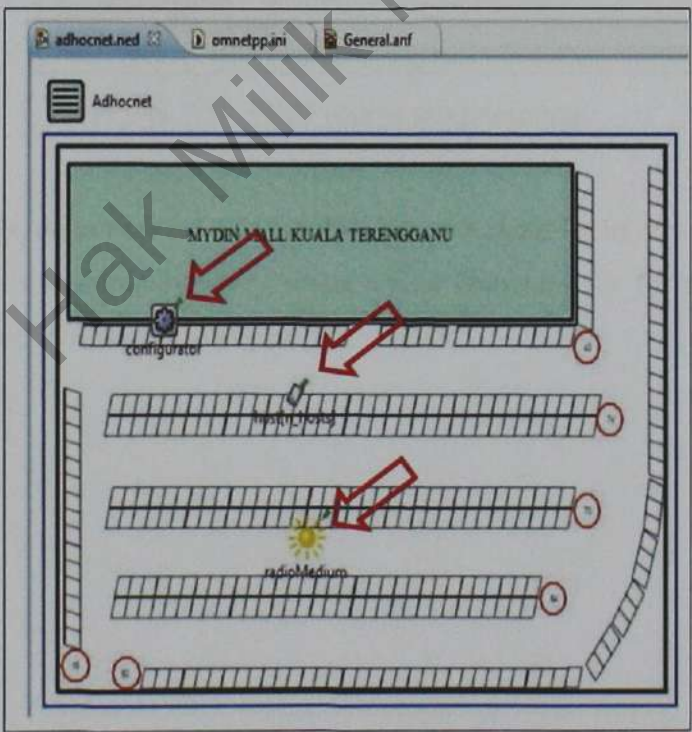

Figure 3.18: Add the component (Step 2)

8. Switch to source mode and add a parameters section to the network and create an int n_hosts as shown in Figure 3.19.

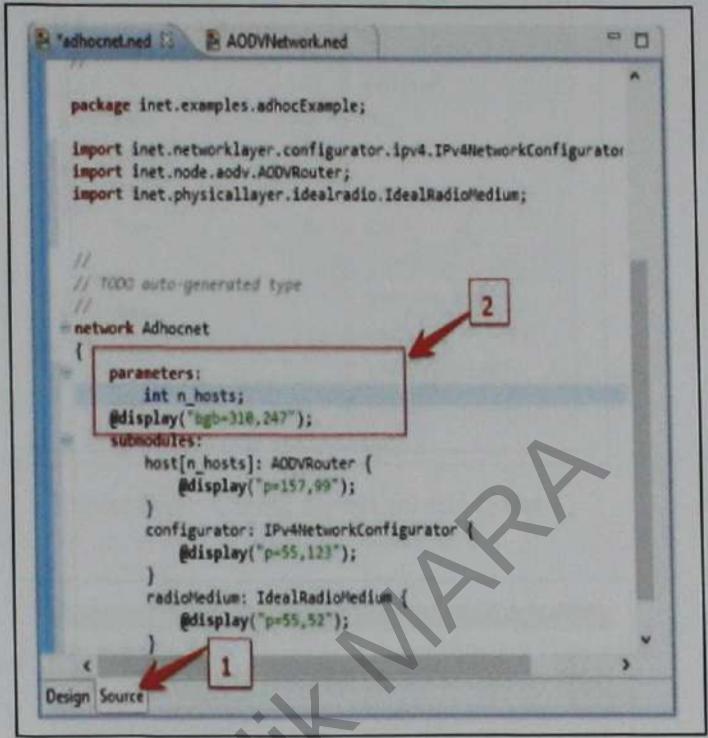

Figure 3.19: NED source programming

9. Up to this, done with the network file, now we should start create our INI file. Under this example directory create a new omnet.ini file for the network in this example as shown in Figure 3.20 – 3.25.

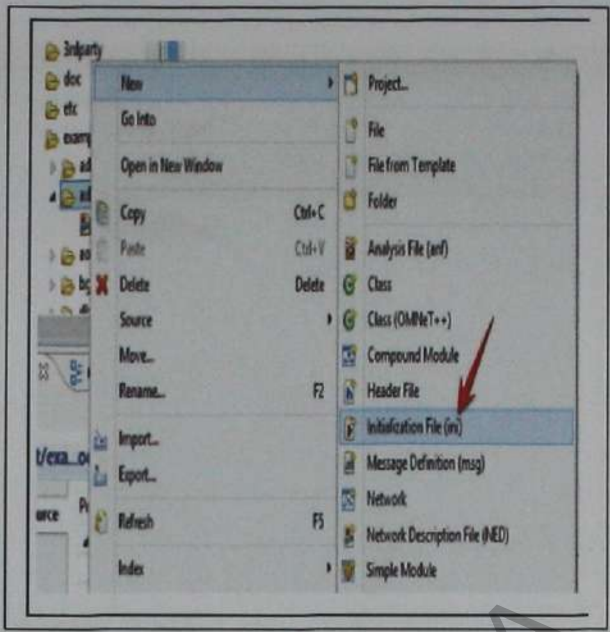

Figure 3.20: Create omnet.ini file (Step 1)

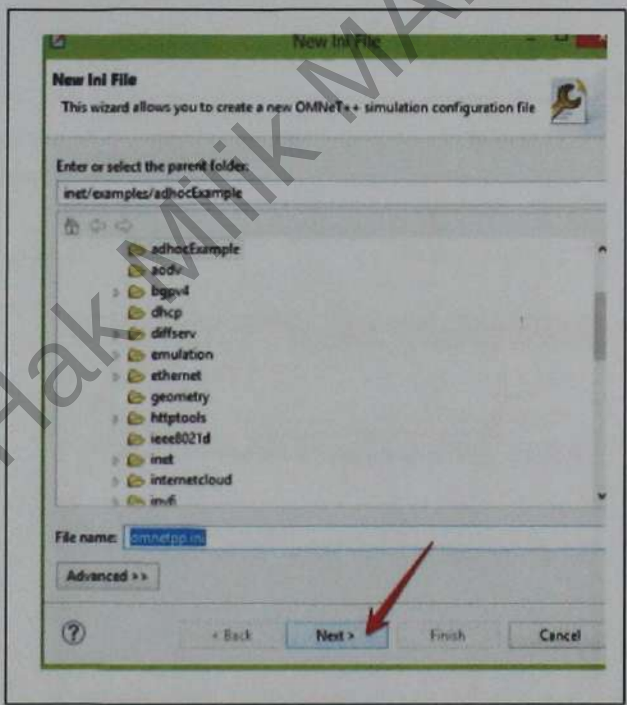

Figure 3.21: Create omnet.ini file (Step 2)

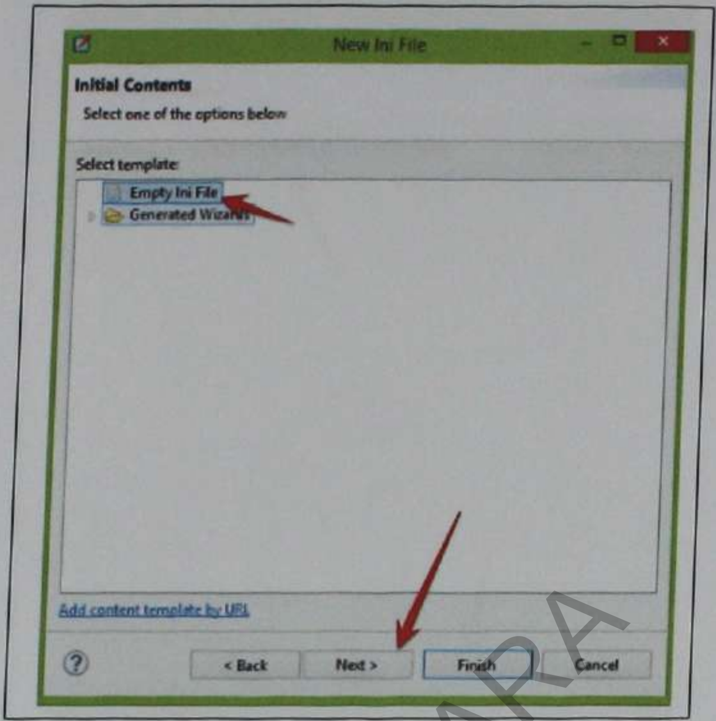

Figure 3.22: Create omnet.ini file (Step 3)

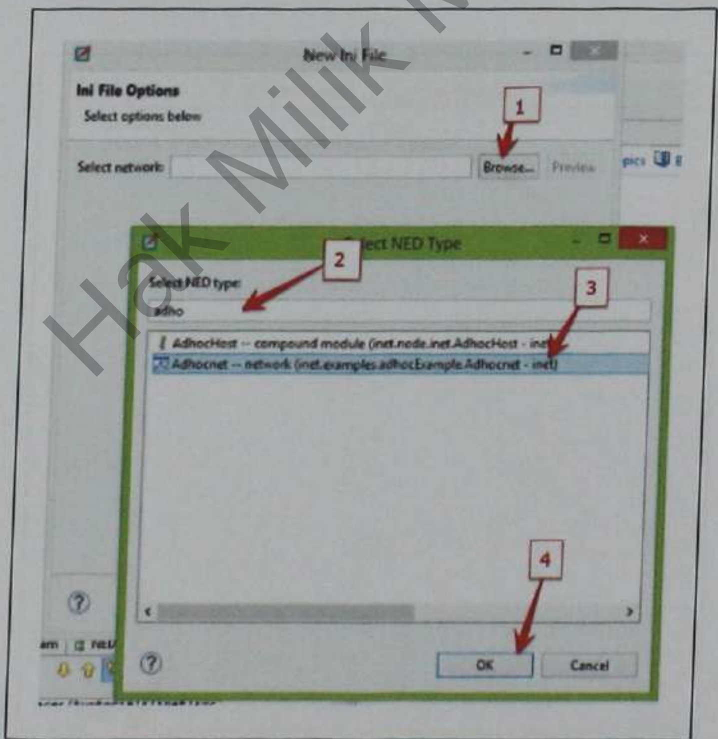

Figure 3.23: Create omnet.ini file (Step 4)

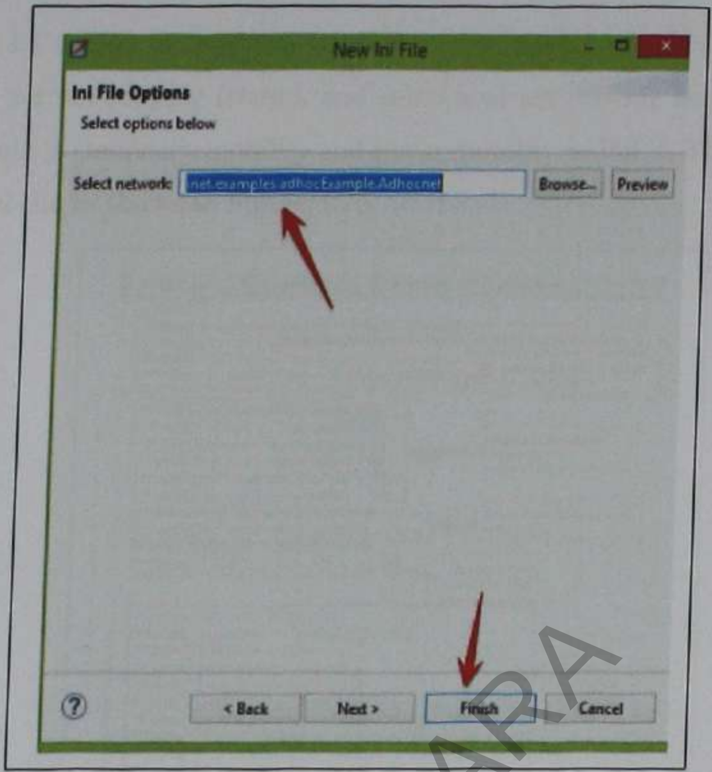

Figure 3.24: Create omnet.ini file (Step 5)

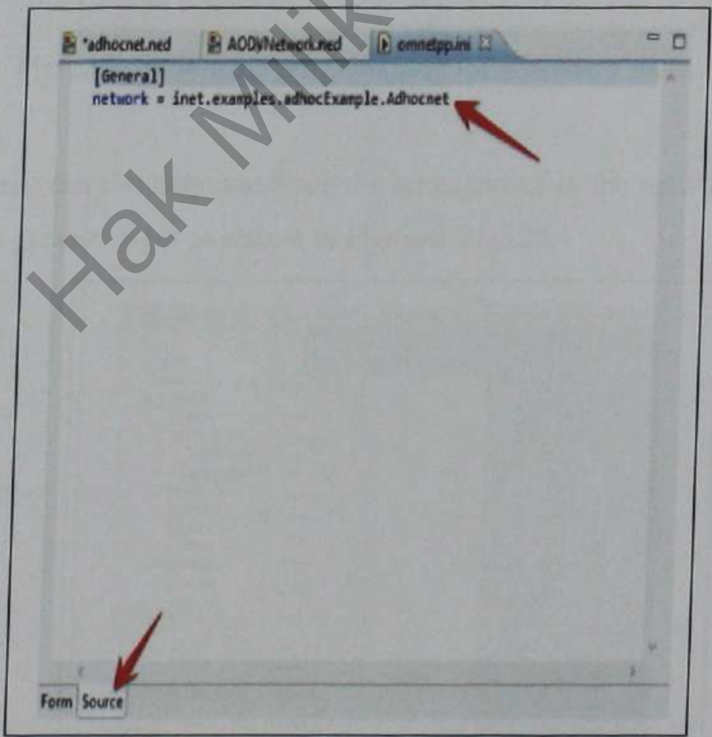

Figure 3.25: Create omnet.ini file (Step 6)

10. First let's setup the mobility configurations for all hosts: In this project some host are not moving (static), and some host are moving hence the mobility module is stationary mobility and linear mobility in INET. The programming in INI file as shown in Figure 3.26.

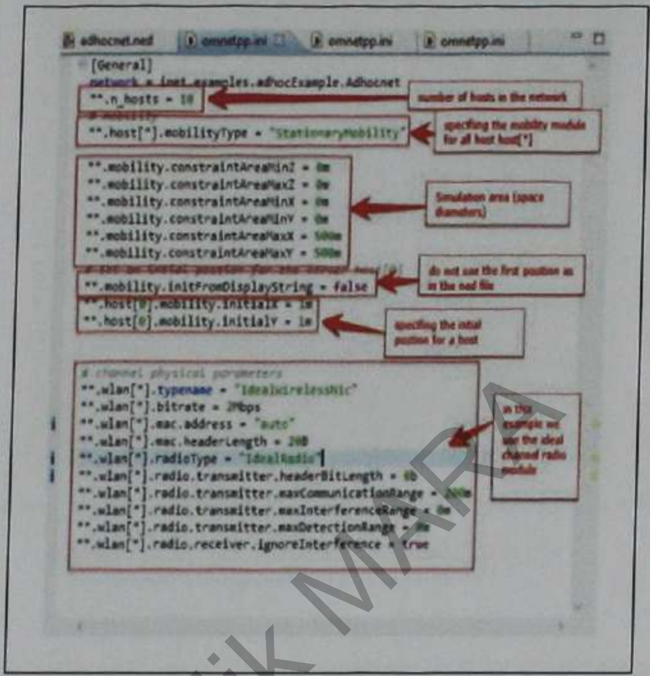

Figure 3.26: INI file programming for stationary mobility

11. Now run the simulation and see the arrangement in the network; however no traffic generated yet as shown in Figure 3.27-3.28.

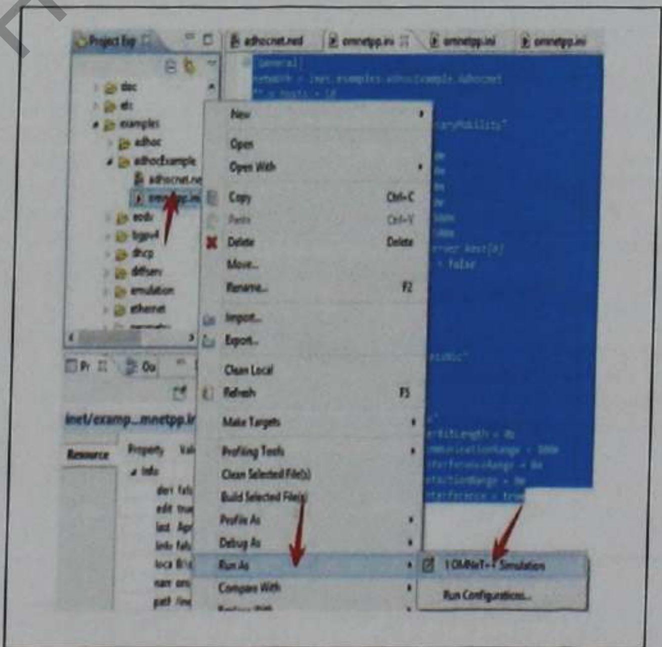

Figure 3.27: Run the simulation (Step 1)

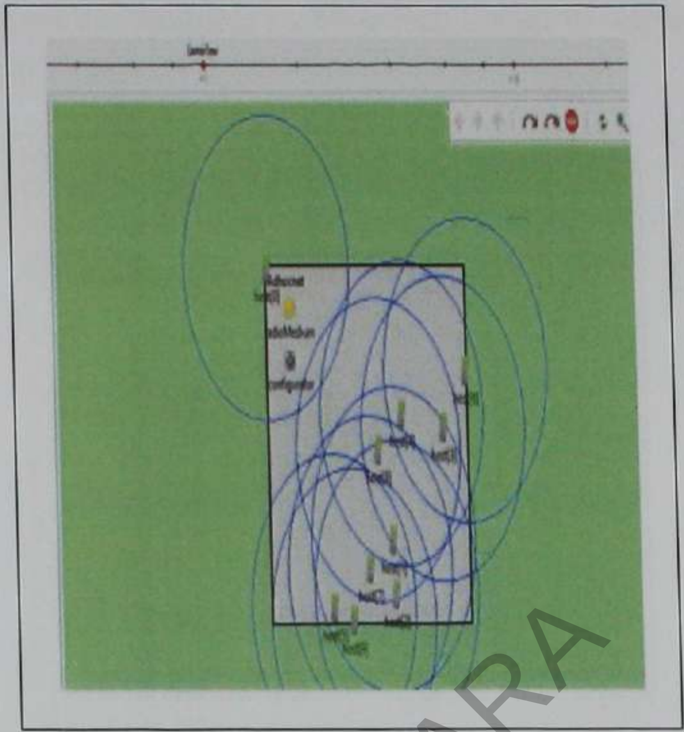

Figure 3.28: Run the simulation (Step 2)

12. Now close the run and move to the omnetpp.ini file to add some traffic to this network. All hosts have a UDPBasicBurst application, but host[0] is a sink. All host are to generate a traffic toward host[0] as 2 packets per second and each packet is 512Byte in size as shown in Figure 3.29.

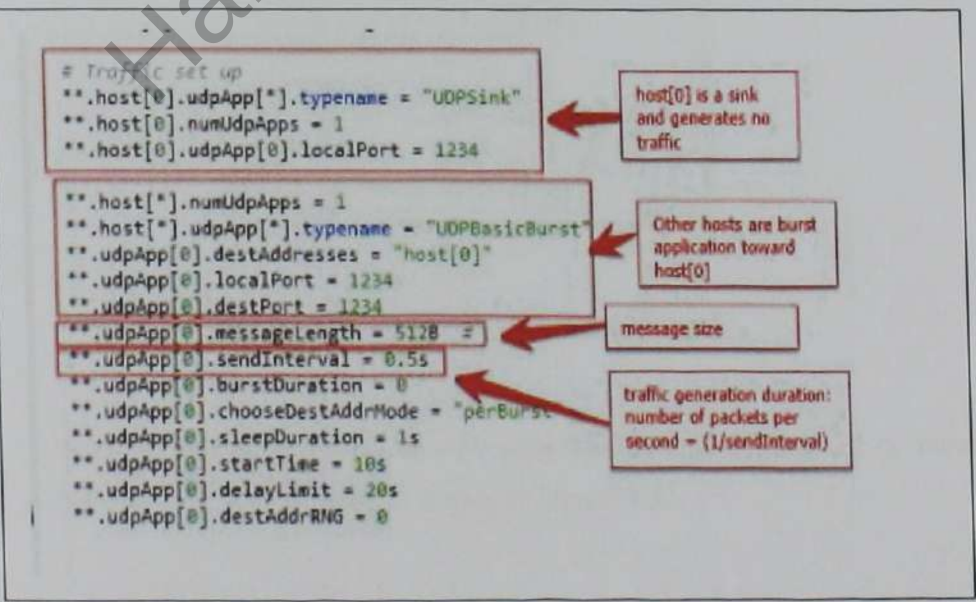

Figure 3.29: Traffic generation in INI file

13. Now let's run the simulation for 50 seconds and see the results as shown in Figure 3.30 – 3.31.

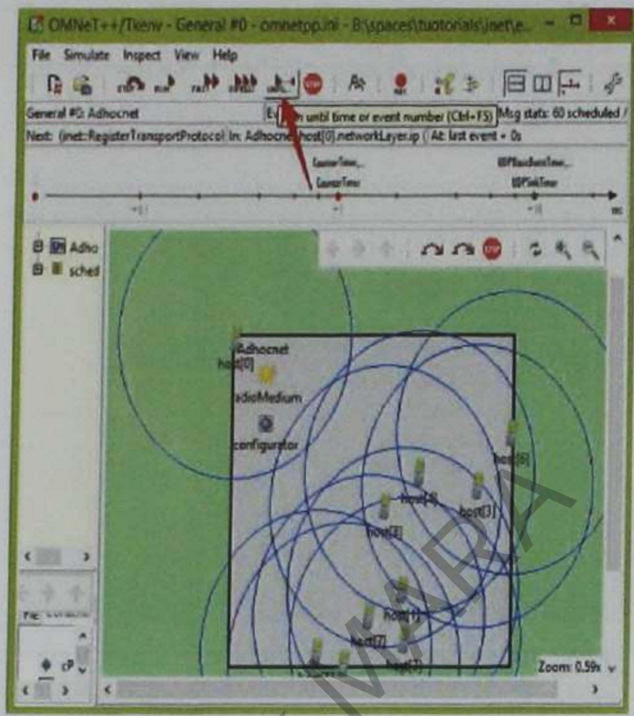

Figure 3.30: Generate simulation (Step 1)

Figure 3.31: Generate simulation (Step 2)

14. Once simulation is closed an auto generated results folder will be created under the project directory as shown in Figure 3.32.

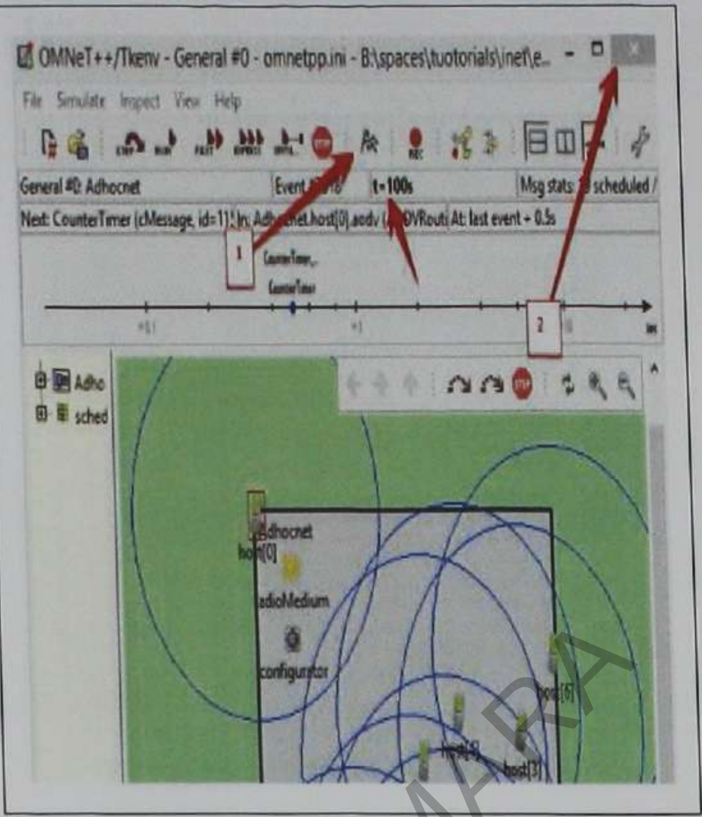

Figure 3.32: Result of simulation (Step 1)

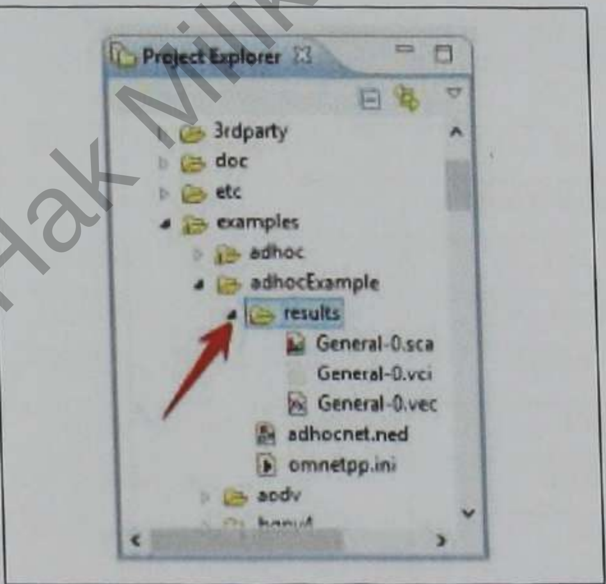

Figure 3.33: Result of simulation (Step 2)

15. Now open the .sca file and observe the results. We can see that no data is received by host[0], and that's is due to short communication range as all hosts are out of range as shown in Figure 3.33 – 3.34.

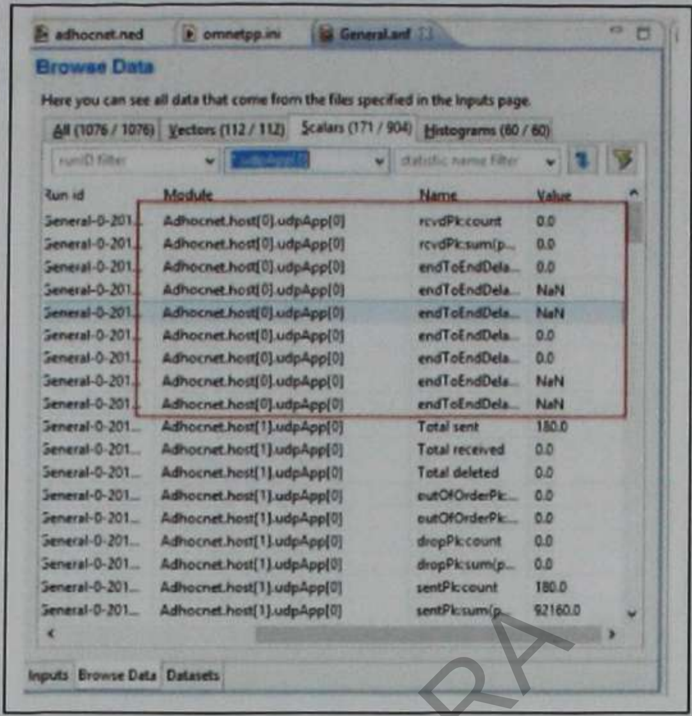

Figure 3.34: Result of simulation (Step 3)

Table 3.1: Parameter used in simulation

PARAMETER	VALUE
Operating System	Win7
OMNeT++ Version	4.6
Channel Type	IdealWirelessNic
Number of nodes	50, 100, 150
Stationary nodes	40, 80, 130
Mobility nodes	10, 20, 20
Speed (Mbps)	10
Data type	UDP
Simulation time (s)	50
Maximum communication range (m)	50
MAC protocol	MAC/802.11
Data packet size (byte)	100, 200, 300, 400, 500
Area of simulation (m^2)	200*275
Radio propagation model	IdealRadio
Routing protocol	DYMO

16. For this study, simulation is done for 50, 100 and 150 nodes respect to different packet size as shown in Table 3.1, Figure 3.35 – 3.37.

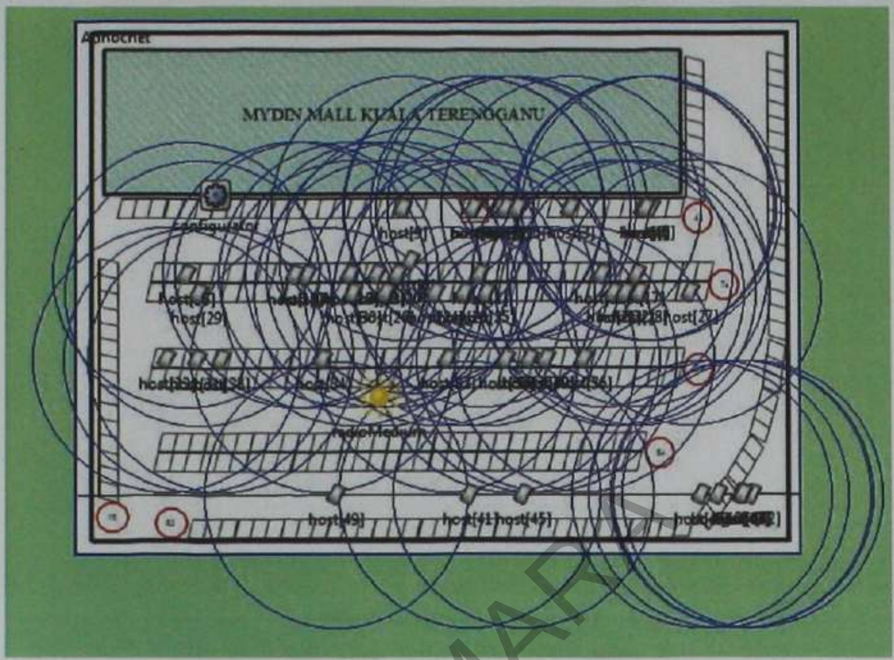

Figure 3.35: Simulation for 50 nodes

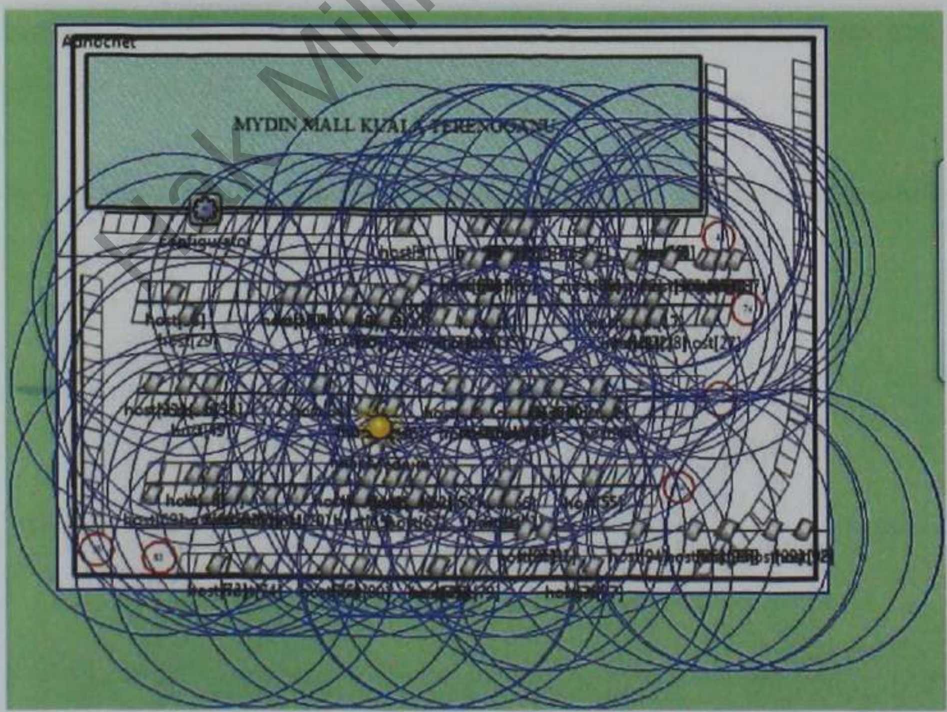

Figure 3.36: Simulation for 100 nodes

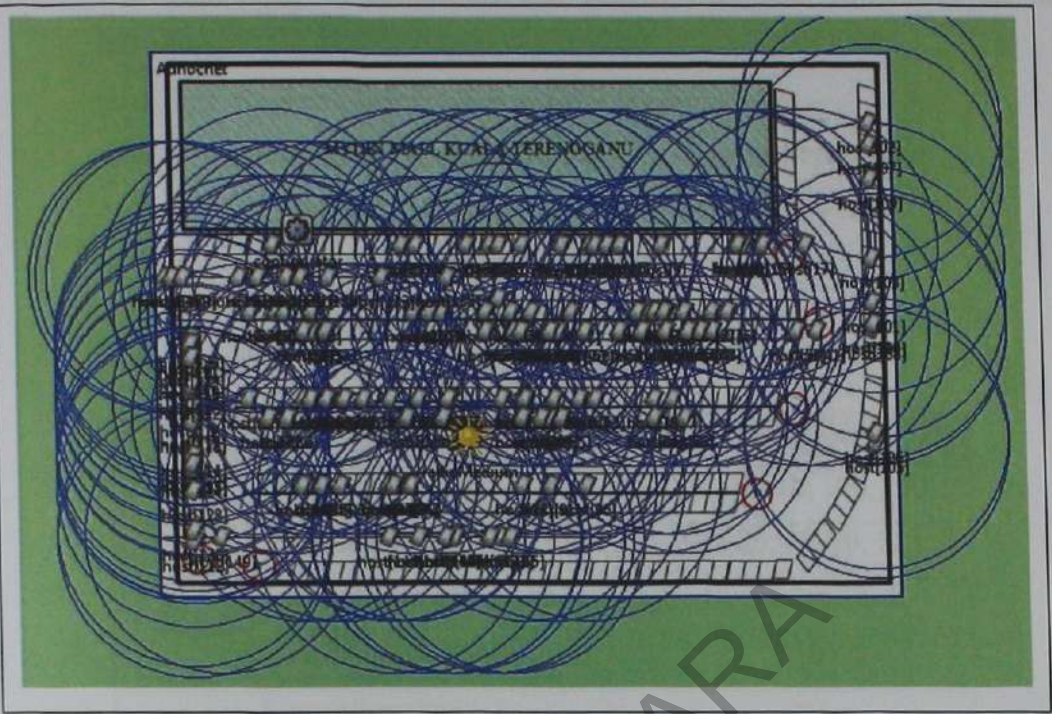

Figure 3.37: Simulation for 150 nodes

17. Throughput values of the simulation is collected in excel file. Data collected results at 50, 100, 150 nodes. These results are collected for different packet sizes 100, 200, 300, 400 and 500 byte.

18. To set simulation in random number generator, INI file programming modified as shown in Figure 3.38.

```
[General]
network = inet.examples.adhocDymo.Adhocnet
num-rngs = 1
repeat = 2
seed-set = ${runnumber}
sim-time-limit = 50s
**.vector-recording = false
```

Figure 3.38: INI commands for seed simulation set

19. The description of commands that related with seed simulation set as followings.

num-rngs = 1 # multiple random number generators

seed-set = ${runnumber} # default, every run has a different seed

seed-set = ${repetition} # measurements have the same seed

seed-set = ${5,6,8..11} # set seed manually

Table 3.2: Parameter changes for seed simulation

PARAMETER	VALUE
Number of nodes	20, 30, 50, 100, 150
Stationary nodes	10, 20, 30, 70, 120
Mobility nodes	10, 10, 20, 30, 30
Speed (Mbps)	10
Simulation time (s)	50
Maximum communication range (m)	100
Data packet size (byte)	512

20. The setting of simulation is modified for other results as shown in Table 3.2. The other settings is remains. The results of simulation for Table 3.2 is collected and analyzed.

3.6 Performance Evaluation

To evaluate the performance of routing protocols in this parking system, this study uses three different quantitative metrics to evaluate the performance of DYMO. There are as followings;

- **Throughput**: The throughput is defined as the total amount of data a receiver receives from the sender divided by the time it takes for the receiver to get the last packet [13].
- **Packet Delivery Ratio (PDR)**: The fraction of packets sent by the application that are received by the receivers [13]. The ratio of data packets delivered successfully to destination nodes and the total number of data packets generated for those destinations. PDR characterizes the packet loss rate which limits the throughput of the network. The higher the delivery ratio, performance of the routing protocol is better [15].
- **Packet Data Size**: The packets transmitting from source to destination. Once changing the packet size also changing the performance of networks [18].
- **Average End-to-end delay**: End-to-end delay indicates how long it took for a packet to travel from the source to the application layer of the destination [13]

CHAPTER 4

PERFORMANCE IMPROVEMENT OF VANET BASED EFFICIENT PARKING SYSTEM USING OMNeT++

4.1 Introduction

In this section, performance results for the DYMO routing protocol in term of Packet Delivery Ratio (PDR) and throughput respect to different traffic generation and packet size are evaluated. There are two run configuration of simulation; first is simulation running with default setting and the second is simulation running with command line which is the simulation is running with seed at random generator. For default simulation, data collected upon of different packet size with different number of nodes while for command line simulation data collected upon of different number of nodes. The details of parameters index has mention in Chapter 3.

4.2 Simulation Results

The simulation results for packet delivery ratio measured for the performance of routing protocol, DYMO in VANET based parking system are shown in Table 4.1, 4.2, 4.3 for 50, 100, and 150 nodes at different packet size 100, 200, 300, 400 and 500 bytes. Table 4.4 shows the results for simulation parameters with different traffic generation.

Table 4.1: Simulation results for 50 nodes

PARAMETER	VALUE				
Data Packet Size (Bytes)	100	200	300	400	500
Stationary nodes	40	40	40	40	40
Mobility nodes	10	10	10	10	10
Acknowledge packets	42554824	38106454	37012740	51331036	39366246
Transmitted packets	50123004	45487586	43681396	59971996	46597384
Throughput	6808772	6097033	5922038	8212966	6298599
Packet Delivery Ratio (PDR	0.84901	0.83773	0.84733	0.85592	0.84482

Table 4.2: Simulation results for 100 nodes

PARAMETER	VALUE				
Data Packet Size (Bytes)	100	200	300	400	500
Stationary nodes	80	80	80	80	80
Mobility nodes	20	20	20	20	20
Acknowledge packets	70886882	78683576	68488598	83549454	78273288
Transmitted packets	85289482	94086432	82666056	99831330	93772836
Throughput	11341901	12589372	10958176	13367916	12523726
Packet Delivery Ratio (PDR)	0.83112	0.83629	0.828497	0.83691	0.83471

Table 4.3: Simulation results for 150 nodes

PARAMETER	VALUE				
Data Packet Size (Bytes)	100	200	300	400	500
Stationary nodes	130	130	130	130	130
Mobility nodes	20	20	20	20	20
Acknowledge packets	82481358	90121276	89202036	96013128	95217024
Transmitted packets	99856060	108727996	107136078	115620156	114250900
Throughput	13197017	14419404	14272358	15362100	15234724
Packet Delivery Ratio (PDR)	0.82600	0.82886	0.83261	0.83042	0.8334

Table 4.4: Simulation results for different traffic generation

PARAMETER	VALUE				
Number of nodes	20	30	50	100	150
Stationary nodes	10	20	30	70	120
Mobility nodes	10	10	20	30	30
Acknowledge packets	11655390	15731520	24569558	32865286	55528378
Transmitted packets	13878430	19033382	29530026	40300510	67729974
Throughput	1864862	2517043	3931129	5258446	8884540
Packet Delivery Ratio (PDR)	0.83982	0.82652	0.83202	0.81551	0.81985

A. Packet Delivery Ratio

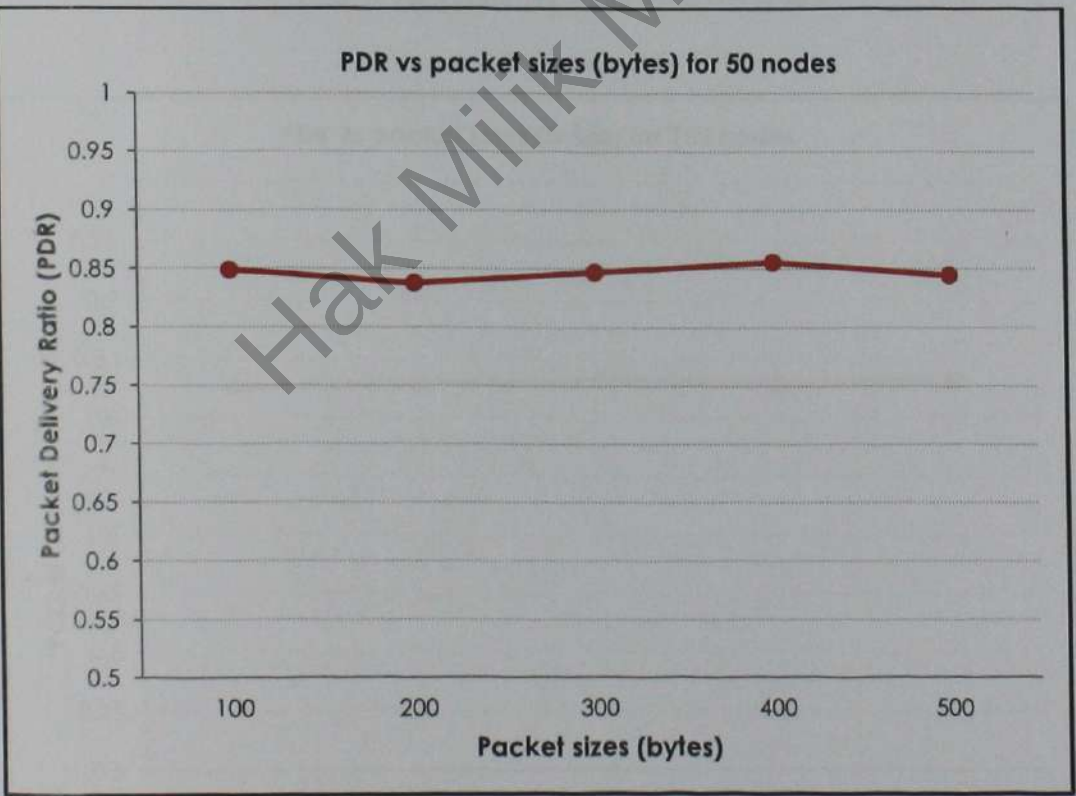

Figure 4.1: Packet delivery ratio (PDR) for 50 nodes

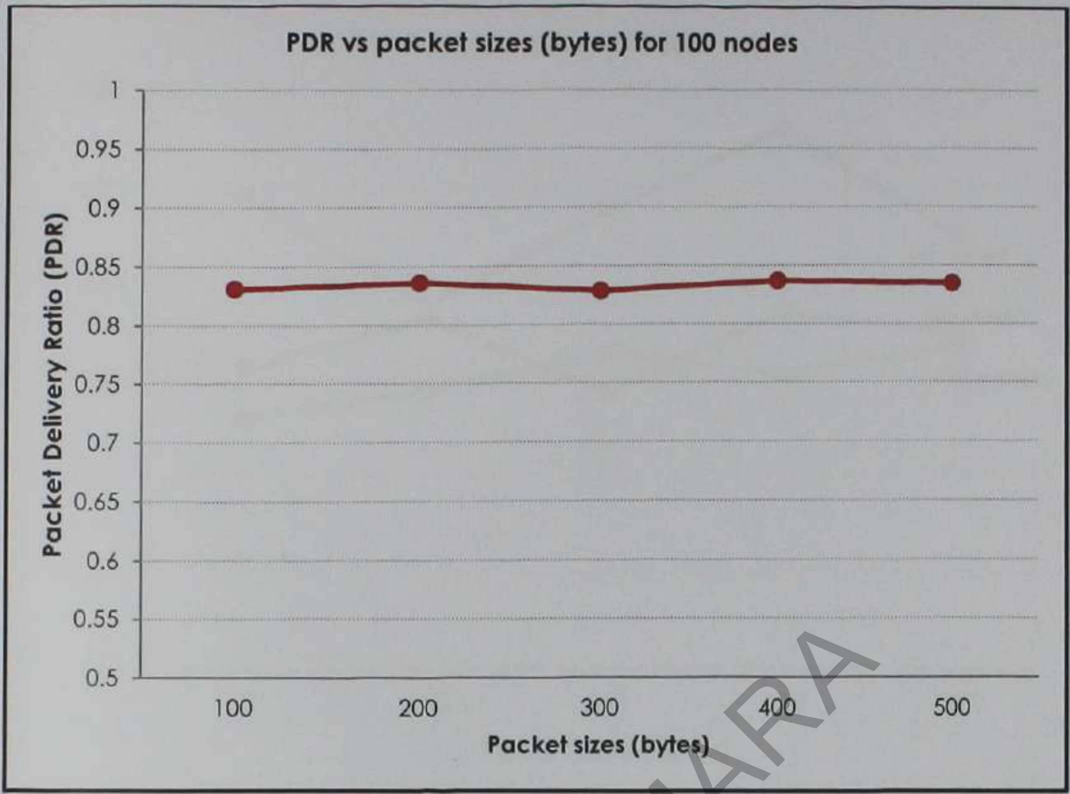

Figure 4.2: Packet delivery ratio for 100 nodes

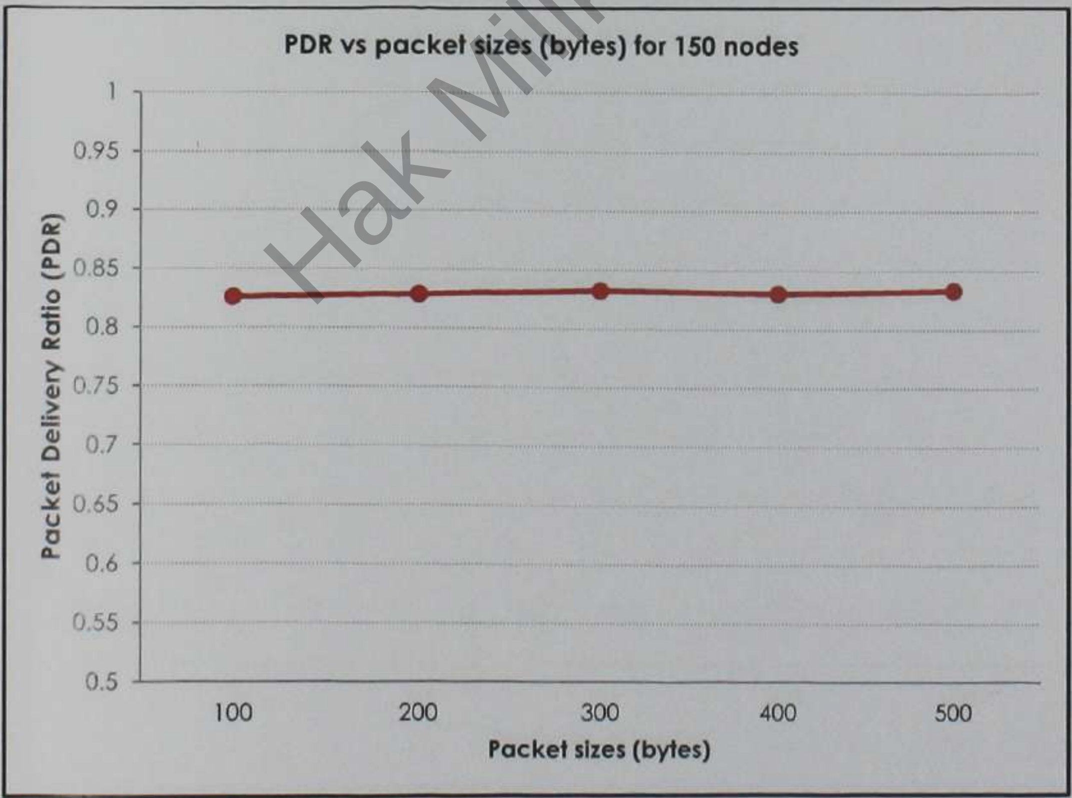

Figure 4.3: Packet delivery ratio for 150 nodes

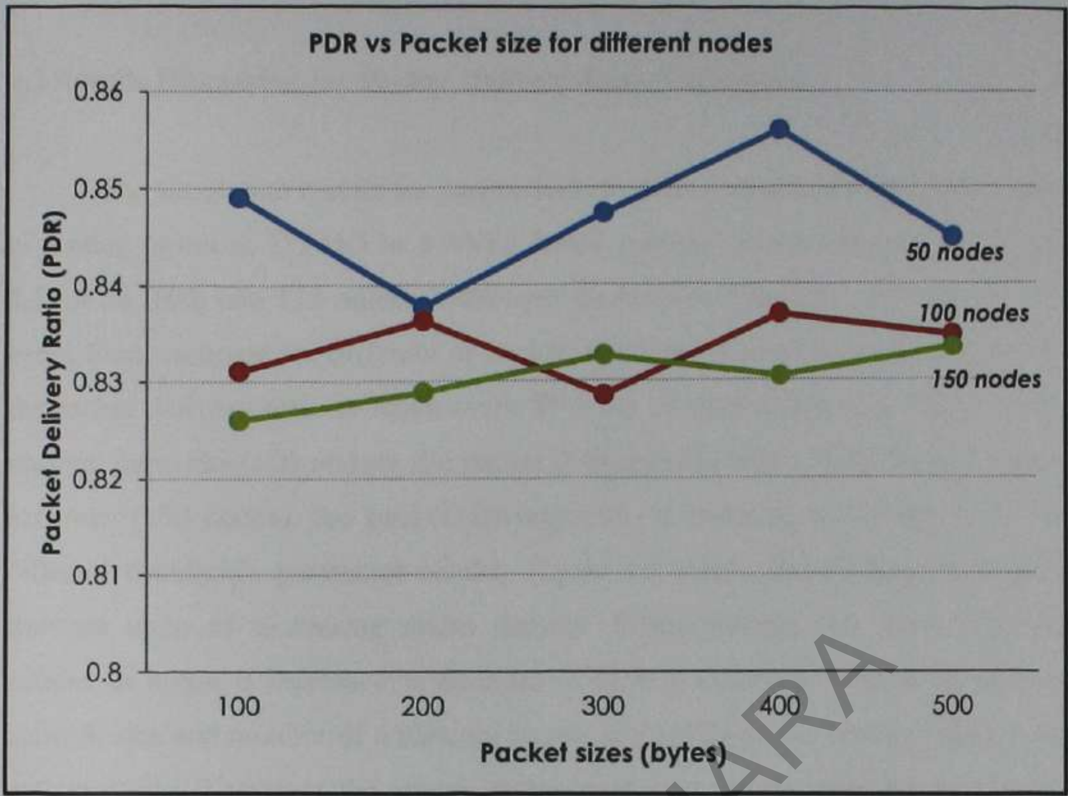

Figure 4.4: Packet delivery ratio for different nodes and packet size

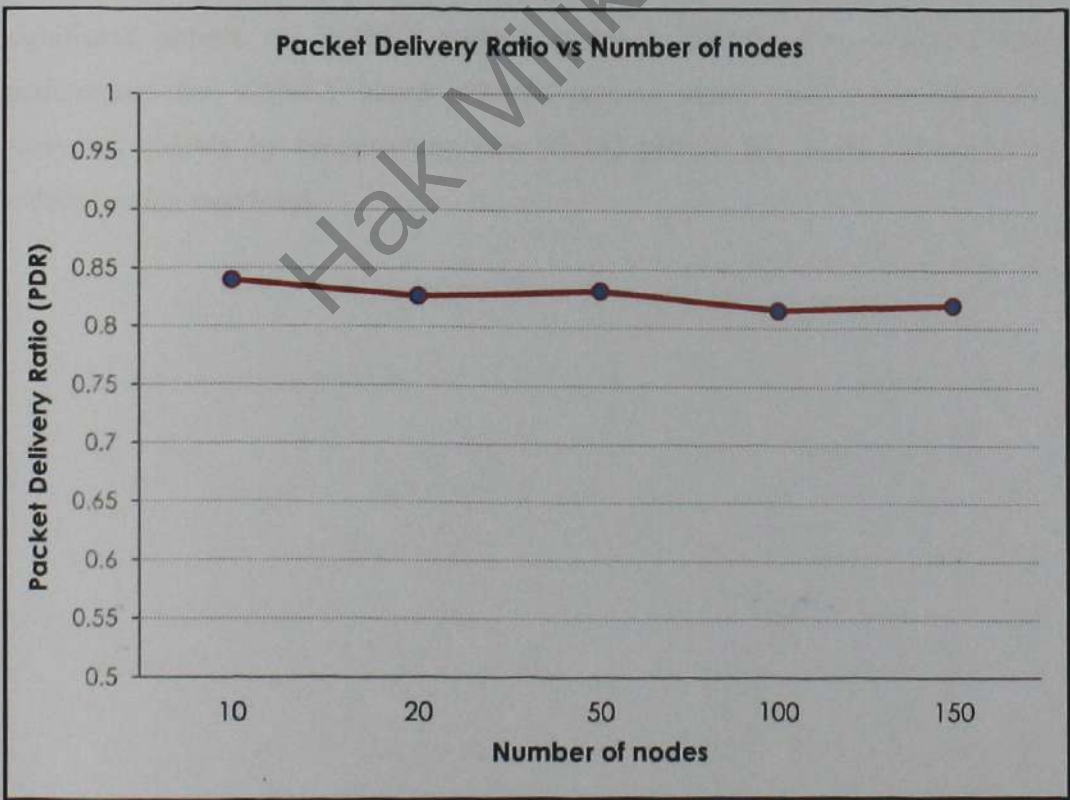

Figure 4.5: Packet delivery ratio for different nodes

4.3 Results Discussion for Packer Delivery Ratio Parameter

The simulation results for packet delivery ratio measured for the performance of routing protocol, DYMO in VANET based parking system are shown in Figure 4.4 for 50, 100, and 150 nodes at different packet size 100, 200, 300, 400 and 500 bytes. First, compare the different of packet data size for small networks (50 nodes), the packet delivery ratio is reach more 85% for all packet data size. However for medium networks (100 nodes), the packet delivery ratio below of 85% and for large networks (150 nodes), the packet delivery ratio is decrease to average 83%. For different simulation parameter results, Figure 4.5 shows that PDR is moderately decrease upon of increasing nodes density. When network size along with the number of nodes is increased, a decrease in PDR is observed. With an increase in network size and number of nodes, the routes at a node become obsolete more often with mobility. Therefore the source nodes need to re-initiate path discovery more frequently in large networks. Therefore in VANET based efficient parking system modelling, increase in packet data size and number of nodes does not have significant impact on DYMO routing protocol. DYMO give relatively better performance for VANET based efficient parking model since route failures are recovered quickly by broadcasting new RREQ packets and at the same time can reduce routing overhead.

B. Throughput

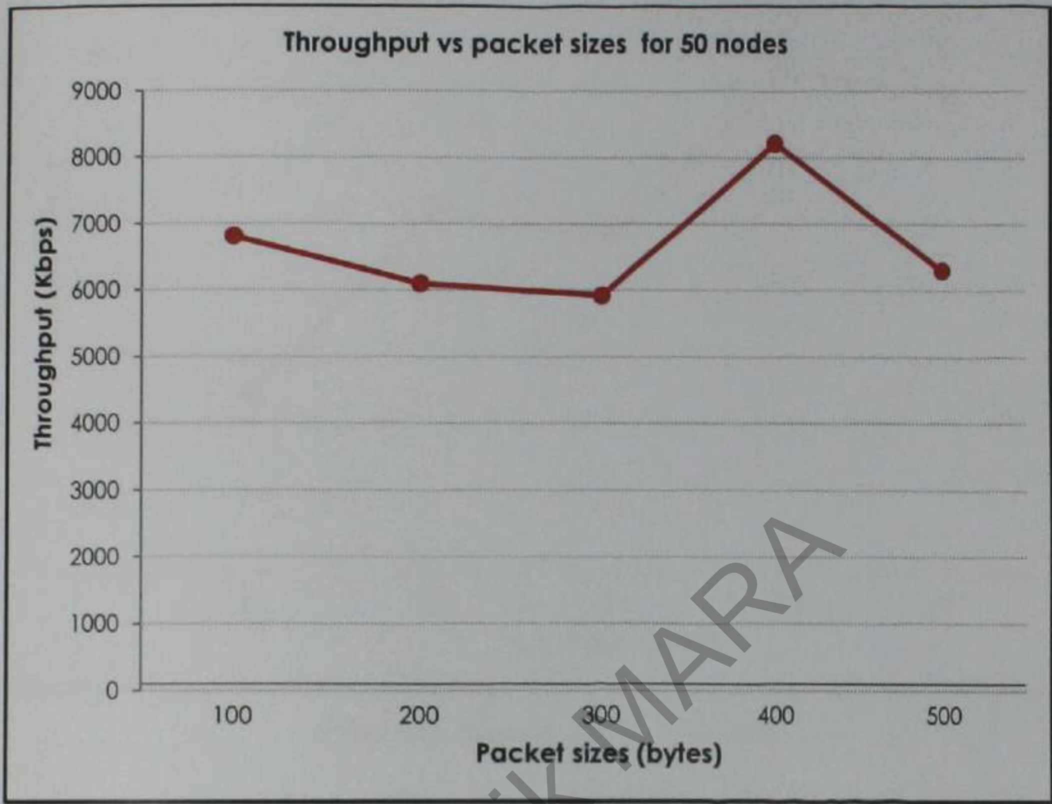

Figure 4.6: Throughput for 50 nodes

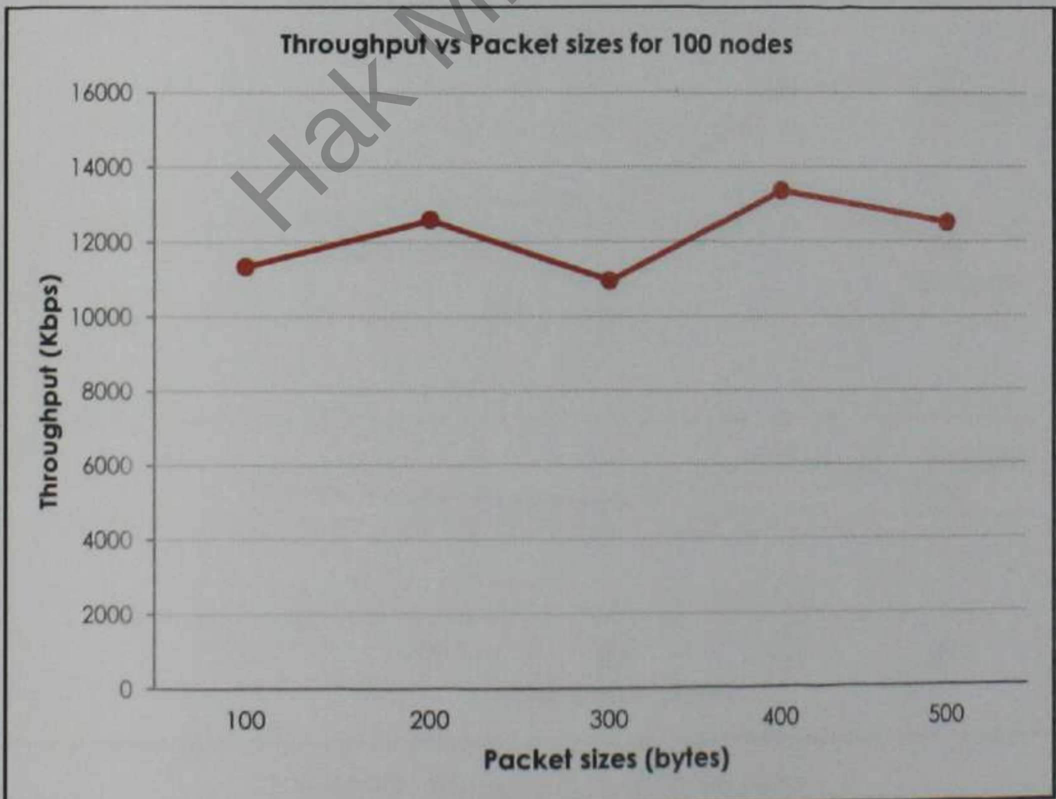

Figure 4.7: Throughput for 100 nodes

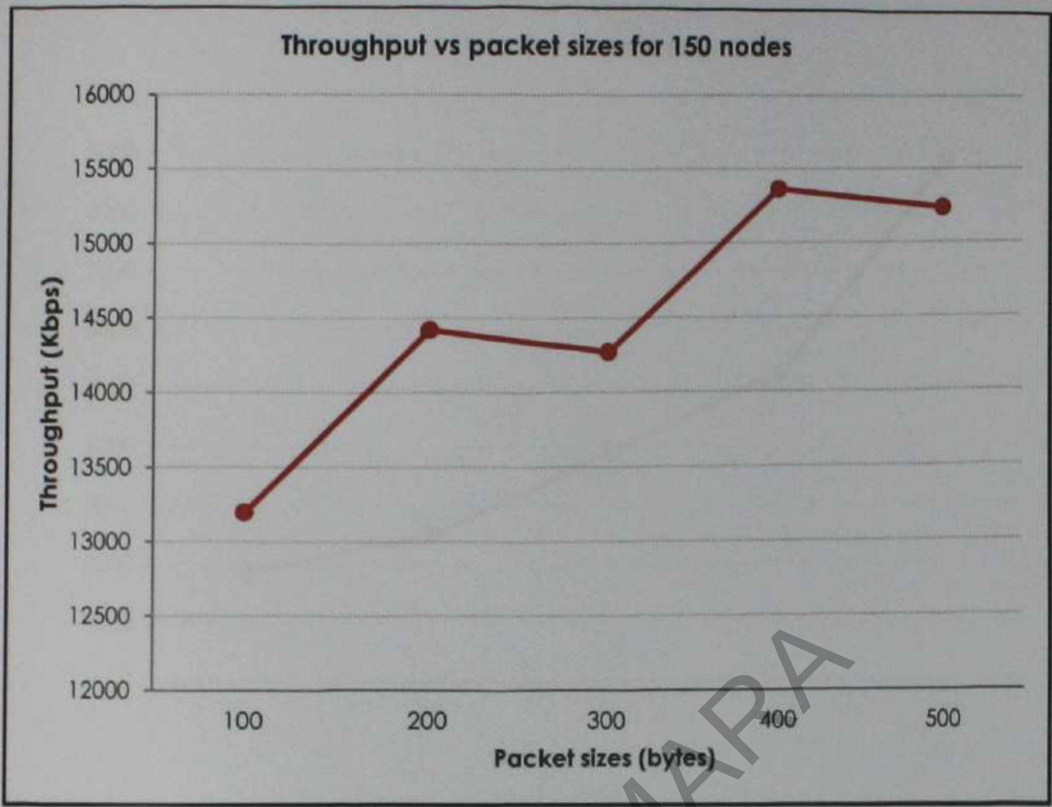

Figure 4.8: Throughput for 150 nodes

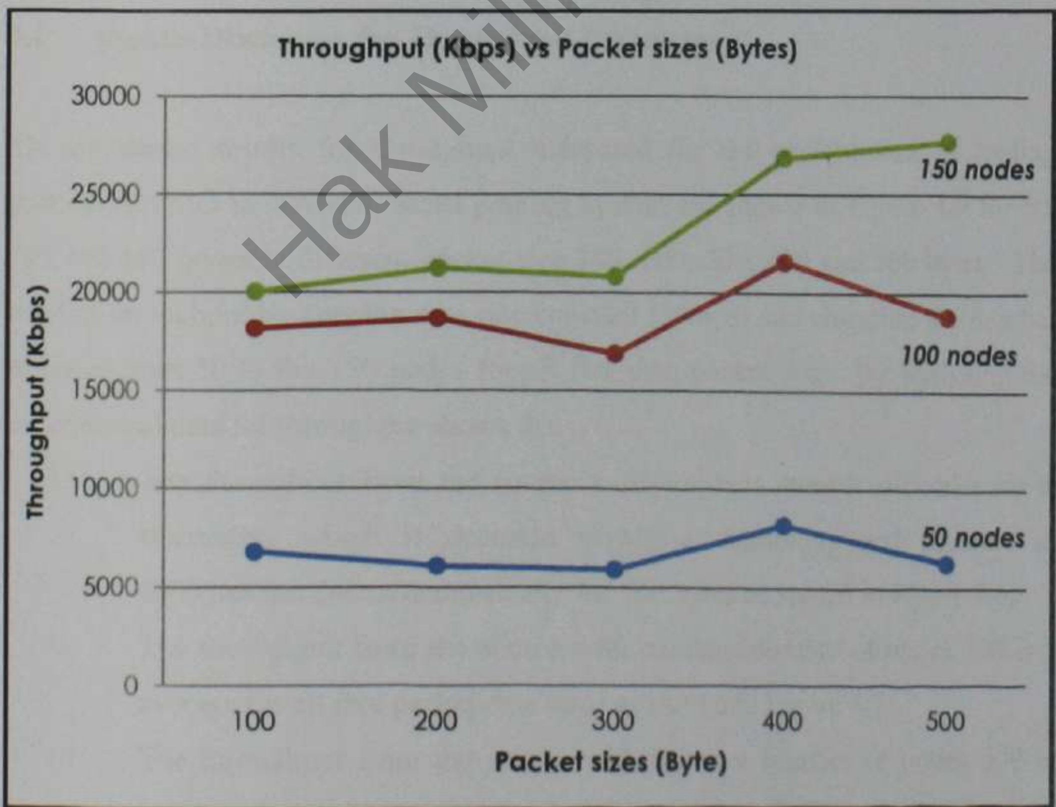

Figure 4.9: Throughput for different nodes

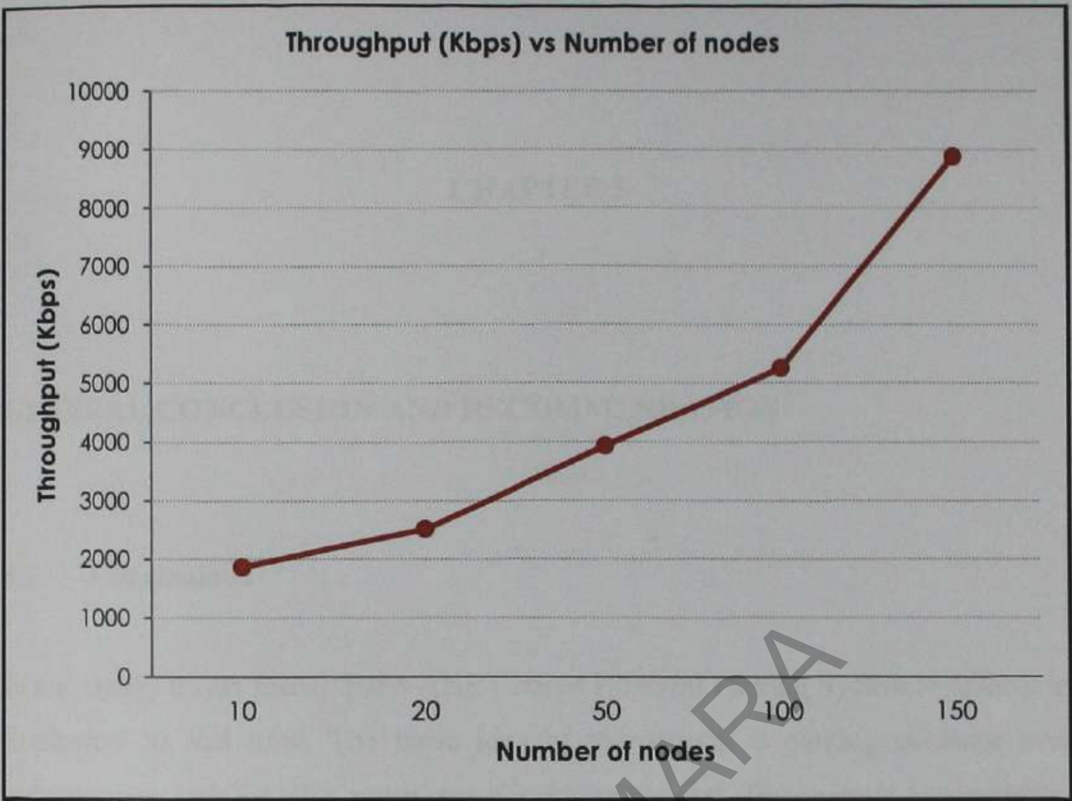

Figure 4.10: Throughput for different nodes

4.4 Results Discussion for Throughput Parameter

The simulation results for throughput measured for the performance of routing protocol, DYMO in VANET based parking system are shown in Figure 4.9 for 50, 100, and 150 nodes at different packet size 100, 200, 300, 400 and 500 bytes. The trend of throughput by keeping data rate constant (10mps) and changing the number of nodes from 50 to the 150 nodes for all five data packet sizes. By analyzing the experimental data for throughput shows that;

I. The throughput from the source with constant number of nodes 50 is fluctuating which is decrease slowly at beginning and increase at 400bytes but decrease drastically for 500 bytes as shown in Figure 4.6.

II. The throughput from the source with constant number of nodes 100 is I average for all five packet data sizes as shown in Figure 4.7

III. The throughput from the source with constant number of nodes 500 is increase upon of increase of packet size as shown in Figure 4.8

IV. The throughput is increasing by increasing of number of nodes as shown in Figure 4.10.

CHAPTER 5

GENERAL CONCLUSION AND RECOMMENDATION

5.1 Conclusions

In this study, it was found that VANET based Efficient Parking System is suitable to implement in real time. The basic idea of this system is parking guidance over infrastructure-less VANET mean that it's saving the cost. The analysis on simulation data illustrates the performance of routing protocol DYMO in term of packet delivery average is more than 83% even in high density nodes with different topologies. The process of delivery message between nodes is good for this protocol. Finally, a routing protocol DYMO is most suitable to use for this parking design because of high efficiency in term of packet delivery even in large network (high density nodes) and reduce routing overhead.

5.3 Recommended Research

In evaluating performance of routing protocols, selecting appropriate parameters is supreme. In this study only two parameters are taken; packet delivery ratio and throughput. Therefore, the authors suggest more parameter such as average end-to-end delay and routing overhead (RO) can be investigated and evaluated for its effectiveness.

13. A. Rahman and F. Anwar, "A Simulation Based Performance Comparison of Routing Protocol on Mobile Ad-hoc Network (Proactive , Reactive and Hybrid)," no. May, pp. 11–13, 2010.

14. G. Jayakumar and G. Gopinath, "Performance Comparison of Two On-demand Routing Protocols for Ad-hoc Networks based on Random Way Point Mobility Model," vol. 2, no. 2, pp. 3–16, 2008.

15. P. Nand and S. C. Sharma, "Performance Study of Broadcast based Mobile Adhoc Routing Protocols AODV , DSR and DYMO," vol. 5, no. 1, pp. 53–64, 2011.

16. F. Maan and N. Mazhar, "MANET Routing Protocols vs Mobility Models : A Performance Evaluation," 2011.

17. M. Tolani, "Effect of Packet Size on Various MANET Routing Protocols," vol. 4, no. 9, pp. 10–13, 2012.

18. K. Rajasekar, C. Kamalanathan, and C. Engineering, "Various Packet Size on Different MANET Routing Protocols," vol. 4, pp. 528–532, 2013.

19. J. V Dhobale, N. V Kalyankar, and S. D. Khamitkar, "Computer Network Performance Evaluation Based on Different Data Packet Size Using OMNeT ++ Simulation Environment," vol. 2, no. 2, 2014.

20. M. Kumar and A. K. Jain, "Effect on Throughput-Delay With Various Routing Protocols in IEEE 802 . 11Wireless Ad-Hoc Networks," *Ijcst*, vol. 4, no. 2, pp. 642–646, 2013.

21. M. Báguena, S. M. Tornell, Á. Torres, C. T. Calafate, J. Cano, P. Manzoni, and C. D. V. S. N, "VACaMobil : VANET Car Mobility Manager for OMNeT ++," no. Scpa, pp. 1057–1061, 2013.

22. H. Bakht, "Survey of Routing Protocols for Mobile Ad-hoc Network," *Int. J. Inf. Commun. Technol. Res.*, vol. 1, no. 6, pp. 258–270, 2011.

23. A. Benslimane, S. Barghi, and C. Assi, "An efficient routing protocol for connecting vehicular networks to the Internet," *Pervasive Mob. Comput.*, vol. 7, no. 1, pp. 98–113, 2011.

24. M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," *Proc. 3rd Int. Work. Veh. ad hoc networks*, pp. 30–39, 2006.

25. J. V Dhobale, N. V Kalyankar, and S. D. Khamitkar, "Computer Network Performance Evaluation Based on Different Data Packet Size Using OMNeT ++ Simulation Environment," vol. 2, no. 2, 2014.

26. D. Kum, J. Park, Y. Cho, and B. Cheon, "Performance Evaluation of AODV and DYMO Routing Protocols in MANET," pp. 1–2, 2010.

27. D. Singh, A. K. Maurya, and A. K. Sarje, "Comparative Performance Analysis of LANMAR , LAR1 , DYMO and ZRP Routing Protocols in MANET using Random Waypoint Mobility Model," pp. 62–66.

28. M. Tolani, "Effect of Packet Size on Various MANET Routing Protocols," vol. 4, no. 9, pp. 10–13, 2012.

APPENDIX A

A. Programming Script .INI file

```
# Configure the simulation in random generator.

[General]
network = inet.examples.adhocDymo.Adhocnet    # file path
num-rngs = 1                                  # multiple random number generators
repeat = 2                                    # default repetition for random generators
seed-set = ${runnumber}                       # default, every run has a different seed
sim-time-limit = 50s                          # simulation time is set to 50s
**.vector-recording = false

# mobility configuration

**.n_hosts = 150                                        # number of participating nodes.
**.host[*].mobility.constraintAreaMinZ = 0m             # initial length of area for Z - axis.
**.host[*].mobility.constraintAreaMaxZ = 0m             # end length of area for Z - axis.
**.host[*].mobility.initialZ = 0m                       # initial position of all nodes for Z - axis

#mobility for host[0]

**.host[0].mobilityType = "StationaryMobility"          # host [0] set to stationary mobility node.
**.host[0].mobility.initFromDisplayString = false       # mobility string for host [0] is not display.
**.host[0].mobility.leaveMovementTrail = true           # movement trail for host [0] is display.
**.host[0].mobility.initialX = 126m                     # location of host [0] is set to 126m from X- axis.
**.host[0].mobility.initialY = 90m                      # location of host [0] is set to 90m from Y- axis.
**.host[0].mobility.constraintAreaMinX = 0m             # Initial length of area for X - axis.
**.host[0].mobility.constraintAreaMinY = 0m             # Initial length of area for Y - axis.
**.host[0].mobility.constraintAreaMaxX = 275m           # End length of area for X - axis.
**.host[0].mobility.constraintAreaMaxY = 200m           # End length of area for Y - axis.
```

```
# mobility for host [1-120]

**.host[1..120].mobilityType = "StationaryMobility"                  # host 1 to 120 are set to stationary mobility.
**.host[1..120].mobility.initFromDisplayString = false               # mobility string for host 1 to 120 is not display.
**.host[1..120].mobility.leaveMovementTrail = true                   # movement line for host 1 to 120 is display.
**.host[1..120].mobility.constraintAreaMinX = 0m                     # Initial length of area for X - axis.
**.host[1..120].mobility.constraintAreaMinY = 0m                     # Initial length of area for Y - axis.
**.host[1..120].mobility.constraintAreaMaxX = 275m                   # End length of area for X - axis.
**.host[1..120].mobility.constraintAreaMaxY = 200m                   # End length of area for Y - axis.

**.host[1..10].mobility.initialX = uniform (30m, 250m)               # host 1-10 located uniformly from 30m to 250m for X axis.
**.host[1..10].mobility.initialY = 70m                               # host 1-10 located at 70m for Y axis.
**.host[11..20].mobility.initialX = uniform (30m, 220m)              # host 11-20 located uniformly from 30m to 250m for X axis.
**.host[11..20].mobility.initialY = 95m                              # host 11-20 located at 95m for Y axis.
**.host[21..30].mobility.initialX = uniform (30m, 250m)              # host 21-30 located uniformly from 30m to 250m for X axis.
**.host[21..30].mobility.initialY = 102m                             # host 21-30 located at 102m for Y axis.
**.host[31..40].mobility.initialX = uniform (30m, 200m)              # host 31-40 located uniformly from 30m to 200m for X axis.
**.host[31..40].mobility.initialY = 127m                             # host 31-40 located at 127m for Y axis.
**.host[41..50].mobility.initialX = uniform (30m, 200m)              # host 41-50 located uniformly from 30m to 200m for X axis.
**.host[41..50].mobility.initialY = 134m                             # host 41-50 located at 134m for Y axis.
**.host[51..60].mobility.initialX = uniform (30m, 200m)              # host 51-60 located uniformly from 30m to 200m for X axis.
**.host[51..60].mobility.initialY = 159m                             # host 51-60 located at 159m for Y axis.
**.host[61..70].mobility.initialX = uniform (30m, 200m)              # host 61-70 located uniformly from 30m to 200m for X axis.
**.host[61..70].mobility.initialY = 166m                             # host 61-70 located at 166m for Y axis.
**.host[71..80].mobility.initialX = uniform (30m, 200m)              # host 71-80 located uniformly from 30m to 200m for X axis.
**.host[71..80].mobility.initialY = 191m                             # host 71-80 located at 191m for Y axis.
**.host[81..110].mobility.initialX = 267m                            # host 81-110 located at 267m for X axis.
**.host[81..110].mobility.initialY = uniform (20m, 145m)             # host 81-110 located uniformly from 20m to 145m for Y axis.
**.host[111..120].mobility.initialX = 12m                            # host 111-120 located at 12m for X axis.
**.host[111..120].mobility.initialY = uniform (95m, 180m)            # host 111-120 located uniformly from 95m to 180m for Y axis.

# linear horizontal mobility for host[121..130]

**.host[121..130].mobilityType = "LinearMobility"                    # host 121 to 130 are set to linear mobility.
**.host[121..130].mobility.initFromDisplayString = false             # mobility string for host 121 to 130 is not display.
```

```
**.host[121..130].mobility.leaveMovementTrail = true          # movement trail for host 121 to 130 is display.
**.host[121..130].mobility.constraintAreaMinX = 0m            # Initial length of area for X - axis.
**.host[121..130].mobility.constraintAreaMinY = 0m            # Initial length of area for Y - axis.
**.host[121..130].mobility.constraintAreaMaxX = 250m          # End length of area for X - axis.
**.host[121..130].mobility.constraintAreaMaxY = 200m          # End length of area for Y - axis.
**.host[121..130].mobility.initialX = uniform(100m, 220m)     # host 121-130 located in range 100m-220m for X axis.
**.host[121..130].mobility.initialY = 82m                     # host 121-130 located at 82m for Y axis.
**.host[121..130].mobility.speed = uniform(15mps, 22mps)      # mobility speed for host 121-130 in range 15-22mps.
**.host[121..130].mobility.angle = 180deg                     # host 121-130 are in horizontal mobility or 180 degree.

# linear horizontal mobility for host[131..140]

**.host[131..140].mobilityType = "LinearMobility"             # host 131 to 140 are set to linear mobility.
**.host[131..140].mobility.initFromDisplayString = false      # mobility string for host 131 to 140 is not display.
**.host[131..140].mobility.leaveMovementTrail = true          # movement line for host 131 to 140 is display.
**.host[131..140].mobility.constraintAreaMinX = 0m            # Initial length of area for X - axis.
**.host[131..140].mobility.constraintAreaMinY = 0m            # Initial length of area for Y - axis.
**.host[131..140].mobility.constraintAreaMaxX = 275m          # End length of area for X - axis.
**.host[131..140].mobility.constraintAreaMaxY = 200m          # End length of area for Y - axis.
**.host[131..140].mobility.initialX = uniform(50m, 220m)      # host 131-140 located in range 100m-220m for X axis.
**.host[131..140].mobility.initialY = 178m                    # host 131-140 located at 178m for Y axis.
**.host[131..140].mobility.speed = uniform(15mps, 22mps)      # mobility speed for host 131-140 in range 15-22mps.
**.host[131..140].mobility.angle = 180deg                     # host 131-140 are in horizontal mobility or 180 degree.

#linear vertical mobility for host [141..150]

**.host[141..150].mobilityType = "LinearMobility"             # host 141 to 150 are set to linear mobility.
**.host[141..150].mobility.initFromDisplayString = false      # mobility string for host 141 to 150 is not display.
**.host[141..150].mobility.leaveMovementTrail = true          # movement line for host 141 to 150 is display.
**.host[141..150].mobility.constraintAreaMinX = 0m            # Initial length of area for X - axis.
**.host[141..150].mobility.constraintAreaMinY = 0m            # Initial length of area for Y - axis.
**.host[141..150].mobility.constraintAreaMaxX = 275m          # End point length of area for X - axis.
**.host[141..150].mobility.constraintAreaMaxY = 200m          # End point length of area for Y - axis.
**.host[141..150].mobility.initialX = 250m                    # host 141-150 located at 250m for X axis.
**.host[141..150].mobility.initialY = uniform(50m, 170m)      # host 141-150 located in range 50m-170m for Y axis.
```

```
**.host[141..150].mobility.speed = uniform(15mps, 22mps)   # mobility speed for host 141-150 in range 15-22mps.
**.host[141..150].mobility.angle = 270deg                  # host 141-150 are in vertical mobility or 270 degree.


# Configure the hosts to have a single "ideal" wireless NIC. An IdealWirelesNic
# can be configured with a maximum communication range. All packets within range
# are always received successfully while out of range messages are never received.
# This is useful if we are not interested how the actual messages get to their destination.
# We just want to be sure that they get there once the nodes are in range.

**.wlan[*].typename = "IdealWirelessNic"
**.wlan[*].bitrate = 10Mbps                                # ALL radios and MACs should run on 10Mbps.
**.wlan[*].mac.address = "auto"                            # Auto generation for MAC address.
**.wlan[*].mac.headerLength = 20B                          # packet size is limited to 20 byte.
**.wlan[*].radioType = "IdealRadio"                        # type of radio network is set.
**.wlan[*].radio.transmitter.headerBitLength = 0b          # no bit length is set.
**.wlan[*].radio.transmitter.maxCommunicationRange = 100m  # mandatory physical layer parameters
**.wlan[*].radio.transmitter.maxInterferenceRange = 0m     # no interference range for each router.
**.wlan[*].radio.transmitter.maxDetectionRange = 0m        # no interference detection for each router.
**.wlan[*].radio.receiver.ignoreInterference = true        # router interference is not set.

# Configure an app that receives the UDP traffic (and simply drops the data)

**.host[0].udpApp[*].typename = "UDPSink"        # host 0 consume and print packets received from the UDP module
**.host[0].numUdpApps = 1                        # application module is set to connect to UDP module.
**.host[0].udpApp[0].localPort = 1234            # local port for UDP module for host 0 is set.

# Configure an application for all host that sends a constant UDP traffic.

**.host[*].numUdpApps = 1
**.host[*].udpApp[*].typename = "UDPBasicBurst"  # all nodes send UDP packets to the given IP address in burst
**.udpApp[0].destAddresses = "host[0]"
**.udpApp[0].localPort = 1234                    # local port for UDP module for host 0 is set.
**.udpApp[0].destPort = 1234                     # destination port for UDP module for host 0 is set.
**.udpApp[0].messageLength = 512B                # UDP packet is can set to any size.
```

```
**.udpApp[0].sendInterval = 1s                    # time interval for each UDP packet send to host 0.
**.udpApp[0].burstDuration = 0
**.udpApp[0].chooseDestAddrMode = "perBurst"
**.udpApp[0].sleepDuration = 1s
**.udpApp[0].startTime = 10s
**.udpApp[0].delayLimit = 20s                     # destination address for random number generator is not set.
**.udpApp[0].destAddrRNG = 0
```

APPENDIX B

B. Programming Script .NED file

```
package inet.examples.adhocDymo;

import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.node.dymo.DYMORouter;
import inet.physicallayer.idealradio.IdealRadioMedium;


//
// TODO auto-generated type
//
network Adhocnet
{
    parameters:
        int n_hosts;
        @display("i=old/proc2;bgi=background/parking,s;bgb=275,200,black,blue,2;bgs=2,m");
    submodules:
        host[n_hosts]: DYMORouter {
            @display("p=109,91;i=device/palm;is=vs");
        }
        radioMedium: IdealRadioMedium {
            @display("p=114,141");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=52,65");
        }
}
```

VITA

The author was born in October 25, 1978, in Terengganu. He went to Sekolah Menengah Nasiruddin Shah, Besut and Sekolah Menengah Sains Dungun, Terengganu, Malaysia for his secondary school. He pursued his degree at the University of Science, Malaysia, and graduated with the B. Eng. (Hons) in Electrical and Electronic Engineering in 2001. Upon graduation, he worked as a Vocational Training Officer in Electronic Department at Institut Kemahiran MARA Beseri, Perlis. He then transferred to MARA-Japan Industrial Institute (MJII), Selangor in Instrumentation and Control Department in 1998. He then transferred to Kolej Kemahiran Tinggi MARA Ledang, Johor in Biomedical Instrumentation Department in 2010. He then transferred to Kolej Kemahiran Tinggi MARA Kemaman, Terengganu as Head of Program in Instrumentation and Control Department in 2013. He is also an active amateur radio operator holding callsigns 9W2RDD with much interest in satellite, moonbounce, and packet communications.

```
# mobility for host [1-120]

**.host[1..120].mobilityType = "StationaryMobility"                 # host 1 to 120 are set to stationary mobility.
**.host[1..120].mobility.initFromDisplayString = false               # mobility string for host 1 to 120 is not display.
**.host[1..120].mobility.leaveMovementTrail = true                   # movement line for host 1 to 120 is display.
**.host[1..120].mobility.constraintAreaMinX = 0m                     # Initial length of area for X - axis.
**.host[1..120].mobility.constraintAreaMinY = 0m                     # Initial length of area for Y - axis.
**.host[1..120].mobility.constraintAreaMaxX = 275m                   # End length of area for X - axis.
**.host[1..120].mobility.constraintAreaMaxY = 200m                   # End length of area for Y - axis.

**.host[1..10].mobility.initialX = uniform (30m, 250m)               # host 1-10 located uniformly from 30m to 250m for X axis.
**.host[1..10].mobility.initialY = 70m                               # host 1-10 located at 70m for Y axis.
**.host[11..20].mobility.initialX = uniform (30m, 220m)              # host 11-20 located uniformly from 30m to 250m for X axis.
**.host[11..20].mobility.initialY = 95m                              # host 11-20 located at 95m for Y axis.
**.host[21..30].mobility.initialX = uniform (30m, 250m)              # host 21-30 located uniformly from 30m to 250m for X axis.
**.host[21..30].mobility.initialY = 102m                             # host 21-30 located at 102m for Y axis.
**.host[31..40].mobility.initialX = uniform (30m, 200m)              # host 31-40 located uniformly from 30m to 200m for X axis.
**.host[31..40].mobility.initialY = 127m                             # host 31-40 located at 127m for Y axis.
**.host[41..50].mobility.initialX = uniform (30m, 200m)              # host 41-50 located uniformly from 30m to 200m for X axis.
**.host[41..50].mobility.initialY = 134m                             # host 41-50 located at 134m for Y axis.
**.host[51..60].mobility.initialX = uniform (30m, 200m)              # host 51-60 located uniformly from 30m to 200m for X axis.
**.host[51..60].mobility.initialY = 159m                             # host 51-60 located at 159m for Y axis.
**.host[61..70].mobility.initialX = uniform (30m, 200m)              # host 61-70 located uniformly from 30m to 200m for X axis.
**.host[61..70].mobility.initialY = 166m                             # host 61-70 located at 166m for Y axis.
**.host[71..80].mobility.initialX = uniform (30m, 200m)              # host 71-80 located uniformly from 30m to 200m for X axis.
**.host[71..80].mobility.initialY = 191m                             # host 71-80 located at 191m for Y axis.
**.host[81..110].mobility.initialX = 267m                            # host 81-110 located at 267m for X axis.
**.host[81..110].mobility.initialY = uniform (20m, 145m)             # host 81-110 located uniformly from 20m to 145m for Y axis.
**.host[111..120].mobility.initialX = 12m                            # host 111-120 located at 12m for X axis.
**.host[111..120].mobility.initialY = uniform (95m, 180m)            # host 111-120 located uniformly from 95m to 180m for Y axis.

# linear horizontal mobility for host[121..130]

**.host[121..130].mobilityType = "LinearMobility"                    # host 121 to 130 are set to linear mobility.
**.host[121..130].mobility.initFromDisplayString = false             # mobility string for host 121 to 130 is not display.
```

```
**.host[141..150].mobility.speed = uniform(15mps, 22mps)   # mobility speed for host 141-150 in range 15-22mps.
**.host[141..150].mobility.angle = 270deg                  # host 141-150 are in vertical mobility or 270 degree.


# Configure the hosts to have a single “ideal” wireless NIC. An IdealWirelesNic
# can be configured with a maximum communication range. All packets within range
# are always received successfully while out of range messages are never received.
# This is useful if we are not interested how the actual messages get to their destination.
# We just want to be sure that they get there once the nodes are in range.

**.wlan[*].typename = "IdealWirelessNic"                       # ALL radios and MACs should run on 10Mbps.
**.wlan[*].bitrate = 10Mbps                                    # Auto generation for MAC address.
**.wlan[*].mac.address = "auto"                                # packet size is limited to 20 byte.
**.wlan[*].mac.headerLength = 20B                              # type of radio network is set.
**.wlan[*].radioType = "IdealRadio"                            # no bit length is set.
**.wlan[*].radio.transmitter.headerBitLength = 0b              # mandatory physical layer parameters
**.wlan[*].radio.transmitter.maxCommunicationRange = 100m      # no interference range for each router.
**.wlan[*].radio.transmitter.maxInterferenceRange = 0m         # no interference detection for each router.
**.wlan[*].radio.transmitter.maxDetectionRange = 0m            # router interference is not set.
**.wlan[*].radio.receiver.ignoreInterference = true

# Configure an app that receives the UDP traffic (and simply drops the data)

**.host[0].udpApp[*].typename = "UDPSink"          # host 0 consume and print packets received from the UDP module
**.host[0].numUdpApps = 1                          # application module is set to connect to UDP module.
**.host[0].udpApp[0].localPort = 1234              # local port for UDP module for host 0 is set.

# Configure an application for all host that sends a constant UDP traffic.

**.host[*].numUdpApps = 1                          # all nodes send UDP packets to the given IP address in burst
**.host[*].udpApp[*].typename = "UDPBasicBurst"
**.udpApp[0].destAddresses = "host[0]"
**.udpApp[0].localPort = 1234                      # local port for UDP module for host 0 is set.
**.udpApp[0].destPort = 1234                       # destination port for UDP module for host 0 is set.
**.udpApp[0].messageLength = 512B                  # UDP packet is can set to any size.
```

```
**.udpApp[0].sendInterval = 1s                    # time interval for each UDP packet send to host 0.
**.udpApp[0].burstDuration = 0
**.udpApp[0].chooseDestAddrMode = "perBurst"
**.udpApp[0].sleepDuration = 1s
**.udpApp[0].startTime = 10s
**.udpApp[0].delayLimit = 20s                     # destination address for random number generator is not set.
**.udpApp[0].destAddrRNG = 0
```

B. Programming Script .NED file

```
package inet.examples.adhocDymo;

import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.node.dymo.DYMORouter;
import inet.physicallayer.idealradio.IdealRadioMedium;


//
// TODO auto-generated type
//
network Adhocnet
{
    parameters:
        int n_hosts;
        @display("i=old/proc2;bgi=background/parking,s;bgb=275,200,black,blue,2;bgs=2,m");
    submodules:
        host[n_hosts]: DYMORouter {
            @display("p=109,91;i=device/palm;is=vs");
        }
        radioMedium: IdealRadioMedium {
            @display("p=114,141");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=52,65");
        }
}
```

VITA

The author was born in October 25, 1978, in Terengganu. He went to Sekolah Menengah Nasiruddin Shah, Besut and Sekolah Menengah Sains Dungun, Terengganu, Malaysia for his secondary school. He pursued his degree at the University of Science, Malaysia, and graduated with the B. Eng. (Hons) in Electrical and Electronic Engineering in 2001. Upon graduation, he worked as a Vocational Training Officer in Electronic Department at Institut Kemahiran MARA Beseri, Perlis. He then transferred to MARA-Japan Industrial Institute (MJII), Selangor in Instrumentation and Control Department in 1998. He then transferred to Kolej Kemahiran Tinggi MARA Ledang, Johor in Biomedical Instrumentation Department in 2010. He then transferred to Kolej Kemahiran Tinggi MARA Kemaman, Terengganu as Head of Program in Instrumentation and Control Department in 2013. He is also an active amateur radio operator holding callsigns 9W2RDD with much interest in satellite, moonbounce, and packet communications.