Deanship of Graduate Studies

Al-Quds University

A Comparison Study on the Performance of Different Applications using MANET Routing Protocols under Various Circumstances

Shatha Dawod Nijem

M.Sc. Thesis

Jerusalem – Palestine

1436 / 2015

Deanship of Graduate Studies

Al-Quds University

A Comparison Study on the Performance of Different Applications using MANET Routing Protocols under Various Circumstances

Prepared By:

Shatha Dawod Nijem

Supervisor:

Dr. Nidal Kafri

M.Sc. Thesis

Thesis Submitted in Partial fulfillment of requirements for the Master Degree of Computer Science from Computer Science Department of Al-Quds University

Jerusalem – Palestine

1436 / 2015

Deanship of Graduate Studies

Al-Quds University

Computer Science Department

Thesis Approval

A Comparison Study on the Performance of Different Applications using MANET Routing Protocols under Various Circumstances

Prepared By: Shatha Dawod Nijem
Registration No: 20714054

Supervisor: Dr. Nidal Kafri

Master thesis submitted and accepted. Date:

The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Nidal Kafri Signature: ……………..

2- Internal Examiner: Dr. Raed Al-Zagal Signature: ……………..

3- External Examiner: Dr. Ahmad Sa'deh Signature: ……………..

Dedication

To my parents, for their love and support throughout my life and for their encouragement to chase my dreams.

To my brothers and sisters, for their continues help.

To my dear husband, for his endless love, remarkable patience and unlimited understanding in many moments.

To my wonderful children who deserve wholehearted thanks and love .

Shatha Dawod Nijem

Declaration

I certify that this thesis submitted for the degree of Master is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed

Shatha Dawod Nijem

Date: 15/ 03/ 2015

Acknowledgements:

I would like to thank first my supervisor Dr. Nidal Kafri for his support and advice during the course of my thesis.

I would also like to thank all teachers who have taught me during my study period.

I would also like to thank Dr Safa Nasser Al- Dien Ministry of

telecommunication for her encouragement.

I would also like to thank all friends and colleagues of Alquds University

To all postgraduate students at Al-Quds University

Finally, I wish to convey my sincere appreciation and thanks to my parents, my dear husband and children for their love and their continuous support to reach my goals.

Shatha Dawod Nijem

Abstract:

Mobile adhoc network is a collection of mobile devices that communicate amongst each other using message passing to collaborate in a wireless medium, without any centralized management; each device acts as a router, sends and receive packets. Nodes can move freely and can set itself in any adhoc network. Adhoc networks are widely use in the absence of the wired network infrastructure.

Quality of service of routing in ad hoc networks is an important and complicated issue with a changing topology.

In this work we carried out a comparison study in a simulation scenarios on the performance of different routing protocols i.e., proactive and reactive, with different standard applications such as FTP, HTTP and database under various circumstances by means of network size, load, and speed of nodes.

As a conclusion of this study, results show when measuring performance of delay and throughput of FTP, HTTP and Database traffic, delay and throughput metrics, using AODV, DSR, OLSR routing protocols, under 10, 50 and 100 nodes with speed of 10, 30 m/s. When using DSR routing protocol it showed the worst results under various network size and speed between other protocols, while when using AODV routing protocol it performed in a better way in which it showed a good performance in small and medium network size. OLSR routing protocol performed the best to be used in all network size especially in large network size.

الملخص:

مانيت هي عبارة عن مجموعة من الاجهزة تحتوي على أجهزة لاسلكية، تتعاون مع بعضها البعض لارسال، استقبال و نقل المعلومات من خلال وسط لاسلكي، هوائي، من دون اي تحكم مركزي، كل جهاز يمكن اعتبارة كجهاز راوتر يعتمد على نفسة بحيث يستقبل و يرسل المعلومات و هذة المجموعة تسمى بالمانيت. أي جهاز بهذة المجموعة يمكنة التحرك بحرية و يمكنة الانضمام لاي مجموعة مانيت بالوقت الذي يريدة و بالمكان الذي يريدة.

تستخدم المانيت بالأمكان الذي لايوجد فيها بنية تحتية ثابتة، بالأمكان التي يحصل فيها كوارث طبيعية و الأمكان الذي يصعب فيها وجود اتصال بالوقت الذي يكون هو ضروري لوجود شبكة بين الأجهزة.

تستخدم هذة الأجهزة بروتوكلات لنقل المعلومات مثل AODV DSR و غيرها من البروتوكلات، حيث ان المانيت شبكة سريعة التغيير فان جودة هذة البروتوكلات هو أمر مهم.

في هذة الرسالة قمنا بدراسة التطبيقات،FTP HTTP database ، لشبكة المانيت باستخدام البروتوكلات AODV ، DSR ، OLSR كلا على حدى، تحت متغيرات و ظروف معينة لمعرفة اي بروتوكل هو الانسب لاستخدامة مع اي تطبيق تحت هذة الظروف.

و كنتيجة لهذة الدراسة، عند قياس أداء FTP، Database، HTTP مقاييس الإنتاجية، وذلك باستخدام AODV، DSR، OLSR بروتوكولات التوجية ، تحت 10،50 و 100 عقدة وتصل سرعتها من 10 ، 30 م / ث.تبين انة عند استخدام بروتوكول توجية DSR أظهرت أسوأ نتائج في جميع الإطارت من حيت حجم الشبكة والسرعة بين البروتوكولات الأخرى، بينما عند استخدام بروتوكول التوجية AODV تنفذ ذلك بطريقة أفضل بحيت أظهرت أداء جيدا في حجم الشبكة الصغير والمتوسطة. اما بنسبة لبروتوكول التوجية OLSR اظهر أداء أفضل لاستخدامها في جميع احجام الشبكة وخصوصا في حجم شبكة الواسعة

Table of Contents

List of tables

List of figures

Chapter 1

1.1 Introduction

The last decade has seen a huge growth in computer technologies in an unpredictable and rapid way especially in wireless fields. Now days, a user can check his email; get onto the internet using his mobile phone, while on the other hand devices such as laptops, tablet PCs, smart phones, and devices containing wireless technology are growing more and more. In the meantime, these devices are getting smaller in size and cheaper in cost, which makes them available to huge numbers of users, and since each device contains wireless capabilities it will be able to connect to a wireless network and take advantage of the internet and many other services. This revolution in technology has made MANET a big challenge to be studied and enhanced, especially in the field of its performance.

Mobile Adhoc Network (MANET) is a wireless communication network; for the past 25 years, it has been of interest to many researchers analyzing and developing its features [10].

Adhoc is a latin phrase that means, for this purpose. MANET consists of number of devices that demand to communicate with each other without cables and without any pre -fixed infrastructure those devices are called wireless nodes [10]. These can be laptops, mobiles, phones, even vehicles: they can be any device that contains a wireless capability, either mobile or fixed (i.e. an airplane or a ship that has a wireless radio can be a node in a MANET network just as a personal computer that is fixed can also be a node in a MANET network).

Every wireless node must be able to communicate, connect on its own in a dynamic way and transfer data between other wireless nodes. Nodes use radio channels and all nodes must be within one another's range.

MANET is formed usually in urgent or emergency situations, such as natural disasters or military needs in war, where a fixed infrastructure is difficult to form [8].

It doesn’t depend on infrastructure, wires or cables it only needs air to be established, which lowers costs for deployment. On the other hand, no obstacles to establish the network anywhere and at any time, independent, free communication. [2] Nodes can leave or enter the networks, where ever their geographic position is, and whenever they want, which changes its topology dynamically, and that’s why MANET is unique.

In MANET networks, nodes cooperate with each other, in which a node can be the source: the one that demands to send the data to the destination node. On the other hand it can be in another situation the destination, or it can be an intermediate node that only helps to find the destination by receiving and forwarding data to the nearest neighbor, or to the destination which can be considered a router.

The act of moving information from a source to a destination through intermediate nodes is called routing [8]. Thus, routing is necessary whenever a packet needs to be sent to the destination by the temporally wireless network, adhoc

Each node in a wireless network moves according to a model called mobility model.

MANETs have limitations, especially in resources, in which it may have limited transmission range, limited memory and limited battery life.

1.2 Problem statement

The needs of forming a network anytime and anywhere has made MANET more valuable to be studied in different ways, in order to create more reliable and more usable MANETs.

The evaluation of different routing protocols became a priority issue in performance subjects, due to MANET's changing topology.

Varying network sizes, traffic, load types and speeds may affect the performance of different routing protocols, however, the important point that we should concentrate on are the applications that is used such as FTP, HTTP and database with high and low loads, which might affect MANET reliability[3].

Much research has been done to analyze and evaluate MANET routing protocols using one type of traffic, constant bit rate (CBR) traffic, file transfer protocol (FTP) traffic, etc., but few studies have concentrated on more than one types of traffic.

1.3 Related work

Many researchers have analyzed MANET routing protocols using different kind of simulators, and different kind of data traffic. Since 2000, till now much research has being done on MANET protocols.

[34] Analyzed the quality of service of MANET protocols such as AODV, OLSR, TORA have been conducted through measuring network congestion, with AODV showing lowest result between the other protocols, 2011 has measured packet delivery ratio, end to end delay, routing overhead and throughput for reactive and proactive protocols, DSR, DSDV and AODV for CBR traffic using different numbers of nodes using NS2.Results show that reactive protocol has a better performance than proactive protocol.

In [36] 2010 has studied the performance of reactive, proactive and hybrid under realistic network scenarios. AODV, DSR, OLSR, ZPR these scenarios were made on a real live network mobility of nodes simulated using GPS. Traffic was created by a generation tool, using 19 mobile nodes and a base station for 4 hours. Another simulation was carried out by a QUALNET simulator, throughput results shows AODV performed the best among protocols, delay shows AODV has the lowest results between protocols, and each live simulation and the QUALNET simulator gave exactly the same results [35].

In [26] 2009 S. Dhulipala, RM. Chandrasekaran and R. Prabakaran measured temporal analyses, the scalability of various types of applications, CBR, FTP, TELNET, and VBR using a different number of nodes (10, 120, 250, 275, 375, 475 and 575 nodes), on AODV protocol. The QUALNET simulator results shows the execution time for CBR was the highest, while VBR was the lowest, when compared with different type of applications. Other applications showed consistent results. One year later [27], V. Tafti, A. Gandomi used OPNET to measure quality of service of AODV, DSR and OLSR protocols using different metrics, delay, throughput, and packet drop rate using FTP traffic. Throughput results showed that OLSR protocol performed in perfectly when compared with other protocols in large network scales. On the other hand, AODV performed better than DSR in small network scale, and delay results shows that DSR has scored the worst results between protocols.

In [28]. 2011 four MANET protocols were measured over different traffic types: FTP, HTTP, with various network sizes and at different speed. Results showed that OLSR has performed in a perfect way in all cases, while AODV performed well in a medium network size, and DSR can be used only in small network size

One year later in [30]. 2102, S. Parulpreet, B. Ekta, and W. Gurleen measured traffic load (HTTP, Email, Video conference) on DSR routing protocols using 40 node high load and a speed of 10m/sec and 800*800 area space. They found that DSR was delayed when it was used with video conference, HTTP scored the lowest in delay. Throughput results were highest in video conference and lowest in HTTP

[33] S. Parulpreet, B. Ekta, and W. Gurleen also measured the delay and throughput of OLSR, AODV, and DSR using different traffic loads, FTP and HTTP with a fixed number of nodes: 40 nodes on 600×600 square meter area. Results showed DSR has the highest result when measuring the delay in traffic, and OLSR has the lowest result where in throughput AODV didn't performed bad; OLSR returned the highest results.

In [6] 2013, presented a performance analysis of five routing protocols in MANET using video conference and email applications for 30, 60 and 90 nodes. When using video conference, ADOV performed the best for lower number of nodes when comparing with other protocols, on the other hand OLSR can be used as a replacement for high number of node, while when using email application OLSR and TORA also performed equally.

At the same year [4] performed a comparison study between four protocols AODV, GRP, DSR and OLSR with traffic loads database in terms of Delay, Load, Media access delay, Network Load, Retransmission and Throughput for 20 nodes.

In [49] March 2014 a simulation study was executed to measure throughput of ADOV, DSR, OLSR using OPNET simulator under HTTP, FTP, video conference, heavy traffic, on large network size 100 node, as a conclusion for this study, they found that

DSR throughput is very less than OLSR and AODV, on the other hand OLSR throughput is higher when comparing it with other protocols.

Throughput is higher when using it with HTTP and lowest when using it with video conference and email.

[73] Another performance has been executed to study between four routing protocols AODV, GRP, DSR and OLSR with traffic loads database in terms of delay, load and media access delay, network load, retransmission and throughput for 20 nodes.

After three months, [50] June 2014, another study has been published, measured AODV and DSR performance based on throughput, delay and packet delivery, while changing speed using NS2, results showed a high results when measuring AODV when comparing it with DSR performance.

1.4 Contribution

The aim of this thesis is to analyze the performance of MANET routing protocol: the proactive and reactive under FTP, HTTP and database traffic, using different metrics delay and throughput by the OPNET simulator.

We have designed scenarios with varying number of nodes, traffic load and speed to find out which protocol is the optimal protocol, finally, create a guide (in the form of a table) which will help future researchers in choosing the best protocol to be used with specific situations, applications: low and high load, small, medium and large networks.

1.4.1 Research questions

In this study, we have tried to answer the following questions:

a) Which protocol will make mobile adhoc networks more reliable, more efficient when using FTP server?

b) Which protocol will make mobile adhoc networks more reliable and more efficient when using HTTP server?

c) Which protocol will make mobile adhoc networks more reliable, and more efficient when using Database server?

A. How do using varying numbers of nodes affect the performance of routing protocols, using the same application?

d) What factors may affect the mobile adhoc network?

To answer these questions, we have designed MANET scenarios using different parameters and evaluated the performance of each protocol: AODV, DSR and OLSR. After that we have analyzed the results to find which of the protocols is most suitable for each specific application.

1.4.2 Research methodology

We have used the following methodology to conduct the research, design and implement different scenarios to evaluate routing protocols with different application using OPNET simulator.

First: Attempt to collect and analyze existing studies which have evaluated routing protocol using different types of data traffic.

Second: Analyze routing protocol behaviors in general cases.

Third: Use the OPNET Simulator to build the required scenarios and analyze MANET routing protocols.

Forth: Study the MANET environment, applications configuration, mobility configuration, and choose the performance metrics that will be used to evaluate Adhoc network.

Fifth: Design and implement MANET scenarios using the OPNET simulator.

Sixth: Run each simulation scenario, collect results.

Seventh: Analyze the collected results and compare them with previous studies.

Finally: Find how each protocol performs in different scenarios.

By designing and implementing MANET scenarios, those questions should be solved.

1.5 Research boundaries

This thesis will help direct researchers and people who will use MANET system, to choose which routing protocol is more suitable to be used with which application, traffic load, and numbers of nodes.

1.6 Keywords and definitions

- **MANET**: Mobile Adhoc Network
- **AODV:** Adhoc On demand Distance Vector
- **DSR:** Dynamic Source Routing
- **OLSR**: Optimized Link State Routing
- **RREQ:** (Route Request) is a message that is broadcast or sent by a source node to a specific destination.
- **RREP** (Route Reply) is a message that is unicasted by destination to source.
- **FTP**: File Transfer Protocol
- **HTTP:** Hyper Text Transfer Protocol
- **Active route:** A route to the destination that is marked in the routing table as valid. Only active routes can be used to forward data packets.
- **OPNET:** Optimized Network Engineering Tool

- **Mobility model**: The way a node move in a simulation, from the minute it begins until its end.

1.7 Thesis organization

This thesis is divided into five chapters:

Chapter 1 introduces MANET, presents related work, research questions and the problem statement.

Chapter 2 describes the background of mobile adhoc network, reactive MANET protocols and proactive MANET protocols and their applications to FTP and HTTP.

Chapter 3 presents a review of literature studying MANET, from 1999 until recent time.

Chapter 4 discusses the performance metrics: delay and throughput. It also introduces the simulation tool OPNET modeler 14 simulation work space and finally the analysis and simulation results of all routing protocols and applications represented as two parts: the first simulation using nodes speed 10m/ s, and the second simulation part using nodes of a speed of 30 m/s.

Chapter 5 presents my conclusions and suggestions for future research.

Chapter 2

Background

Wireless technologies have become an important issue recently, due to many reasons. From one perspective it's utility in accessing the internet and exchanging information, another point is the relative ease with which users can create such types of networks, and the relative inexpensive cost in deploying and adding devices to such networks.

2.1 Wireless network types

Two types of wireless networks:

2.1.1 Infrastructure wireless network

Wireless networks that depend on fixed infrastructure: these types of network nodes are connected to a base station, or access points through either cables or wireless links. All data packets have to pass through the access point, which works as a centralized system to all nodes, and all communication can be done only within the access point's radio range [7], [8], [47].

2.1.2 Infrastructure less network.

These are wireless networks that don't depend on any fixed infrastructure and are created in places where no fixed infrastructure exists. They are called adhoc networks (MANET), and each node is connected in a wireless links; a node can be the source that sends data packets, or one can be the router that helps pass data onwards[7], [8], [47].

2.2 MANET overview

MANET is a rescue technology that can be easily created in critical situations, in emergency situation, where cables and wires are difficult to use or obtain.

Figure 2.1 describes MANET consist of, this figure is originally from The Handbook of Adhoc Wireless Network [12]

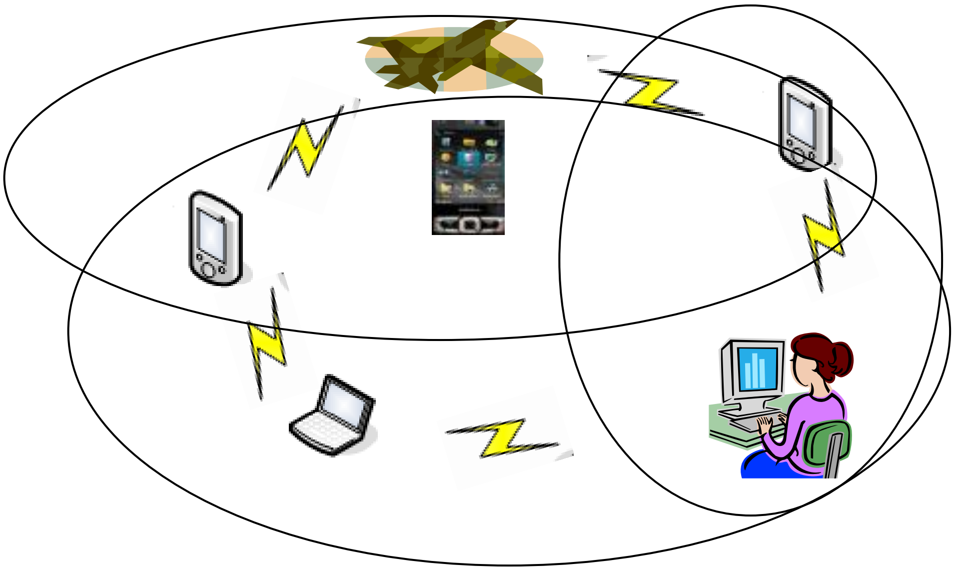


Figure 2.1 Overview of mobile adhoc network

MANET is a group of devices, such as laptops, mobile devices or any other devices that have both a transmitter and receiver with smart antennas, as shown in figure2.1. Each device is called a wireless node, and these nodes must be in each other's range. They can connect and communicate with each other using wireless links. MANET doesn't depend on any fixed infrastructure, nor a centralized administrator. Each node can act as either a router for other devices, which help pass data, or it can act as the source, which demands other devices (acting as nodes) to send the data. Additionally, each node can be the end system the destination. In all, nodes act in a complicated elegant, distributed way, as a mobile mesh network[47], as illustrated in figure 2.2 [8].

The topology of MANET changes dynamically in a predictable and rapid way, whenever a new node joins the network, or whenever an existing node leaves the network.

Nodes must work with each other; serve each other to enable effective data transfers, using routing mechanisms.

Routing is to find the best way for migrating data from the source to its destination. Routing protocols AODV, DSR and OLSR, and TORA are discussed below [21].

Energy is consumed whenever a node participates; on the hand, energy is also consumed in the network traffic, which may result in an energy limitation.

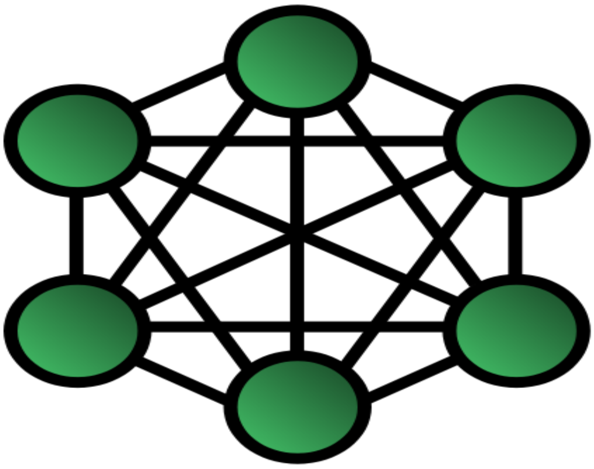


Figure 2.2 Mesh network

MANET can also be called short live networks, which can be set at anytime, anywhere, and can be created freely, in a dynamically and relatively fast way. Each node can organize itself into an arbitrary and temporary adhoc network, the life time of which can be short according to devices around, since we cannot know if the node will remain in the network the next minutes or disappear.

2.2.1 MANET: a history

The history of MANET can be divided into three stages: the first began when the idea of MANET appeared in 1970 in a concept called Packet Radio Network (PRNET) meaning store and forward radio communications. Three years later, it was represented officially by (DARPA) The Defense Advanced Research Project Agency [32]. Extensive work has since been done to develop PRNET, which features many advantages, including mobility, the ease with which networks may be deployed at anytime and anywhere without cables. PRNET consists of a number of devices: personal computers attached to an interface wire, High level data Link Control (HDLC). The second stage was in \80, when survivable adaptive radio. Networks program (SURAN) improved radio performance by making them smaller, cheaper, and safe from attacks. The third stage was in the beginning of 1990 with the birth of a new generation of computer and notebooks that contain radio waves, bluetooth technologies, and mobile adhoc networks were proposed in research conferences [32].

2.2.2 MANET characteristics

MANET networks is unique and valuable due to its characteristics, in which MANET is easily and relatively quickly created and deployed, its nodes can be fixed or wireless, they can be personal computers, devices in ships, airplanes or any other device which contains wireless radio capabilities.

MANET has no centralized management it works in a decentralized way, its nodes communicate using mobile platforms, and each can connect to another using a discovering process.

MANET is known with its dynamic topology, nodes can change their position anytime and anywhere, they can be self creating, self configuring and self managed, playing two roles; acting as a host on the other hand acting as a router, according to circumstances. When a source desires to send to a destination out of its range it exhibits multi-hop routing.

MANET has two weaknesses, its nodes relays on batteries so MANET conserve energy, its second weakness is limited security.

2.2.3 MANET applications

Because of MANET's independent nature, MANET is considered to be one of the most useful networks in fields of communication where infrastructure is difficult to establish.

MANET can be used in critical situations such as battlefields; battlefields are a complicated and critical area where communication rarely exists. Soldiers can communicate with each other using adhoc networks in this case, with which they might use mobile devices.

MANET can be useful in disaster situations, in which MANET plays a potential role, especially in places where communication has been destroyed, such as an airplane crash, earthquake, or flood, due to its flexible nature.

On the other hand it can help police officers, such as fire fighting, which arises in unpredictable places.

MANET can be used to share data between students and instructors in classrooms, by creating virtual classrooms, it can be used in campuses and in conferences on the other it can be used in business meetings**,** supporting systems, support nurses, doctors [47].

2.2.4 MANET challenges

MANET's many characteristics mentioned above lead to different challenges, including energy. Nodes usually depend on batteries, which result in energy limitations, especially because MANET has a dynamic topology in which nodes can move freely, and can use bidirectional and unidirectional. On the other hand, nodes can join the network any time and from anywhere.

Security is an important subject in all kinds of networks. MANET has the most difficult security situation due to its nature and topology. Channels are unprotected from other signals, which could be wormhole attacks or hidden nodes [47]

2.2.5 The MANET IETF working group

The Internet Engineering Task Force (IETF) has created a working group for MANET routing [51]. The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies. The fundamental design issues are that the wireless link interfaces have some unique routing interface characteristics and that node topologies within a wireless routing region may experience increased dynamics, due to motion or other factors [51].

Many protocols have been proposed, but only three protocols were accepted as experimental Request for Comments (RFC), AODV, OLSR, and Topology Dissemination Based on Reverse Path Forwarding (TBRPF) [52].

2.2.6 MANET experiments in real world

Multiple experiments were executed in real life to test MANET. In a real world experiment, all parts of the system are fully functional in a real world setting. The whole network is deployed and tested under realistic, albeit experimental conditions [18]. In 1998 [53] began to make real experiments they worked on a DSR prototype, their experiments consist of five mobile nodes installed in cars moving at a speed of 40 km/h, a mobile node was connected via mobile IP and two stationary nodes which were installed 671 m apart at opposite ends of the course traveled by the mobile nodes. Loss rate per hop reported from 11% to 5%, overall end to end loss rate is reported to be 10%. About 90% of the packets used two and three hops.

One of the largest MANETs experiments was implemented by university of Washington; its main idea was deploying a team of 100 autonomous robots for the surveying of an indoor area. The robots used proactive link state routing protocol, when the experimenters were deployed all robots at once, the network broke down. The solution was by bringing 10 to 18 at a time [9].Another experiment was executed by university of Mannheim called The Fleetnet Router to implement the forwarding strategy of the position based on routing protocol GPSR, nodes are installed in cars with windows based application PC, a linux-based 802.11b router, onboard GPS and GPRS to monitor the internal state of the node. As a conclusion they found that the maximum achievable throughput of 400 Kb/s depends on the size of the packets as smaller packets lead to more collisions [18].

2.3 MANET routing protocols

2.3.1 Routing

Routing is the transfer of data from a source to a destination. Routing has two important goals, the first of which is to find and choose the best path from the source to the destination. Choosing the best path means the shortest one, minimizing the number of intermediate nodes, and the time the process takes. The second goal is to transfer the data by connecting and communicating MANET nodes. This process requires rules to allow nodes to communicate with each other, called protocol [21], [19].

In MANET, routing is accomplished using routing tables, which are saved in the cache memory of each node.

2.3.1 Routing types

There are various types of routing techniques:

a. **Source routing**: The source node decides which the best path to the final destination is. Intermediate nodes only forward the packet to the destination, such as dynamic source routing protocol (DSR) [21], [19].

a. **Hop by hop routing**: The source node only decides which the next node is, and each following node decides the next hop, such as adhoc on demand distance vector (AODV) [21], [19].

2.3.2 Routing classifications

Many routing protocols have been discussed and proposed for MANET. They can be classified into three categories figure (2.2):

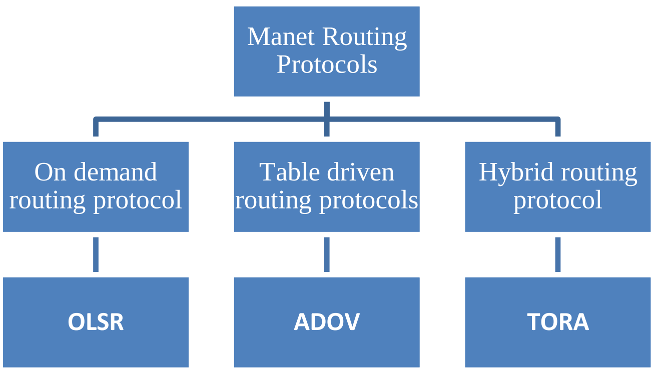


Figure 2.3 MANET routing protocols classifications

2.3.2.1 On demand routing protocol /Reactive protocol:

Reactive Protocol is also called on demand routing protocol. Nodes use these protocols only when required; they don't save a predefined route to the destination, but rather when a node needs to send data, it is sent based on a flooding algorithm using a discovery process.

The source node sends a route request message (RREQ) to all of its neighbors. Each of its neighbors then forwards the RREQ to their neighbors until it reaches the destination. A route reply message (RREP) is then sent back to the destination, giving information about the destination's place. This type of protocol minimizes the number of hops by finding the shortest path. On the other hand, however, this may cause a high delay protocol [19].

This type of protocol can works in an optimal way even in large networks.

Types of on demand routing protocols:

b. Adhoc On Demand Distance Vector (AODV).

c. Dynamic Source routing protocol (DSR).

d. Temporally ordered routing algorithm (TORA).

e. Other types of protocol.

2.3.3.2 Table driven protocols/proactive protocol:

Proactive protocol is also called table-driven protocol. Nodes that use these protocols (more specifically, all participating nodes) update their routing table every increment of time (discover the network) even if there isn't a request. When a node needs to send a data packet, it is sent through a predefined route discovered earlier. This protocol reduces traffic load because all routes are predefined all the time [21]. One disadvantage to this protocol is that too much saved information might slow the process of reforming link breakages.

In this type of protocol, energy conservation is high due to its discovery process to routes which may go unused. This type of protocol can work in an optimal way only in small networks.

Types of table driven routing protocols:

a. Optimized Link State Routing (OLSR)

b. Destination-Sequenced Distance Vector (DSDV)

c. Fish-eye State Routing (FSR)

2.3.3.3 Hybrid routing

A new protocol was created attempting to utilize the advantages of both proactive and reactive protocols. These types of hybrid routing protocols.

	Routes	First packet delay	Power consumption	Route to every node
Proactive;	Routes must be updated and available	First packet delay is less than reactive protocol	Power consumption is higher than reactive protocol	Available
Reactive	Route is created when needed	First packet delayed more than proactive protocol	Power consumption is lower than proactive protocol	Not Available

Table 2.1 Differences between proactive, reactive protocols

Description of reactive routing protocols

Reactive protocols are on-demand protocols. Routes are created only when a source demands so, initiating a discovery process within the network. When an optimal route is found to the destination, the process is completed. Another process used in the reactive protocol is maintenance process, which ascertains that only valid routes exist, and deletes any invalid ones.

2.4.1 Dynamic source routing protocol (DSR)

Dynamic Source Routing protocol is an on demand routing protocol, developed by Johnson, Maltz, and Broch to enable users to communicate over wireless links [13]. It works based on the concept of source routing and doesn’t depend on a routing table. The source node demands to send a packet to the destination.

This protocol consists of two mechanisms: route discovery and route maintenance, which work together to allow nodes to discover and maintain routes. The optimal route from the source to the destination is found by a discovery process. Route maintenance is responsible for checking that the path is optimal path remains valid.

Each source node adds to its header packet a complete path to the destination. Every node in the path forwards the packet to its next existing hop in the header, without needing to check its routing table. Figure 2.4 illustrates the way DSR works.

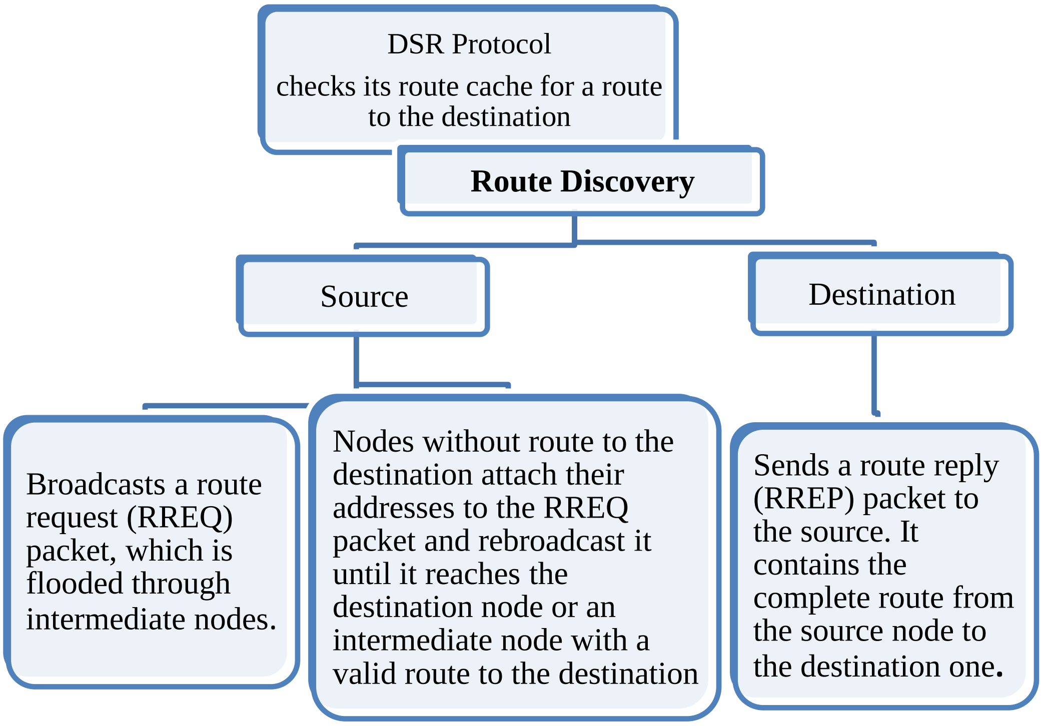


Figure 2.4 How DSR works

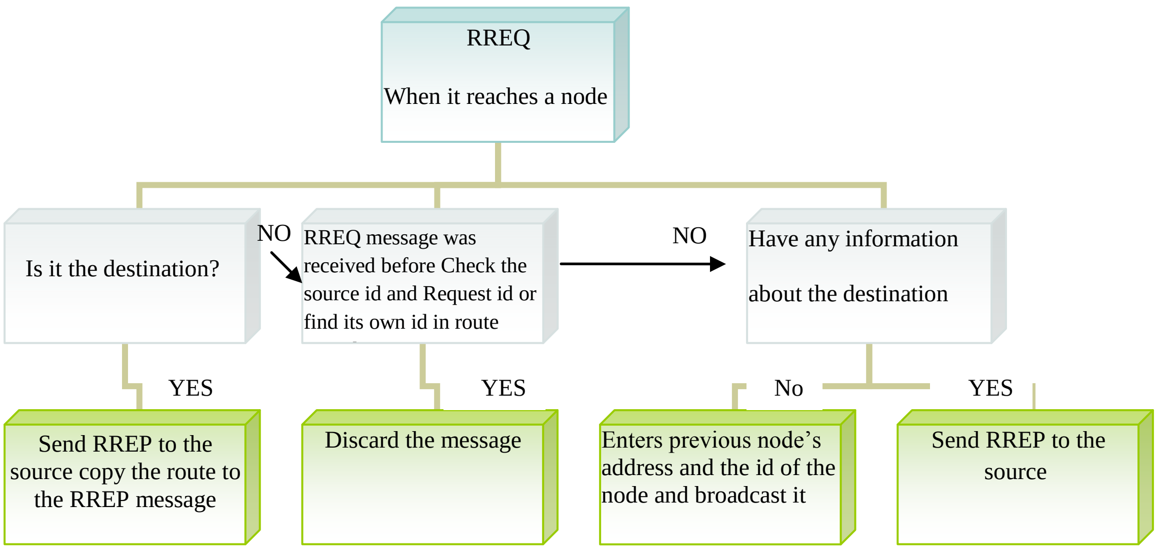


Figure 2.5 DSR when A RREQ message is received

- **Route discovery**

The source node fetches its route cache for a valid route. If it doesn't find any, it sends a route request message (RREQ) to the entire network, using a flooding process as shown in figure 2.6. Each node maintains a table that contains all RREQ messages received recently, and each new RREQs message will be entered in the table on a pair (initiator, request id). When the packet is received it is first checked whether the TTL (Time To Live) counter in the packet is greater than zero; if not, the RREQ message is discarded. If yes, then it checks if this node is the destination. If it isn't, then it reviews the table, to see if the RREQ was received earlier, by checking (initiator, request id). If it was received before, it broadcasts the one that was received first, and discards any others.

When the RREQ packet finds the destination node; the destination node sends a reply message (RREP) on the reverse path to the sender as shown in figure 2.7.

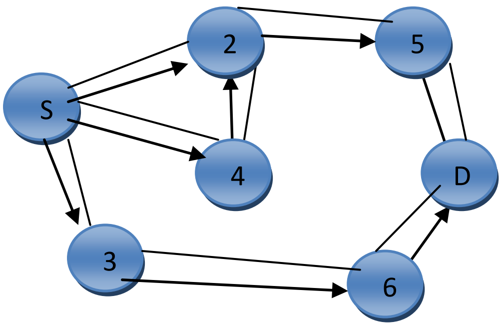


Figure 2.6 Flooding process of RREQ message from the source (S) to the destination (D)

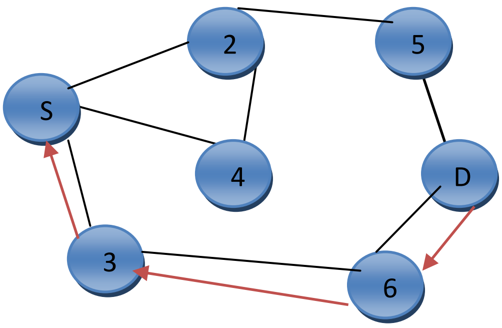


Figure2.7 Optimal paths from the source (S) to the destination (D) RREP message

The source node then stores the routing information in its routing cache to provide the route to the destination to begin sending data.

- **Route maintenance**

When there is a broken link between two nodes, a route error packet (RERR) is sent by the participating node back to the source node. The source node first removes any route entries in its cache to that destination node, and then it initiates the route discovery phase again to find a new path to the destination.

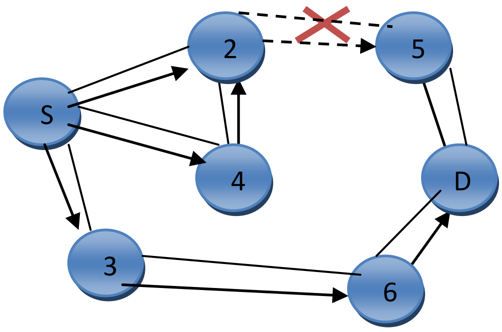


Figure 2.8 Example of route maintenance DSR

As shown in figure 2.8, if there is a link between 2 and 5 fails, node 2 sends a route error to S by the route 2-S; all other nodes hear that there is failure between 2 and 5.

- **DSR issues**

1- In source routing DSR, nodes don't have to broadcast every period of time their routing tables to the neighboring nodes. This saves a lot of network bandwidth and energy.

2- In source routing DSR, nodes do not need to maintain routing information in order to route the packets that they receive.

3- When network size increases, the routing overhead increases because each packet has to carry the entire path to the destination with it.

4- The use of route caches has benefits in reducing the propagation delay.

5- On the other hand, using cache often reduces performance [21].

6- Link breakage is not repaired in the maintenance phase, but re-initiates a new route discovery process.

2.4.2 Adhoc on-demand distance vector routing protocol (AODV)

ADOV protocol is a hop by hop protocol also called reactive protocol. Routes will be created and updated only when needed. A hello messages is broadcast at intervals to keep track of its neighbors, and each node keeps track only of its next hop, not the entire route [23][24][25].

There are three types of messages in AODV routing algorithm

a. Route Requests (RREQs)

b. Route Replies (RREPs and Route Errors (RERRs)) [21].

When a node requires communicating with a specific node which is not its neighbor, it broadcasts a RREQ message to the entire network. Parameters that a RREQ message contains are as follows:

2 **Route request messages format**

0 7 8 16 24		
TTL (8 bits)	Previous Hop (8 bits)	Next Hop(8 bits)
Type	G	Hop Count
Request ID	Destination IP Address	Destination Sequence Number
	Source IP Address	Source Sequence Number

Figure 2.9 Route Request Messages Format

a. Source IP Address.

b. Destination IP Address.

c. Source Sequence Number: the last sequence number, the recent, to be used in this route.

d. Destination Sequence Number: the last sequence number, the most recent, and received by the source for any route to the destination.

e. Request ID.

f. Time to live (TTL).

g. Hop Count: number of hops from the source to the intermediate node trying to find the destination.

3. **Routing table format**

Each node in the AODV adhoc network contains information about the recent route by keeping the following data:

1. Destination node IP address.

2. Destination sequence number.

3. Hop Count: Number of hops to destination.

4. The next hop to forward the packet in a route.

5. The valid time for a route.

6. Active neighbor list.

7. Request buffer to be sure that one request will be processed once.

Each route table entry for every node in the network must contain the most recent sequence number for the nodes. These entries are updated whenever RREQ or RRER messages are received.

To discover the network, all nodes send and receive hello message to and from each one's neighbors.

First, each node checks its route cache (route table) for a route to the destination, called an active route. If it doesn't find a valid route, it begins the route discovery process.

- **Route discovery**

The node broadcasts an RREQ to its neighbors to find a valid route to the destination. RREQ is identified by the source address and request id. Every time a source sends a new message, the request id will be incremented, and the node that receives the RREQ message checks if the RREQ has reached the destination. If not, it will check if this RREQ was received earlier, by checking the request id and source id. If it was received before, this message will be discarded, and if not, the node will re-broadcast the RREQ to its neighbors and increment the hop count.

How this node can determine if it reached the destination or an intermediate node is by finding a route that is fresh enough. Fresh enough means that the sequence number in the table is near the sequence number in the RREQ message. In AODV, sequence numbers are used and updated at the destination to ensure the freshness of routing information.

Every participating node receives an RREQ message, following which it enters the previous node’s address and the id of the node that broadcasted the original RREQ message, to the previous node [21].

If a neighbor doesn’t have any information about the destination, it will rebroadcast the message to all of its neighbors, and so on. If it has reached the destination, or an intermediate node which has a route to the destination, it will send a route reply message to the source that sent the RREQ message by sending the RREP to the neighbor which will send the RREQ message. The neighbor will do the same until it reaches the source, a process which is called the reverse path. When the route reply message reaches the source, this route would be called a complete route, and source can begin sending packets of data.

When a node receives an RREP message, information about the previous node is stored in it, to forward the packet to it as the next hop of the destination. Each time an intermediate node receives an RREQ message, it sets up backward path information. RREP indicates the route from destination to source, and data is forwarded according to that path.

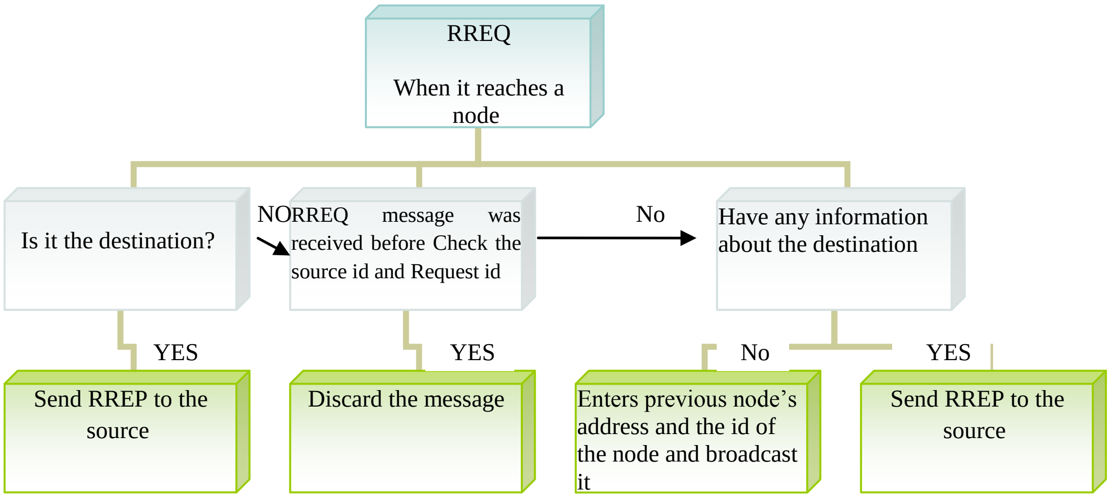


Figure 2.10 AODV when A RREQ message is received

- **Example on AODV discover process**

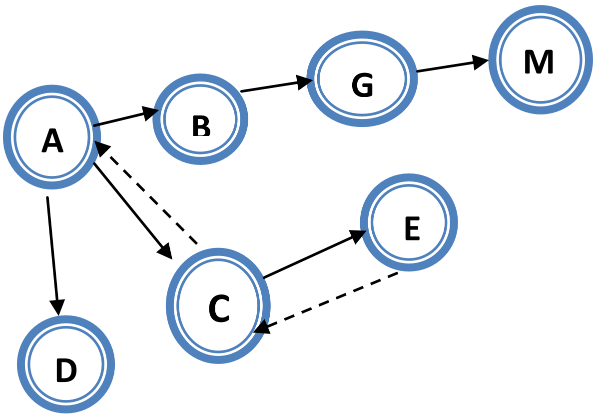


Figure 2.11 AODV example

If we suppose that node A wants to send packets to node F, then node A will check in its cache for a valid route. If it didn’t find a valid route to F, node A starts a route discovery process: it will prepare RREQ message which will contain a source identifier, destination identifier, a sequence number for the source, a sequence number for the destination Broadcast identifier and time to live for the RREQ.

And send it to its neighbors. When the RREQ packet reaches them, these nodes check their route caches for an existing route to node F. If they haven’t found any valid route, they forward the RREQ to their neighbors. However, if they did find a route to node F, they will compare the destination sequence number in the RREQ packet and the destination sequence number (DSEQ) in their route cache. If the DSEQ number in the RREQ message is greater than the RREQ in the route cache, it will send a route replay message to the source node.

When it sends route reply message from node E to node C, the intermediate node will save and update the destination sequence number in their routing table, to match the RREP packet, which is sent from the destination to the source.

- **Route maintenance**

When link breakage is found to a known node as an active node, it generates a route error message (RERR) and broadcasts the message to its active neighbors. This is called reverse or recursive process, which continues until the source will receive the message and sends a new RREQ for an alternative route [21].

- ## Difference between AODV and DSR

ADOV and DSR routing protocol share many characteristics: they only start a route discovery when there is no information to the destination and AODV stores information about other nodes in each node routing table of the network. DSR, however, uses it cache to keep a complete path to the destination.

2.4.3 OLSR (Optimized Link State Routing)

OLSR is the optimization of link-state routing algorithm. Link state is a concept in which every node in MANET creates a plan or a map that contains all relevant information about the network. In other words, every node is known in the network, to which node it is connected. This process is done at intervals, to update topology information at each node.

OLSR works on three main concepts: neighbor sensing mechanism, efficient flooding mechanism, and how to select optimal routes [22].

OLSR is also a table driven, proactive protocol, meaning routes are always available when needed. Before any source node intends to send a message, routes are built by a process in which each node in the network sends HELLO messages to their neighbors to ensure connectivity between nodes, a process called neighbor sensing.

If two nodes D and C are neighbors, D sends hello messages to C. This link is called asymmetric, meaning there is connectivity with D. If C also has a connection with D, this link is called symmetric, two way communication, as shown in figure 2.12.

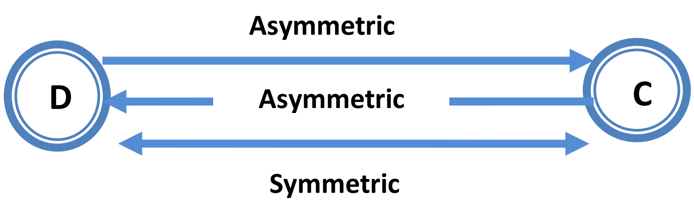


Figure 2.12 Connectivity between D and C nodes

Three types of messages is used in OLSR: hello messages, TC messages, MID messages

- Hello messages: Used for gathering information about the link status and neighbors'. They help in choosing MPRS, and these kinds of messages are sent up two hops away, exclusively.
- Topology control (TC) messages: Broadcast to the entire network, and are used to advertise set of neighbors for each node.
- Multiple Interface Declaration (MID) messages: which are broadcasted to the entire network. These are broadcasted exclusively by the MPRs, and these kinds of messages are used to advertise to other nodes that this node can have multiple interface address [5], [29], [33].

- **OLSR packet**

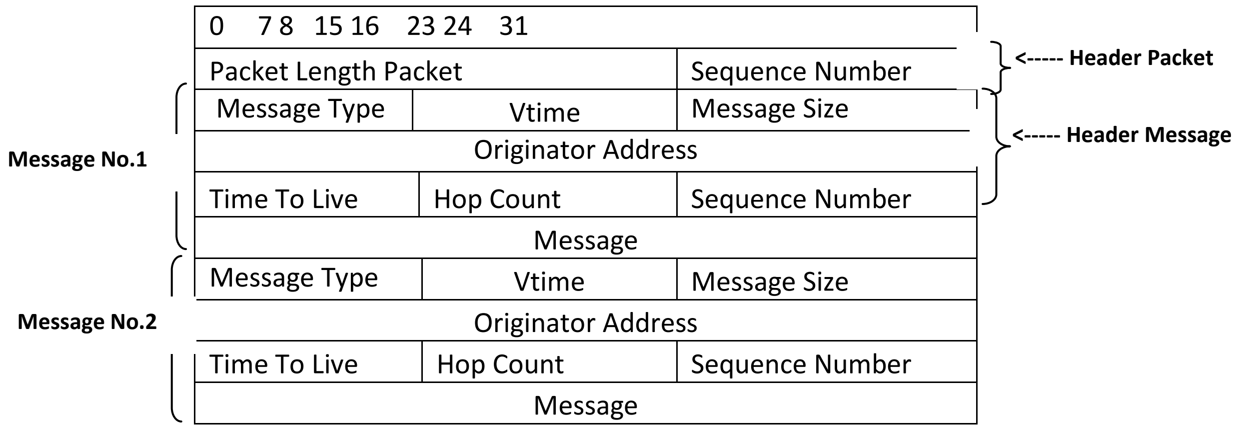


Figure 2.13 OLSR packet format

OLSR packet formats consist of two parts: packet header, packet body.

1. Packet header contains packet length and packet sequence number, updated by each interface of OLSR nodes.

Packet body contains one or more OLSR messages. Each message in the packet contains a message header, which includes message type, validity time (VC), message size, originator address, the source address that created the message, time to live, hop count and a message sequence number

- **HELLO messages format**

0 7 8 15 16 23 24 31		
Reserved	Htime	Willingness
Link Code	Reserved Link	Message Size
Neighbor Interface Address		
Neighbor Interface Address		
…		
Link Code	Reserved Link	Message Size
Neighbor Interface Address		
Neighbor Interface Address		

Link type	Neighbor type

Figure 2.14 OLSR HELLO message format

- **OLSR hello message format**

Hello messages are used as link sensing. They sense link statuses between a node and it neighbors. It is used also as neighbor detection, in which at every period of time, each node broadcasts a hello message that contains information about its neighbors and the link status [5], [29], [33].

Link code: link type, neighbor type.

HTime: Holding time which contains the time intervals at which hello messages will be broadcasted.

Willingness: (will never, will always): This field specifies the ability of a node to forward traffic to and for other nodes. Nodes specified as ‘will always,’ will always be selected as MPR and those with ‘will never,’ will never be selected as MPR.

- **Topology control message format**

TC messages are broadcasted using MPR messages

0 7 8 15 16 23 24 31	
ANSN	Reserved
Advertised Neighbor Main Address	
Advertised Neighbor Main Address	

Figure 2.15 OLSR TC Message

- **OLSR mechanism**

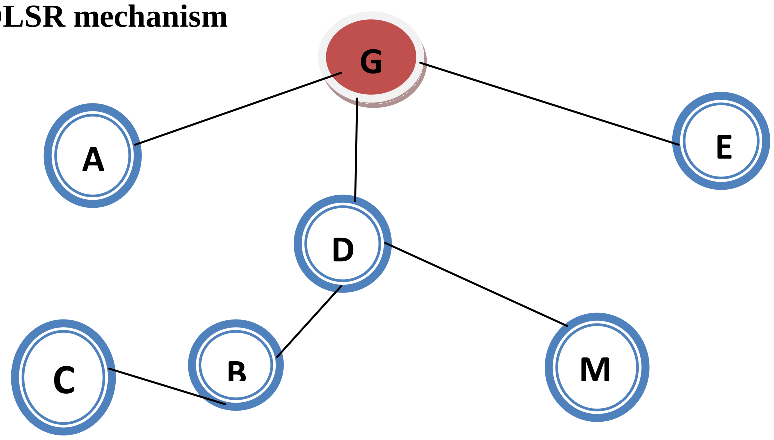


Fig 2.16 Network example to select MPR

Every node will send a hello message to it neighbors, to check link status and to select its one and two hops neighbors'. Each node saves the information hello messages have gathered, in a table called neighbors table. It enters a holding time for each neighbor; in which holding time expires they are removed. It also decides, based on Hello messages, its set of multi point relay (MPR), and enters a sequence number to each MPR in order to identify the most recent MPR in its MPR table that created. MPR is an optimization of flooding standard, and is as an intermediate node or an interface, in which all other nodes can communicate with a specific node through MPR. MPR is selected by one hop neighbor, and after selecting each node its MPR set, it will advertise to all nodes which MPR set was selected through the next hello messages that will be broadcasted. Each node will broadcast and forward TC messages only through MPR nodes. Based on MPR selectors and TC messages, nodes will update topology tables to record the MPR of other nodes. A route table will be created at this stage, based on topology table and neighbors table, as figure 2.16 illustrate. To communicate with node G, G sends a hello message to a node that is far away from G. One hope is saved in a table and all nodes that are two hops or further are determined by attaching each node to its list of neighbors. Once node G knows its one and two hop neighbors, it can decide which their MPR node is. In the case of G, its one hop neighbors are A, D, and E; its two hops neighbors are R, M and the MPR is D.

MPRs are responsible for the transmission of broadcast messages during flooding, and for generating link state information.

Every node keeps a table of routes to all known destinations through its MPR nodes.

Four things will be changed when hello message arrive: a neighbor list will be updated by specifying the link as a symmetric or as an asymmetric neighbor; hop sets store a list of node pairs by specifying which two hop neighbors can be reached, through which symmetric one hop neighbor; MPR set maintains the set of nodes which were elected as MPRs by this node [5], [29], [33].

MPRs nodes forward packets for the nodes that selected them as MPRs and announce to all nodes that selected them as MPR by topology content packets, as well as to the rest of the network.

Regular nodes that are not MPRs can receive and process control packets but cannot retransmit them and cannot dictate network topology to other nodes in the network.

The optimal route for each node is calculated based on its one hop and two hop neighborhood, and the topology information.

- **OLSR issues**

1- MPRs technique reduces the message overhead (reduces retransmission in the same range.)

2- Minimize the number of control messages flooded in the network.

3- Reduces the route discovery delay.

2.4 OSI model

OSI is an open system that enables two systems to communicate with each other without making any changes; it's a layered system for the flow of a network process.

It consists of seven layers: physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer [48].

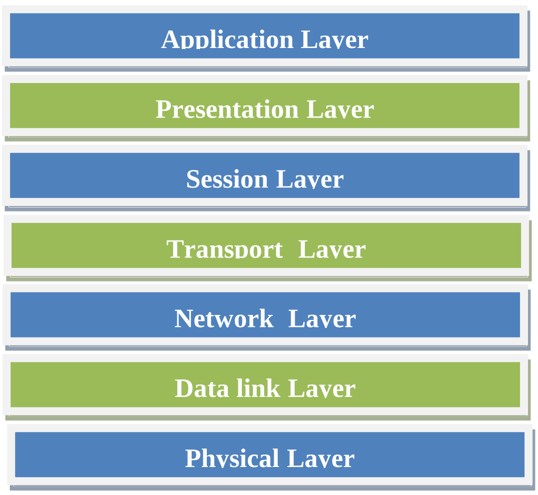


Figure 2.17 OCI layers

a. Application layer

In this layer, the end user will communicate with an application such as FTP, HTTP, or email to transfer data, access data and remote sharing. This layer can be called the end user layer [48].

b. Presentation layer

This layer is responsible for data format, data security and data compression. In other words it the translator to the network

Data format is responsible for converting the file sent, from a file format into the language format. It must be a general format in syntax and semantics for all users.

Data security is encryption of data by cryptography and data compression [48].

c. Session layer

This layer is responsible for establishing, managing and terminating connections and sessions between applications.

d. Transport layer

This layer is responsible for ensuring that all messages are delivered in sequence and without any loss or duplications. It accepts a message from the layer above and splits the message into smaller parts, and provides a message of acknowledgment and traffic control by notifying the transmitting station when there are no messages in its buffer.

e. Network layer

This layer is responsible for deciding which physical path the data should go, according to network conditions, by being responsible for routing. It controls subnet traffic, frame fragmentation, and conducts logical-physical address mapping [48].

f. Data link layer

This layer is responsible for the transfer of data frames, without errors, from one node to another. By being responsible for link establishment and termination, frame traffic control, frame sequencing, frame acknowledgment, frame delimiting, frame error checking, and media access management [48].

g. Physical layer

The lowest layer of the OSI model, its main responsibility is the transfer of data as bits from one place to another.

It defines the main characteristics of the communication interface, transmission media, number of bit sent, and type of encoding to transfer bits to signal or digital baseband. Finally, it defines how nodes are connected -the topology [48].

2.6 Applications FTP, HTTP, Database

2.6.1 Server

A program that runs on a computer, considered a remote computer that provide services. It has the ability to receive a request from local computers and provide services [31].

2.6.2 Client

A program that runs on a local computer, considered as a client computer that demands a service from a server [31].

2.6.3 Socket interface

An interface is a group of instructions to communicate with another machine. Those instructions are, like any instruction used, and will demand the machine to open the transport layer to send and receive data.

Socket is software or a program that will request the operating system to create a socket on the hardware to enable the application to send and receive through it [31].

2.6.4 FTP (File Transfer Protocol)

FTP is a two way protocol that helps transfer data between two nodes over a network. When a user requests to transfer a file between two computers, the user will use FTP.

To transfer a file from a client to a server, uses a process called uploading; to transfer a file from a server to a client is called downloading.

FTP uses two types of connections: the first type is for commands, named command or control connection. The other is for sending and receiving data, and is named a data connection. See figure [2.18].

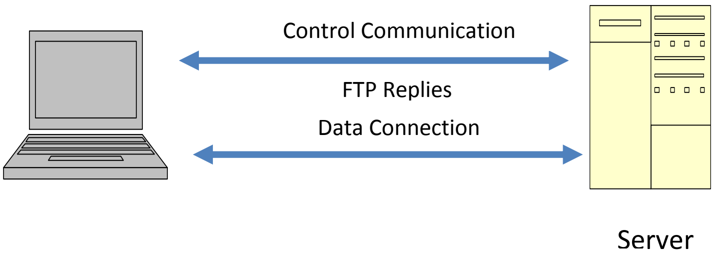


Figure 2.18 Types of FTP connections server

A data connection is opened and closed each time a file is transferred. A control connection is opened for once the session begins and closed when the session is closed. Sessions may contain more than one file to be transferred.

- **FTP ports**

FTP uses a logical connection point called a port for communicating. Two such ports are used in FTP: one for commands, the command port; the second is for sending and receiving data, the data port. The standard port number that is used for commands is 21 and the standard port number used for data is 20. The port used for sending and receiving data depends on the mode connection. Mode connection determines who will connect the server or the client [31].

- **FTP modes**

FTP uses two types of modes connection in data connection, active or passive modes.

In an active mode connection, the client will open a port, listen, and the server will connect.

Passive mode means the server will open a connection, listen, and the client will connect.

The client needs to define three attributes to solve the problem of different types of data between the server and the client: file type, data structure, and transmission mode [31].

- **FTP file types**

There are three kinds of file types. The first one is ASCII file. This type of file is for transferring text files. The second file type is EBCDIC, which is a type of encoding. The final type is Image file, which is a binary file and is sent as bits.

The data structure of the file to be sent must match how the file will be divided: if the file is text, it can be divided as records. The file can be divided as pages, where page includes number and a header, or it can be sent as a stream of bytes [31].

- **FTP transmission mode**

Transmission mode is the shape of data delivered to the TCP. As stream mode, data is delivered to TCP as stream of bytes; as block mode, data is delivered as blocks, in which every block will contain 3 bytes. The block descriptor is the first byte, and the remaining bytes are for the size of the block. The last type is compressed mode, wherein if the file is large, it can be compressed [31].

- **FTP sending data process**

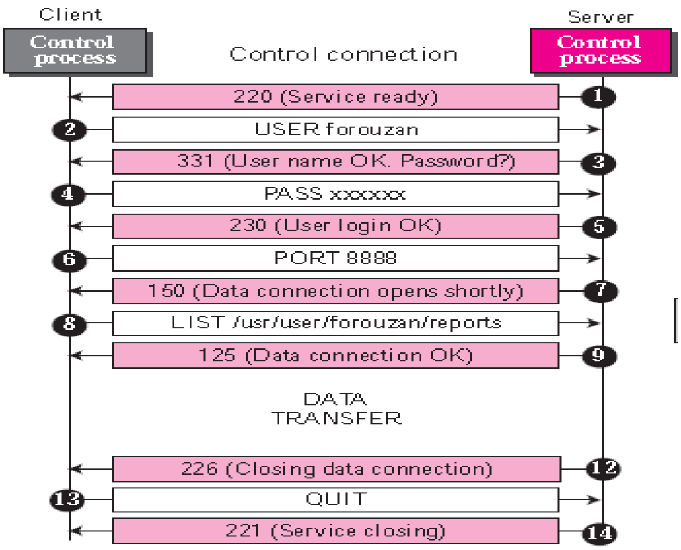


Figure 2.19 FTP Process [31]

When a client demands to connect to a server, a control connection will be created on port 21. The server will check to see if the service is ready, and it will send command 220 to the client, which means the server is ready. The following step will be an authentication step, in which the user will send their user name and the server will check to see that it is correct and respond to the client asking for the password. The user will send the password and the server will also check if this is correct and respond with 230 commands, which means everything is correct and the client is logged on successfully [31].

Creating the data connection is done usually by the client. The client will send a passive open command on a temporary port for data connection, and send this port number to the server. The server prepares to open an active open between port 20 on the server and the temporary port sent by the client. The server will respond with a 150 command which means a connection will open for a short time. The client will send a LIST command which means the name of directory client needs to open, and the server will send a 125 data connection will be opened and data transfer will begin. After transfer is complete, the server will send a 126 command, meaning the data is transferred, and it will demand to close the connection. The client will either respond to quit or need to start another process.

- **FTP sending data process through network layers**

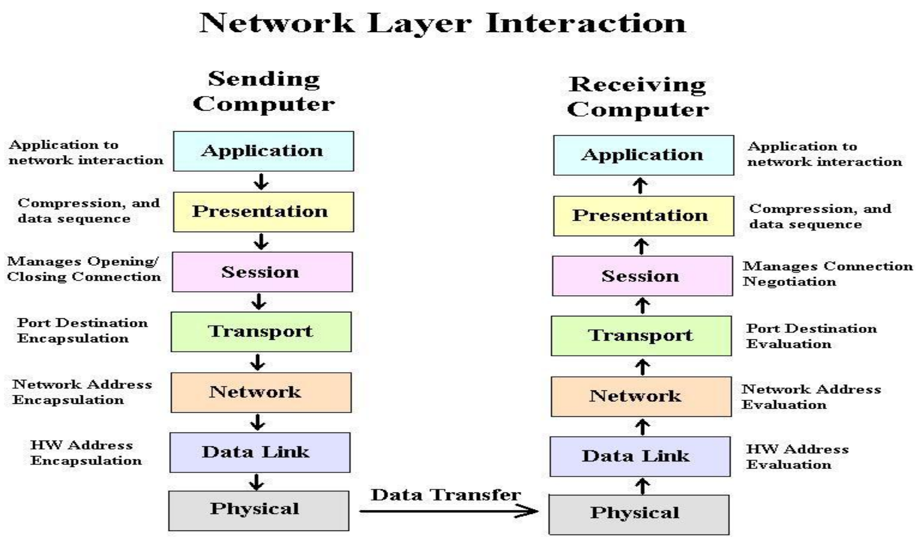


Figure 2.20 How FTP works through network layers [17]

Figure 2.20 describes how OSI layers works with FTP when a user sends files from one computer to another. The user will first open the FTP client program such as cute FTP, and specify the address that he wants to send data to, as illustrated in figure 2.20. The application layer is the layer which enables users to pass data to the session layer through the presentation layer. The session layer is responsible for opening and establishing connection with the client by sending synchronization signals between client and server. The client will send synchronization signals to the transport layer, and the transport layer will add a TCP header which contains the source port and destination port, and pass it to the network layer, which will add the source IP address and the destination IP address and then it will pass it to the data link. It determines the hardware address of the computer receiving the data that will then be transmitted across the physical wire [17].

2.7 HTTP (Hyper Text Transfer Protocol)

Hyper text protocol is a transfer protocol that transfers files from a web server to a browser. It works just like FTP, but uses one connection, a data connection. HTTP uses port number 80 to connect and transfer data between the client and server. A request message is sent from the client to the server, to which the server replies. HTTP is a stateless protocol; it doesn't save any information about the client [31].

- **HTTP messages format**

1. Request message

HTTP Request messages consist of three parts: request line, header and body.

Request line

Request line contains three parts: methods, URL, and version. Method is the request type. For example, a request for a file from a server or a request of information, would request to send a file from server to the client. Version is the version of the http protocol, for example http 1.1.

Header

Header line sends extra information from the client to server. There may be more than just a header in the request message, and each header has a header name and a header value.

Body

Body contains some comment sent from the client to the server.

2. **Response message**

HTTP response messages consist of four parts: a status line, header lines, a blank line and a body.

Status line: Status line consists of three fields: the first is the version of HTTP; the second is status code, as a result of the request message.

Header The Header line sends extra information from the server to client. There may be more than header in the response message, and each header has a header name and a header value.

Body contains the actual file client requested from the server.

- ## HTTP connections

There are two types of connections: Persistent connection and non persistent connection, the difference between them is that in the persistent the server leaves the connection opened after the response for more requests, whereas non persistent doesn't leave it opened. HTTP version 1.1 uses persistent connection [31].

Persistent connection

Client will open a tcp connection by sending a request message to the server, after which the server sends a response message that will contain the request of the client in the body and the connection will remain, opened until a time out will occur as shown in figure 2.21

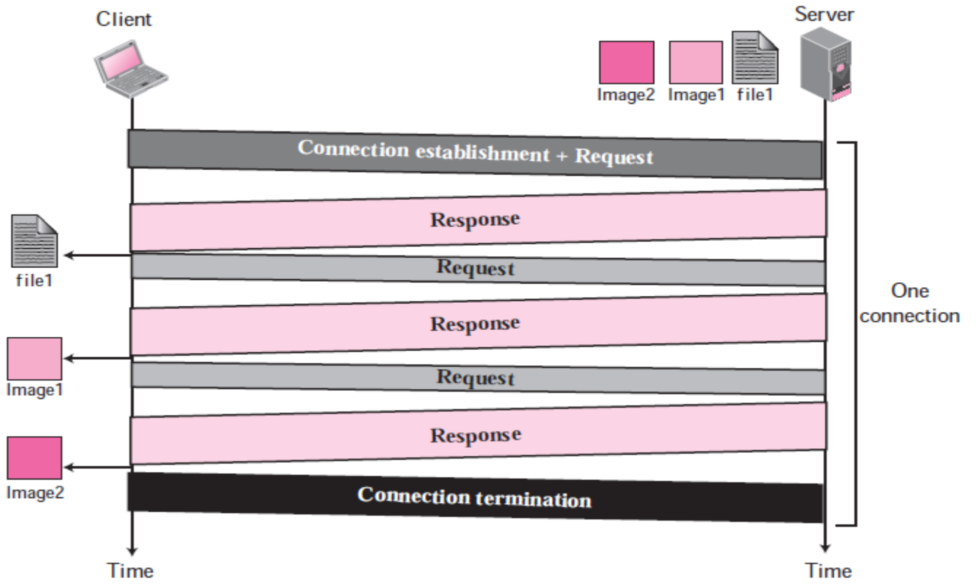


Figure 2.21 HTTP example process [31]

Difference between Http and FTP

FTP and http use TCP connections. FTP uses two connections, while http uses one. FTP is used to upload and download files, by copying the file to the client while http is used only to view webpage's. FTP requires a username and password while http doesn't.

2.7 Event driven simulation

In real life, examining performance of any network is considered to be complicated and difficult; so many event-driven simulators were suggested. Event driven simulation is a modeling paradigm where control flow is controlled by events (not by sequence) and events are organized by an event scheduler and represented by an event queue.

Some Examples on Network simulator that were suggested: NS2, (Network simulator 2), QUALNET, Glomosim, OPNET (Optimized Network Engineering Tools), among other simulators that make it easier to design, and examine a network [49].

2.8 Simulation

In our thesis, we have used OPNET modeler 14. Why using OPNET? OPNET is a comprehensive development environment which supports a huge number of built-in standard network protocols, devices, and applications. OPNET is a user-friendly program because it contains GUI, and it helps user to analyze results in a particularly flexible way [14].

2.8.1 OPNET work flow

There are four main steps when using the OPNET simulator:

The first and main step is to design the network. In other words, create the model by specifying parameters. The second step is to choose which statistics on which level the user wants to examine. The third step is to run the simulation, and finally, to collect and interpret the results. Figure 2.22 illustrated the main steps of workflow.

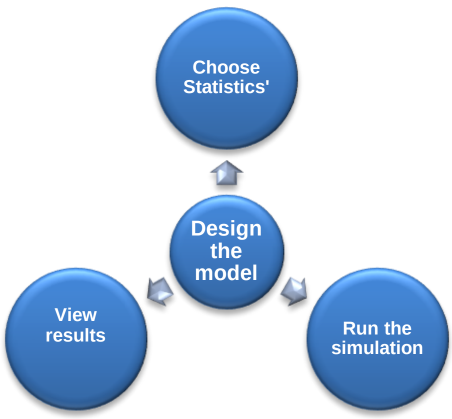


Figure 2.22 Main steps of OPNET workflow

2.8.2 Building the model by OPNET

There are two ways to build a model using OPNET: the first way is automatic, by using the wizard; the second is to build a model manually.

Building a model with the wizard is done relatively simply, by starting the wizard. The project editor will appear and here, the user can choose network environment he or she desires to build. The second way, to build a model manually, is done by dragging objects (i.e. server, MANET wireless node, etc.), any kind of object from the object palette to the project editor workspace. After finishing, nodes need to be configured in two ways by specifying parameters.

The main entities that must exist in any model are application configuration, profile configuration and nodes. Server and mobility configuration are needed in our simulation, besides the main entities mentioned earlier.

- **Application configuration**

Application configuration is used to specify the applications, which will be used in our simulation. This can include FTP, HTTP, or Database.

- **Profile configuration**:

Profile configuration is used to create user profile that will identify what application will be used on each node, by choosing or specifying the application in the profile.

2.8.3 Performance metrics

MANET routing protocols can be examined and evaluated by many different kinds of performance metrics, but to study network behavior of routing protocols, specific metrics need to be examined. The most important metrics for our study are delay throughput and network load.

- **Delay**

Delay is a performance tool that measures the efficiency of a communication network by measuring the average time a packet takes to begin and end its trip from the source to the destination. It is called end to end delay, though it can also be called also latency in many cases. Delay can be expressed in three kinds of delays: transmission delay, propagation delay and processing delays [15]. Delay is a very important metric and can be considered a critical parameter to be studied.

End to end delay = transmission delay + propagation delay + processing delay.

- **Throughput**

Throughput is the ratio of all number of bits a destination node receives from a source, over a communication network. Throughput is considered an accurate choice to measure performance of a network.

Throughput= (number of delivered packet * packet size)/ simulation time [20].

- **Network load**

The total load in bit/sec submitted to wireless LAN layers by all higher layers in all WLAN nodes of the network [20].

2.9 Mobility modeling

Mobility model represents the movement of the nodes from the beginning of the simulation. To simulate the movement of nodes in MANET, many mobility models have been proposed [21].

2.9.1 Random waypoint (RW)

The Random waypoint model [16] is a way a node moves, each node is placed initially at a random position within the area of the simulation.

When the simulation begins, each node chooses a destination and sends packets to it with a constant speed that is randomly selected from the interval [*v*min, *vmax*]. After that, it pauses for period called the pause time

Chapter 3

Literature Review

Wireless technology is growing more and more these days, and the need for a wireless network is becoming more and more vital, so that users can access the internet or applications without being constrained to a place or time. This is what makes MANET an important field to be studied, in satisfying the needs of a wireless network.

Mobile Adhoc Network (MANET) is a new wireless communication network; from 25 years until now, it was the interest of many researchers in analyzing and developing its features.

The trip of MANET began from 1970 and kept developing for many years, to reach the present day, with the huge proliferation of technology.

A review of many studies from 1998 until today follows:

In the beginning of October 1998 an article was published [49] A performance comparison of multi-hop adhoc wireless network routing protocol. It measured three metrics: packet delivery ratio, routing overhead, and hop count, for four multi-hop wireless adhoc network routing protocols on 50 mobile nodes network. These protocols included Distance-Sequenced Distance Vector (DSDV,) Temporally Ordered Routing Algorithm (TORA,) Dynamic Source Routing (DSR,) and Adhoc On Demand Distance Vector (AODV.), results showed when mobility increases, packet delivery ratio for DSDV decreased more than the rest of the protocols, DSR and AODV packet delivery ratio is independent of the traffic load. [38] At the same year, several scenarios were measured to compare between TORA and ideal link state (ILS) by varying network size, changing topology and network connectivity. Results showed that when increasing network size and changing topology, TORA performed better than ILS. For network connectivity, it was found to not be a significant factor to be measured. [46] measured performance of MANET protocol by measuring fractions of packets delivered, end-to-end delay and routing load using MaRS (Maryland Routing Simulator)[47], results showed a good packet delivery and delay performance link state and distance vector protocols provide, in general, better packet delivery and delay performance. In [37] 2000 they measured performance of DSDV, TORA, DSR, and AODV using NS-2. Their main goal was to measure how routing protocols will react with changing network topology, while doing its original work in delivering data packets to each destination.

By measuring the lengths of the routes over which the protocols had to deliver packets, and the total number of topology changes in each by measuring packet delivery ratio, routing overhead, the total number of routing packets transmitted during the simulation, Path optimality, the difference between the number of hops a packet took to reach its destination and the length of the shortest, path that physically existed through the network. Results showed DSVD delivered all data when network and movement speed is low; when it increased, it failed to cover all nodes. TORA delivered 90% of packets with 10 and 20 source nodes. But when increases to 30 sources, it was unable to control traffic generated. DSR was superior even when speed and number of mobility changes. AODV also performed well like DSR, but is expensive since it requires the transmission of overhead of packets at high rates. Also in 2001 [1] compared the performance of two on demand routing protocols, dynamic source routing (DSR) and adhoc on demand distance vector routing (AODV), both of which use route discovery, but vary in mechanism. The metrics which were analyzed include packet delivery fraction, throughput, average end to end delay, and routing load. Results showed DSR performed better than AODV in small networks, while AODV performed better than DSR in large networks.

In [39] 2003 began to expand to examine new factors. In the previous studies of mobility rate, speed was examined without considering the network size. In this study, network size was considered by testing the routing performance of four different routing protocols (AODV, DSR, LAR 1 and ZRP), in different network size. The researchers used QUALNET simulator. The performance of AODV was found to be superior to DSR in all network sizes.

LAR1 performed better than AODV for 200 nodes in routing overhead, and delivery ratio.

In 2002 [40] used Constant bit rate traffic to evaluate OLSR and AODV. Results showed that OLSR had better performance with high dynamic topology, while AODV had less over head when networks remain static.

In 2007 [43] more scenarios have been developed to analyze the performance of AODV, DSR and OLSR protocols, using different numbers of nodes and different variants of TCP, TCP tahoe, reno and new reno. Throughput was measured: protocols that used TCP variants had lower throughput as the network size increased, while DSR and TORA had high delay when using TCP variants. Congestion was highest in TORA than other protocols.

In 2008 [42] measured variable bit rate (VBR) traffic using NS2 from AODV, DSR and OLSR. The observations from this simulation indicated that DSR performed well in Delivery ratio, while AODV had less delay. DSR has lower overhead than other protocols.

In 2011[41] two proactive protocols (AODV, DSR) and one reactive protocol (OLSR) were measured using different number of simultaneous video transmissions. Packet delivery ratio, delay, packet delay variation (jitter) and routing overhead were measured. CBR traffic was used with results showing DSR performed best among the protocols. [36] 2010, thesis studied the performances of reactive, proactive and hybrid routing protocols under realistic network scenarios using AODV, DSR, OLSR, and ZPR. Scenarios were made on a real live network, and mobility of nodes was simulated using GPS. Traffic was created by a generation tool using 19 mobile nodes and a base station for 4 hours. Another simulation was carried out by a QUALNET simulator throughput, the results of which showed AODV performed the best between the protocols. Delay shows AODV has the lowest results between protocols, and each live simulation and QUALNET simulator had exactly the same results.

In 2102 [30] a study measured traffic load (HTTP, email, and video conference) on a DSR routing protocol on 40 node high load, speed 10m/sec,800*80 to find out that DSR scored delay when used with video conference, while HTTP scored the lowest in delay. Throughput results showed highest in video conference, and lowest in HTTP. [33] Also measured the delay and throughput of OLSR, AODV, and DSR using different traffic loads on FTP and http with fixed number of nodes (40) on a 600×600 square meter area. Results showed DSR had the highest results when measuring delay in traffic and OLSR had the lowest. In throughput, AODV didn’t perform in a weak, while OLSR had the highest results.

[44] All protocols had the same usual performance; TCP and UDP were used with FTP traffic to DSR, OLSR, TORA and AODV. The results showed that a half packet was lost for TCP, and UDP had higher packet delivery. [45] OPNET modeler 14.5 was used to measure AODV, OLSR and TORA routing protocols using Random mobility. A

throughput analysis showed that TORA had the worst throughput, among AODV and OLSR. AODV showed higher efficiency over high traffic than OLSR and TORA.

[45] Used a QUALNET simulator in low and medium node density. AODV exhibited the best performance, while OLSR and DSR showed lowered performance, and OLSR performed well in both low and high node density.

Security issues have also received the attention of many researchers. To improve security, adhoc network have no fixed infrastructure, no centralized monitoring (meaning nodes cooperate with each other to provide connectivity and services), feature dynamic changes in topology; because of these reasons, mobile adhoc networks are left vulnerable to many attacks.

Our research was concerned with various types of traffic load and applications.

Chapter 4

Experiment Results

We assumed that the performance of some applications using MANET as a platform may depend on the utilized routing protocol, the size of the network, speed of the nodes, and the traffic load. Therefore, in this work, we carried out the simulation experiments to evaluate the performance of HTTP, FTP and database applications using the DSR, AODV and OLSR protocols. The simulations were held using discrete event driven simulation software OPNET (Optimized Network Engineering Tool) modeler version 14.0.

Fifty four scenarios were created using various traffic loads and three network sizes: small size (10 nodes,) medium size (50 nodes,) and big network size (100 nodes.) Each scenario was executed for 10 minutes (simulation time). In each simulation, we checked the behavior of reactive and proactive protocols.

Traffic model

Traffic models are a core component of any network performance evaluation therefore they need to be very accurate. Depending upon the type of network and the characteristics of the traffic in the network, a traffic model can be chosen for modeling the traffic. There are two main models for inter arrival time in adhoc network applications Poison and Pereto models. In this work we used poisons model which base on exponential distribution, since this model is suitable for large number of independent processes.

4.1 Performance Metrics

Scenario size	1000*1000 m
Scenario time	10 Min
802.11 data rate	11 Mbps
Number of nodes	10, 50,100
Nodes speed	10 m/s, 30 m/s
Pause time	of 0 sec
Services	FTP, HTTP and database
Different routing protocols	AODV, DSR and OLSR
Mobility model	Random waypoint mobility model
Applications modes	high load, low load

Table 4.1 Parameters chosen for the simulation

Table 4.1 shows the parameters that have been used in the simulation. We created all scenarios on a 1000* 1000 meter space for nodes to move on, and we used random waypoint as a mobility model. Mobility models represent the movement of the nodes from the beginning of the simulation. Random waypoint model is the way a node moves according to it, in which it assumes that each node is placed initially at a random position within the area of the simulation [16]. When simulation begins, each node chooses a destination and then sends packets to it with a constant speed which is randomly selected from the interval [vmin, vmax]. After that, it pauses for a period called the pause time. In our scenarios, we assumed that the pause time is zero.

FTP load types	FTP high load	FTP low load
Command mix	50%	50%
File size	constant(5000)	constant(1000)
Inter request time	exponential(360)	exponential(3600)

Table 4.2 FTP load parameters

Command mix: The percentage of files gets commands to the total of the FTP commands.

Inter request time: The time between file transfers. The start time of the file transfer is computed by adding the inter request time to the time that the previous file transfer started.

File size: Defines the size (in bytes) of a file that will be transferred.

Type of service: Type of service (ToS) and/or differentiated service code point (DSCP) assigned to packets sent from this client. Two FTP loads were used in the simulation as shown in table 4.1.1. The first is high load, with a file size of 5000 bytes, and total get command to total commands 50%. The second FTP load is low load; the file size is 1000 byte.

4.1.2 HTTP application load types:

HTTP load types	High browsing		Low browsing	
HTTP version	HTTP1.1		HTTP1.1	
Page interval time	exponential(60)		exponential(720)	
Page property				
Object size	Constant(1000)	Medium image	Constant(500)	Small image
Object per page	Constant(1)	Constant(5)	Constant(1)	Constant(5)
Initial repeat	Browse		Browse	
Pages per server	Exponential (10)		Exponential (10)	

Table 4.3 HTTP loads used in the simulation

Page interval time: Is the time between page requests in seconds (start time of a page request is calculated by adding the inter arrival time to the time of the previous page request).

Object size: Size in bytes of a single requested object.

Object per page: Number of objects contained in a page.

Two HTTP loads were chosen, as shown in table 4.3. The first type, high browsing, simulated 1000 byte page size and images of medium size with an inter request time of 60. The second type used was low browsing, with page size of 500 bytes and small images with an inter request time 720.

4.1.3 Database application load types:

Database load types	Database high load	Database low load
Transaction mix (Queries/Total transaction)	100%	100%
Transaction interval time	exponential(12)	exponential(30)
Transaction size	constant(32768)	constant(16)
Type of service	Best effort	Best effort

Table 4.4 Database loads used in the simulation

Transaction Mix: The percentage of database query transactions of the total number of transactions. The remaining percent of the transactions are database entry transactions.

Transaction Interval time (sec): Defines when the next database transaction will start.

The start time of the next database transaction is computed by adding the inter arrival time to the time at which the previous database transaction completed.

Transaction size: Defines the size in bytes of the database transaction request.

Type of service: Represents a session attribute which allows packets to be processed faster in IP queues. It is an integer between 0 and 252, 252 being the highest priority. To assignment at the client is not affected by the ToS value specified at the server.

Transaction size: Is the size in bytes of the database transaction request.

Transaction interval: Defines when the next database transaction will start.

The third type of traffic load is the database load as shown in table 4.4. Database high load, uses transactions with a percentage 100% byte of queries to the total transactions, each transaction has a size of 32768 byte. Database low load uses a transactions percentage 100% of queries to the total transactions; each has a size of 16 byte

4.2 Analyzing results

We have evaluated two key performance metrics for three different applications using reactive and proactive protocols: delay and throughput. They are considered important factors which affect the behaviors of network communication.

We measured the number of control packets that have been sent by the source to the destination, and how much time it takes to reach the destination. This gives us an indication of how protocol efficiency acts, a parameter called end to end delay. For example, the number of packets that reach a destination in 3 seconds is not like the number of packets that reaches its destination in 10 seconds, if we considered million and millions of packets sent and received in simulation scenario. Secondly, we checked the number of packets that has been sent and the number of packets that have been received. This parameter is called throughput and is considered an accurate choice to measure performance of a network.

4.2.1 Simulation workspace

When designing a simulation model design we assigned more than one entity in network workspace, as shown in figure 4.1: application configuration, profile configuration, mobility configuration, server and nodes.

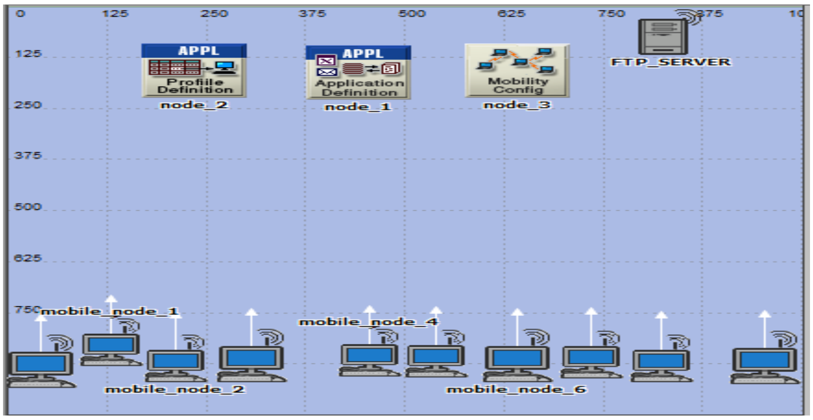


Figure 4.0 Example of the design model for AODV, DSR, and OLSR 10 node using FTP high load

Application configuration is used to assign the application that will be used such as FTP, HTTP, email, database, etc. Each application is defined in a profile configuration. In this simulation, we used FTP, HTTP and database.

Profile configuration is used to assign application traffic that will be used, such as FTP high load, FTP low load, etc. Each profile can be defined to more than one node, and in this simulation, we have used: high FTP load, low FTP load, high HTTP load, low HTTP load, high database load and low database load.

Mobility configuration defines which mobility model nodes will used in the simulation network. They control parameters such as the way nodes will move, the speed, etc. In this thesis, we have chosen 10 meters/sec, 30 meters/sec and random mobility model.

4.3 Analysis

In order to evaluate FTP, HTTP and database traffic using AODV, DSR, OLSR protocols, we executed a simulation of 54 scenarios divided into two groups. The first group is created of 48 scenarios, using various loads. Sizes of network varied between 10, 50 and 100 nodes, with speeds of 10 m/s. The second group consisted of 6 scenarios set to examine FTP, HTTP and database traffic using high and low loads with a medium size of network (50 nodes) and at a speed of 30 m/s. The second group was set with the same traffic loads and protocols of the first group but with different speeds of nodes, to observe if increasing nodes' speed will affect the performance of FTP, HTTP and database traffic. In this simulation, we used delay and throughput as a performance key measurement.

4.3.1 Simulation part one

The first part of the simulation was measured using FTP, HTTP and database, high and low load, with 10, 50, 100 nodes and a speed of 10 m/s, as discussed below.

A. <u>FTP Analysis Results:</u>

Since FTP is considered a wireless node that gives services, when a node demands to transfer files between a node and an FTP server in an adhoc network, it will try to find the shortest path between the source and the FTP server by using route request messages and route reply messages according to the procedure of the protocol that have been set, either AODV or DSR or OLSR. After finding the shortest path, a control connection will be established for sending and receiving commands that will allow establishing a data connection for uploading and downloading data.

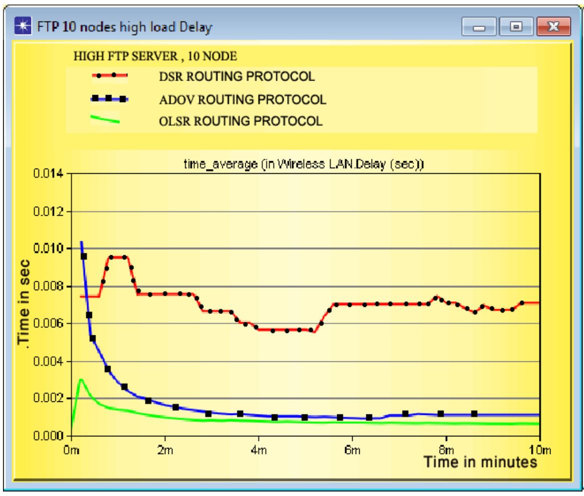


Figure 4.1 Delay AODV, DSR and OLSR 10 node using FTP high load

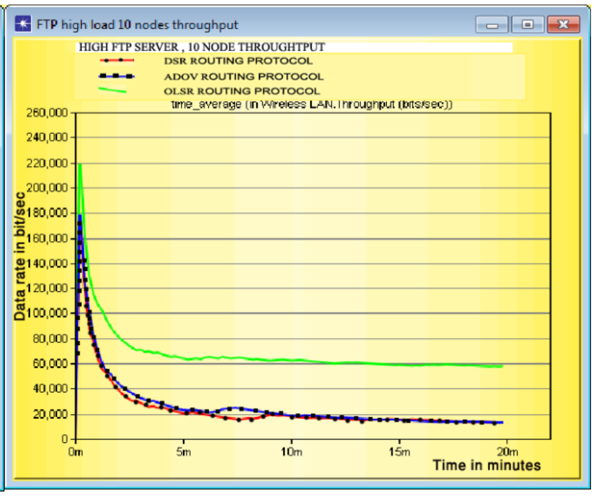


Figure 4.2 Throughput AODV, DSR and OLSR 10 node using FTP high load

In figure 4.1, we can observe FTP high load traffic simulation delay results using AODV, DSR, and OLSR protocols for 10 nodes and a speed 10 m/s. The x-axis denotes time in minutes and the y-axis in seconds. Results show a very high delay when using DSR protocol; it also shows unstable behavior, in which it increases and decreases around the point 0.007078 sec. This behavior is due to DSR mechanism nature in carrying the whole path along the network which makes DSR routing packet larger than others. On the other hand DSR exhibited a large routing overhead packet, while using AODV protocol showed a better result than DSR, in which it became stable at 0.0010763 sec. AODV protocol shares many characteristics with DSR, but it keeps information about the next hop in each node routing table, which makes delay less especially in the discovery process. When using OLSR protocol, it divides the network into groups called Multi point relay that contain a table about each node in its group, which reduces delay when the discovery process begins. In our simulation it, showed a constant and very low delay at 0.00064892 sec.

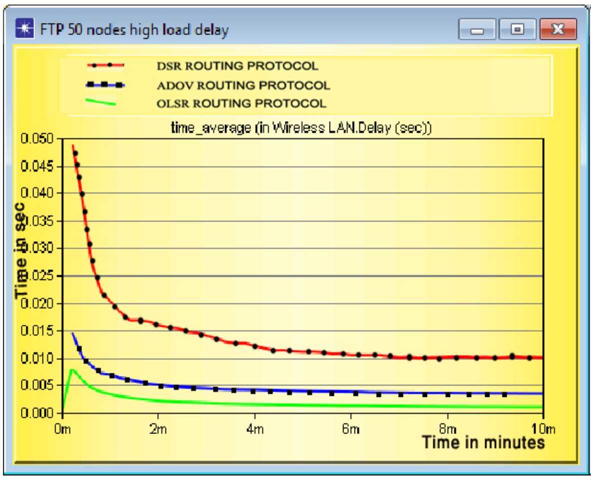


Figure 4.3 Delay AODV DSR and OLSR 50 node using FTP high load

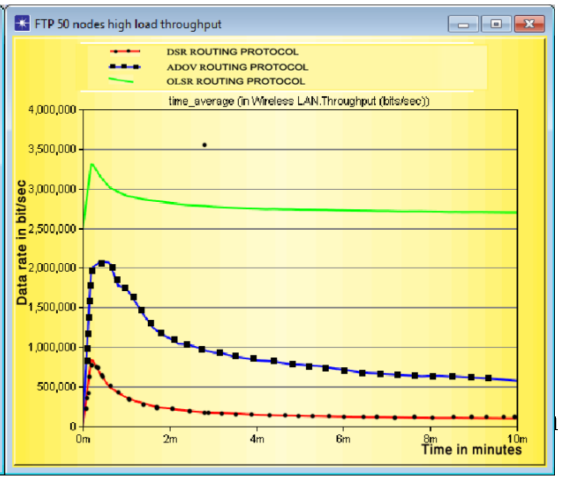


Figure 4.4 Throughput AODV DSR and OLSR 50 node using FTP high load

throughput results of FTP high load traffic metrics in small network size (10 nodes), and a speed of 10 m/s using AODV, DSR, OLSR protocols. The graph is shown in the time average form: the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results show that OLSR routing protocol gained the highest performance results between AODV and DSR routing protocols due to its neighbor sensing and flooding mechanism. As we can observe in the first seconds of the simulation, OLSR increases to reach 200,000 bit/sec and then decreases to be stable at 626,110.946 bit/sec. Since AODV and DSR share many characteristic but differ in the discovery process, in which AODV uses hop by hop, while DSR uses source routing, we can observe a small difference between AODV and DSR throughput results or even a similar result in the simulation at 19,000 bit/sec.

A second scenario was created for FTP high traffic load using AODV, DSR, and OLSR routing protocols and a speed of 10 m/s. In this scenario, we increased the number of nodes to 50 nodes to check if increasing size of the network will affect the performance of the network. The scenario model was executed for 10 minutes.

Figure 4.3 shows a delay simulation result for FTP high traffic load using AODV, DSR, OLSR routing protocols with 50 nodes and a speed of 10 m/s. When using the DSR protocol, it shows a more aggressive increment in delay than other protocols, then it gradually decreases to 0.010 sec and remain constant. When using the AODV protocol, we observed not a very high delay, but a higher delay than the OLSR protocol, in which it becomes stable at 0.003518 sec which is not far away from OLSR protocol that becomes stable at 0.0010584 sec's. DSR and AODV reactive protocols showed a higher delay than proactive routing protocols due to their mechanism in broadcasting a route request message to the whole network and waiting until a response message returned with the destination address. The OLSR routing protocol showed a constant delay, since OLSR depends on a routing table that uses routes saved in its table, which will lead to a lower latency.

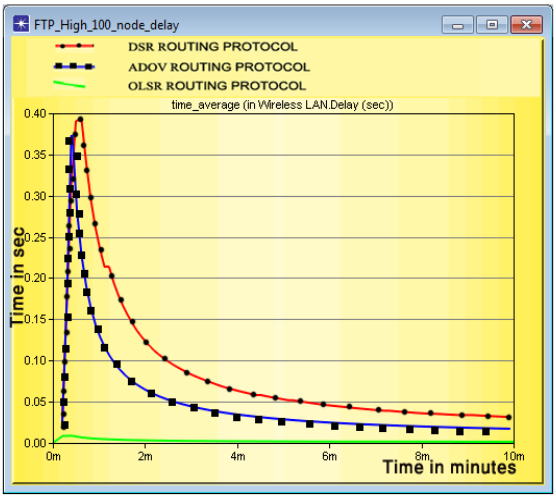


Figure 4.5 Delay AODV, DSR and OLSR 100 node using FTP high load

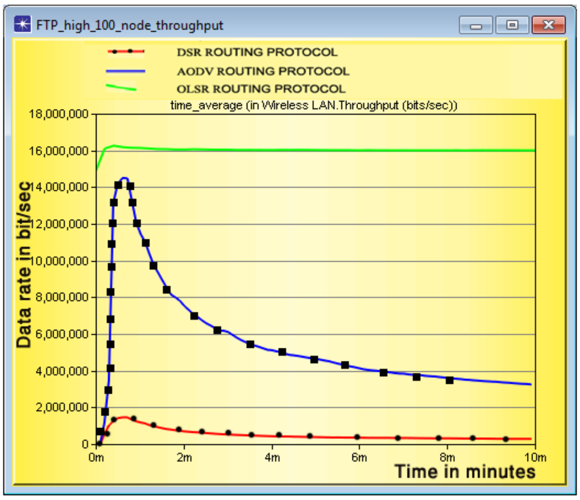


Figure 4.6 Throughput AODV, DSR and OLSR 100 node using FTP high load

We have checked throughput results of FTP high traffic using AODV, DSR, and OLSR routing protocols when increasing number of nodes to 50 nodes and a speed of 10 m/s. As shown in figure 4.4, the graph is shown in the time average form, the x-axis represents time in minutes and the y-axis data rate in bit/sec. We found a huge gap between proactive, reactive results, due to OLSR characteristic in periodic updating information between other nodes in the network.

OLSR acted in a very good way, and became stable at around 2,970,000 bit/sec, which is a very high performance. When using DSR, we observed a very low throughput result when compared with other protocols becoming constant at 268,474 bit/sec. AODV acted in a better way: it initially increased to a point more than 2000,000 bit/sec, then decreased to be around 581,853.4 bit/sec.

A third scenario was created for FTP high traffic load using AODV, DSR, and OLSR routing protocols and a speed of 10 m/s, but in this scenario we checked the performance of the network on large network size (100 nodes,) with the same parameters as the first and the second scenarios.

When checking high FTP load delay for big network sizes using AODV, DSR, and OLSR routing protocols at a speed of 10 m/s, we found that OLSR delay decreases when increasing number of nodes, because of its nature of dividing the network into groups, leading to a low latency. Delay behaves in an opposite way when using reactive routing protocols, in which it increases with increasing number of nodes. As shown in figure 4.5, when using AODV, protocol delay becomes constant around 0.017335 sec, while when using DSR protocol delay, results become constant around 0.031731 sec’s.

We also checked high FTP load throughput results for big network size using AODV, DSR, and OLSR routing protocols at a speed of 10 m/s, as shown in figure 4.6. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. We found that when increasing the number of nodes, OLSR routing protocol throughput increases to reach 16,001,374 bit/sec which is very high. While when using DSR routing protocol we observed a very low throughput result to reach 282,202.9 bit/sec. When using the AODV protocol, it showed a moderate result, becoming constant around 3,273,90.5 bit/sec.

We can conclude that FTP high load is best used with OLSR routing protocol, in which it showed the best performance between other protocols in small, medium and large network size. When using AODV routing protocol, we observed lower results in smaller network sizes. On the other hand we could observe acceptable results in medium and large network size. For DSR routing protocol, we saw a very poor results in all size of network.

FTP Low Load using 10m/s

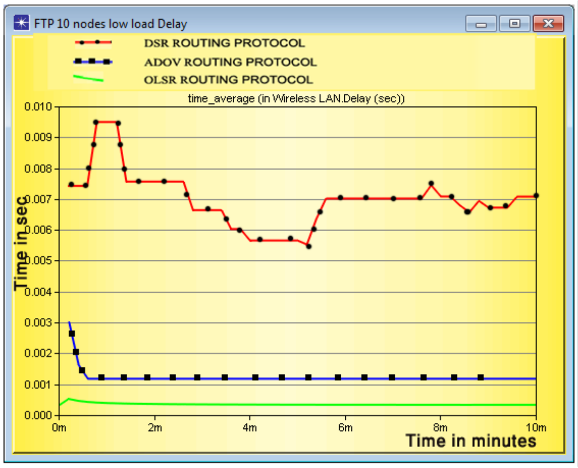


Figure 4.7 Delay AODV, DSR and OLSR 10 node using FTP low load

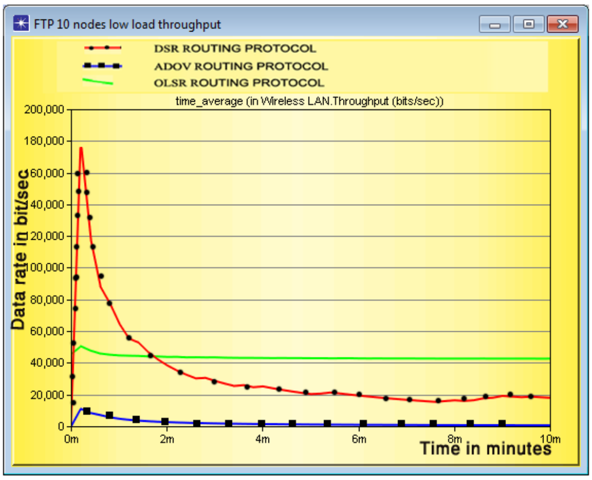


Figure 4.8 Throughput AODV, DSR and OLSR 10 node using FTP low load

Another scenario was set for this study, using the same parameters with a lower load of FTP traffic server with a file size 1000 bytes. We checked FTP server low load performance, delay and throughput when using AODV, DSR and OLSR protocols with a node speed 10 m/s on small, medium and big network sizes.

Figure 4.7 shows delay results of FTP low load traffic using AODV, DSR and OLSR protocols on small network size (10 nodes), and a speed 10 m/s. In this graph, the x-axis denotes time in minutes and the y-axis in seconds. We could observe a high and instability delay when using DSR routing protocol around 0.007 sec's, due to its discovery procedure which carries the whole path through the entire network, leading to a huge packet size and large routing overhead in the payload of the packets. When using the AODV routing protocol we saw a lower delay, yet still a higher delay than the OLSR routing protocol. This is due to its nature in broadcasting routing requests to the entire network and waiting until route reply messages arrived. As shown in the figure, it initially decreased around 0.0019 sec, then remains constant. OLSR showed the shortest delay between the protocols, because of its independent nature of the traffic and network density.

Figure 4.8 shows throughput result of FTP low load traffic using AODV, DSR, and OLSR protocols on small network size (10 nodes) and a speed of 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. We observed a gradual increase in the first seconds when using DSR routing protocol, which then decreased to be constant at 20,000 bit/sec. when using the OLSR routing protocol we observed the highest throughput results and the highest stability between other protocols, at 43,000 bit/sec. On the other hand, it showed a very low throughput result when using AODV routing protocol.

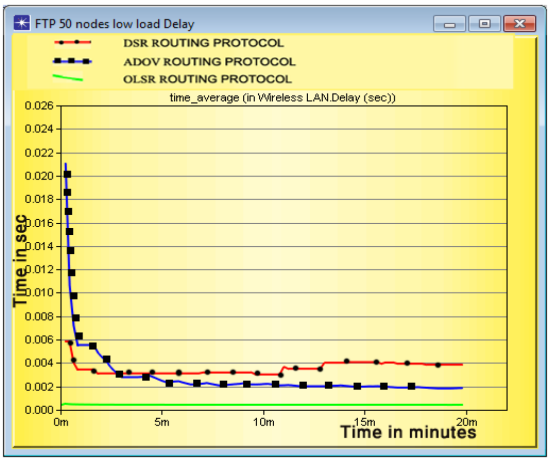


Figure 4.9 Delay AODV, DSR and OLSR 50 node using FTP low load

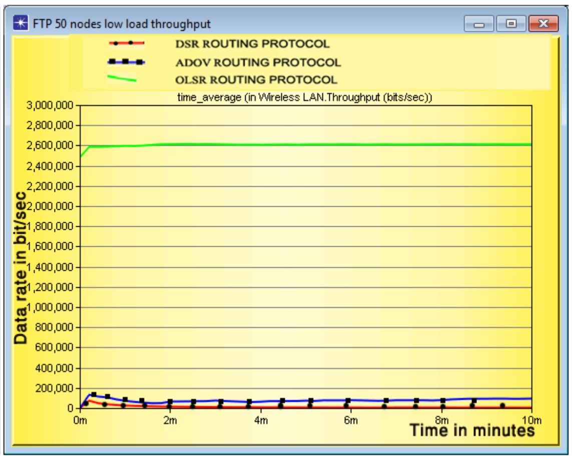


Figure 4.10 Throughput AODV, DSR and OLSR 50 node using FTP low load

A second scenario was set to measure the performance delay and throughput of FTP server low load with increasing the number of nodes to 50, to see if FTP server low load using AODV, DSR and OLSR routing protocols will be affected when increasing network size. We used the same parameters as the first scenario AODV, DSR and OLSR routing protocols, and node speed 10 m/s.

Figure 4.9 shows delay results of FTP server low load for 50 nodes, where the x-axis denotes time in minutes and the y-axis in seconds. We can observe a constant delay when using OLSR routing protocol due to its nature in up to date maintenance and collecting information on the network, which causes low latency.

When using AODV routing protocol it shows an acceptable delay, as shown in the simulation results, which gradually decrease to 0.002 sec. It then remains constant. When using DSR routing protocol, we could observe a very high delay, when compared with other protocols at 0.004 sec.

When measuring throughput results of FTP server low load we could observe a huge gap between the reactive and proactive routing protocols, in which OLSR routing protocol showed a very good throughput result. It remained constant at 2,500,000 bit/sec, due to its procedure in updating information about the network and the participating nodes, while AODV and DSR had far lower results at around 100,000 bit/sec, as shown in figure 4.10.

A third scenario was set for FTP low load using AODV, DSR and OLSR routing protocols and nodes speed 10 m/s, but with a huge network size consisting of 100 nodes, to observe how increasing number of nodes to huge network size will affect the performance of FTP low load traffic.

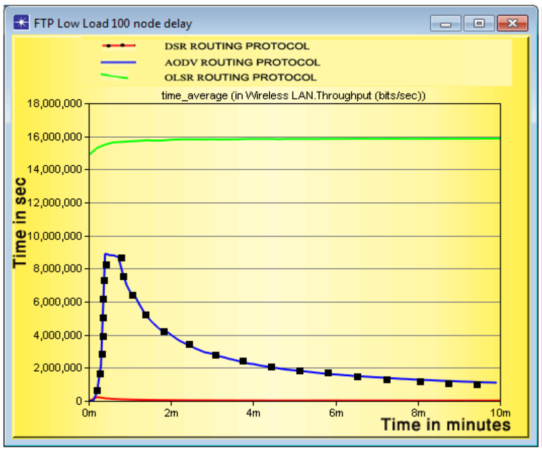


Figure 4.12 Throughput AODV DSR &OLSR 100 node using FTP low load

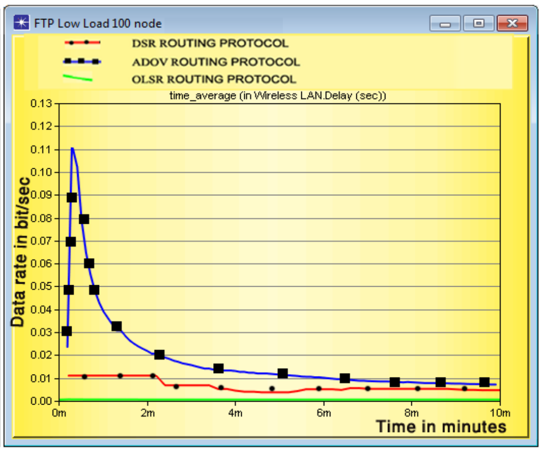


Figure 4.11 Delay AODV DSR &OLSR 100 node using FTP low load

Figure 4.11 shows delay results of FTP server low load using AODV, DSR and OLSR routing protocols and nodes speed 10 m/s on a big network size (consisting of 100 nodes.) The x-axis denotes time in minutes and the y-axis in seconds, we can observe delay behavior when using AODV protocol, in which it gradually begins increasing at the beginning of the simulation to reach 0.11 sec. It then decreases in a sharp way and continues until it reaches 0.007 sec. This behavior is caused by AODV protocol's procedure in finding the destination by sending route request message and continues waiting until a response message returns. It still has a higher delay when compared with the DSR routing protocol, which gained a constant delay at 0.004 sec. DSR behaved in a better way because it may have found in its cache a valid path to the destination, which caused a lower latency in finding the shortest path to the destination. OLSR Routing protocol had the lowest delay results, which remained constant at 0.00056 sec, since OLSR depends on a routing table that uses routes saved in its table, we believe that this would lead to a lower latency.

Figure 4.12 shows throughput results for FTP low load using AODV, DSR and OLSR routing protocols and a nodes speed 10 m/s on big network size. The graph is shown in the time average form, the x-axis represents time in minutes and the y-axis data rate in bit/sec. Results show a big difference between proactive and reactive routing protocol, in which when using OLSR protocol, throughput results remained constant at 16,000,000 bit/sec. When using AODV protocol at the beginning of the simulation, it increased to reach 8,000,000 bit/sec, and gradually decreased to 2,000,000 bit/sec and remained constant thereafter. DSR showed the lowest throughput results.

We can conclude when measuring performance for FTP low load in small, medium and large network size, the OLSR protocol showed the shortest delay and the highest throughput between other protocols, while AODV and DSR showed a very high delay and very low throughput performance

B. <u>HTTP analysis results:</u>

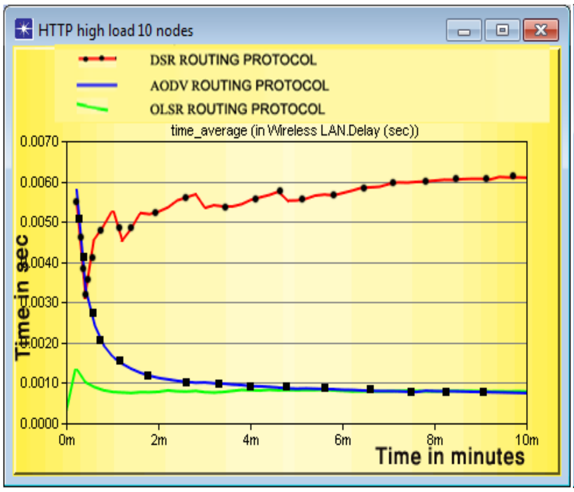


Figure 4.13 Delay AODV DSR &OLSR 10 node using HTTP High load

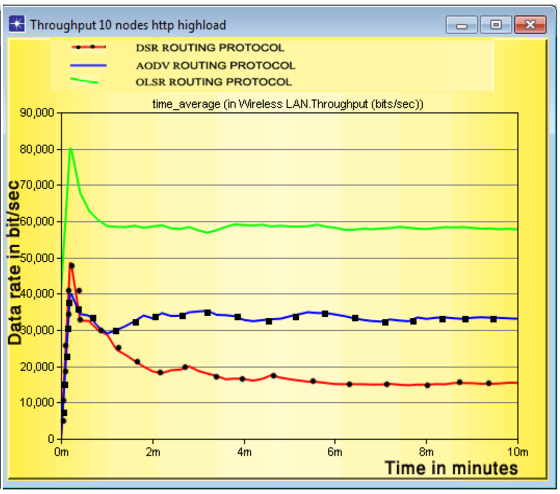


Figure 4.14 Throughput AODV DSR &OLSR 10 node using HTTP High load

When a node demands to enter to a website in an adhoc network, it will first try to find the shortest path between the source and the http server, (the destination which contains the requested site) by using OLSR, AODV, or DSR protocols. This sends a route request message, and replies by route reply messages to the certain the destination position. After finding the shortest path, one TCP connection will be established by sending an http request message to the server. The server, in turn, will send an http response message which contains in its body the request of the node, and the connection will remain opened until a time out occurs or the client will close the connection.

In our study, we used two types of HTTP loads: high load with an object size of 1000 bytes, and low load with an object size 500 bytes. These loads were examined in various size of networks (10, 50 and 100 nodes) using AODV, DSR and OLSR protocols and node speed 10 m/s. After simulation setup was designed, the simulations were for 10 minutes and then results were collected.

- ❖ **<u>HTTP high load using 10m/s</u>**

In figure 4.13 we can observe delay results of HTTP high load in a small network size with a node speed 10 m/s, using AODV, DSR and OLSR routing protocols. The graph is in time average form, where the x-axis denotes time in seconds and y-axis in minutes. HTTP high load appears to show a very high delay when using DSR protocol, compared to the other protocols. This is likely due to its mechanism of finding the shortest path and carrying the whole path along the network. At the beginning of the simulation, DSR gradually increases to reach 0.0061 sec. This is caused by the process of finding the shortest path to the destination, after which it appears to remain constant. At this stage, one connection is established and a source request, the HTTP page is delivered. HTTP high load showed better delay results when using AODV and OLSR protocols. At the beginning of the simulation, AODV showed a high delay, due to its mechanism in finding the shortest path. It then gradually decreases, reaching a point of 0.0010 sec, to meet OLSR protocol results, after which the two protocols results remain constant. In this stage of the simulation, the connections are established and the request is delivered. OLSR, at the beginning of the simulation, showed a very high delay result of 0.0059 sec. This high delay is caused by the OLSR protocol's nature of gathering and sharing information between its neighbors and deciding Multi point relays (MPR). OLSR then gradually dropped to 0.0010 in sec and remains constant thereafter.

In figure 4.14, we show throughput results of HTTP high load in a small network size (10 nodes) using AODV, DSR and OLSR routing protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes, and the y-axis data rate in bit/sec. We can observe very high throughput results for http high load. When using OLSR routing protocol in the first seconds of the simulation, it initially increases to reach a data rate of 80,000 bit/sec, then gradually decreased to 60,000 bit/sec and remains constant. This result is scored as the best result compared to the other protocol's scores. While the AODV protocol also showed good results, but still less than OLSR, in which it stabilizes at 35,000 bit / sec. When using this protocol, it shows a very low result, in which it becomes stable at 18,000 bit/sec—the lowest result compared to the other protocols.

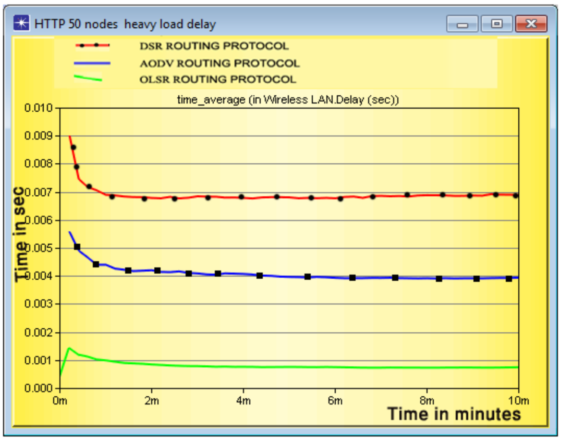


Figure 4.15 Delay AODV DSR &OLSR 50 node using HTTP high load

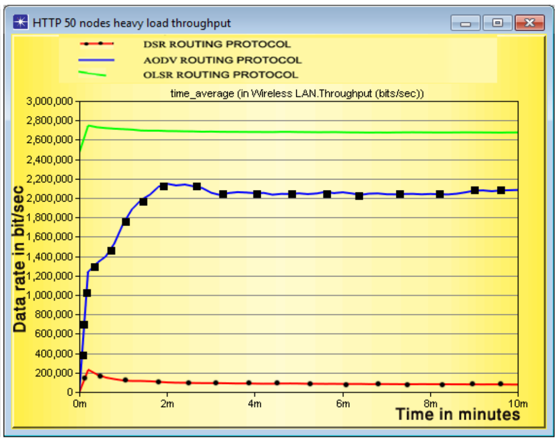


Figure 4.16 Throughput AODV DSR &OLSR 50 node using HTTP high load

We increased number of nodes to 50 node to check if delay and throughput behaviors of High HTTP loads using DSR, AODV and OLSR protocols with a node speed 10 m/s will be affected when a bigger network size exists.

In figure 4.15 we analyzed delay results for high HTTP load under 50 nodes using DSR, AODV and OLSR protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. OLSR shows an approximately constant delay of around 0.001 sec, while AODV has a higher delay than OLSR but still lower than DSR. AODV initially decreased to 0.004 sec, after which it remained constant. DSR gained the highest delay between other protocols, in which it gradually decreased to 0.007 sec and remained constant. This high result is due to the DSR mechanism of carrying the whole path along the entire network. When comparing DSR delay results with 10 node networks and 50 node networks, we don’t observe a huge difference. DSR using 10 nodes was stable around 0.0060 sec, while with 50 nodes, it was stable around 0.007 sec. When we compare delay results for AODV we find that it gained around 0.0010 sec when using 10 nodes and 0.004 sec when using 50 nodes, while OLSR gained the same delay results when using 10 and 50 nodes.

In figure 4.16 we analyzed throughput result for high HTTP loads using DSR. AODV and OLSR protocols with a node speed 10 m/s and network size of 50 nodes. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. OLSR showed a very high throughput result because of its mechanism in updating information about its neighbors and dividing the network, in the first seconds of the simulation, it increased to 2,800,000 bit/sec and it then remains constant. While AODV also showed a very good throughput, in the first two minutes of the simulation, it increased from zero to 2,100,000 bit/sec and then it decreased to 2,050,000 bit/sec and remains constant. DSR showed very low results.

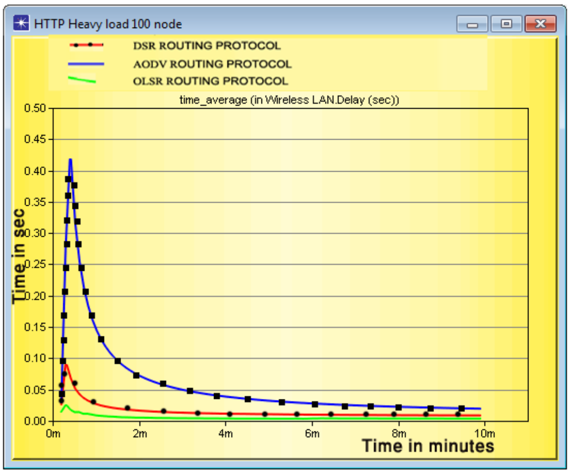


Figure 4.17 Delay AODV DSR &OLSR 100 node using HTTP high load

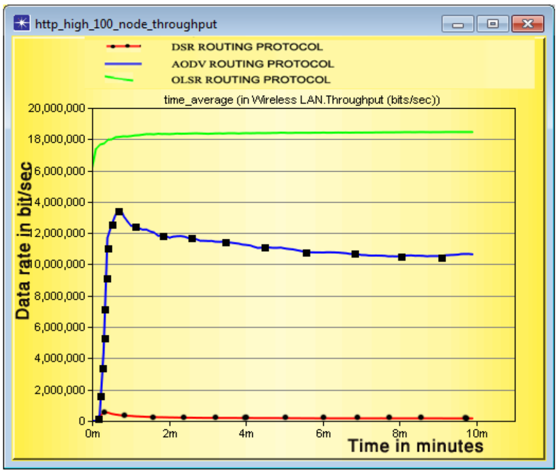


Figure 4.18 Throughput AODV DSR &OLSR 100 node using HTTP high load

We increased number of nodes to 100 nodes to see if delay and throughput behavior of high HTTP loads using DSR, AODV and OLSR protocols with a node speed 10 m/s will be affected when using a very big network size.

Figure 4.17 shows the delay results for HTTP high load using AODV, DSR, and OLSR routing protocols in a large network size. The graph is shown in the time average form; here the x-axis denotes time in seconds and y-axis in minutes. Results show that AODV has the highest delay when compared to the other protocols. During the first seconds of the simulation, it increased to 0.45 sec, then gradually decreased to reach 0.04 sec and remains constant. OLSR and DSR showed lower delays, in which their results were quite similar. DSR decreased to 0.3 sec and remained constant, while OLSR remained constant at 0.2 sec.

In figure 4.18, we display throughput results for high HTTP load using AODV, DSR and OLSR routing protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results when using OLSR protocol show the highest results between other protocols. In the first seconds of the simulation, it increases to 18,500,000 bit/sec, due to its mechanism of sharing information and dividing the network to multi relay points which help in sending and receiving data. DSR had the lowest results, in which it remained constant at 158690.88 bit/sec. AODV protocol began the simulation with low results due to the protocol's procedure in finding the shortest path to the destination. It then increased to remain in the middle with a moderate result at 10,664,850 bit/sec.

We can thus conclude the performance of HTTP high load in a small, medium and large network sizes, from delay and throughput measurements. When using the OLSR protocol, it showed very good results. When using AODV protocol it behaved the same way as OLSR protocol, but didn't gain the best results, but still good ones. We observed that when increasing the number of nodes, OLSR gains an even higher performance.

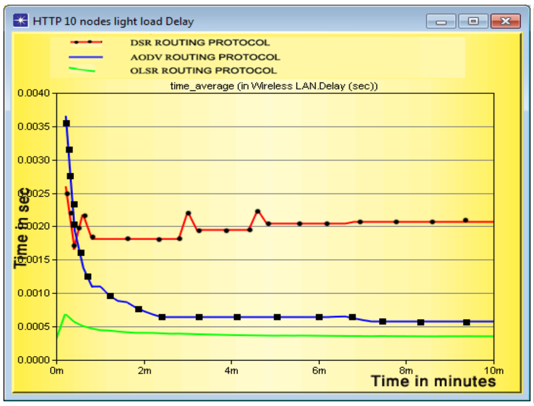


Figure 4.19 Delay AODV DSR &OLSR 10 node using HTTP low load

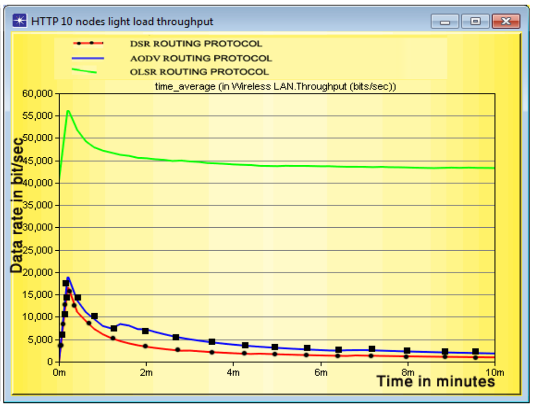


Figure 4.20 Throughput AODV DSR &OLSR 10 node using HTTP low load

❖ <u>HTTP Low Load using 10m/s</u>

Another scenario has been created: HTTP server with low browsing load traffic, an object size of 500, to analyze performance, throughput and delay using AODV, DSR and OLSR under 10, 50, 100 nodes with a node speed 10 m/s.

Figure 4.19 shows the delay results of HTTP low for a small network size, using AODV, DSR and OLSR routing protocols. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes.

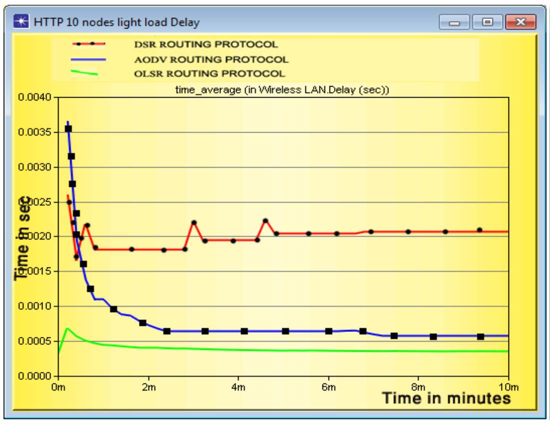


Figure 4.21 Delay AODV DSR &OLSR 50 node using HTTP low load

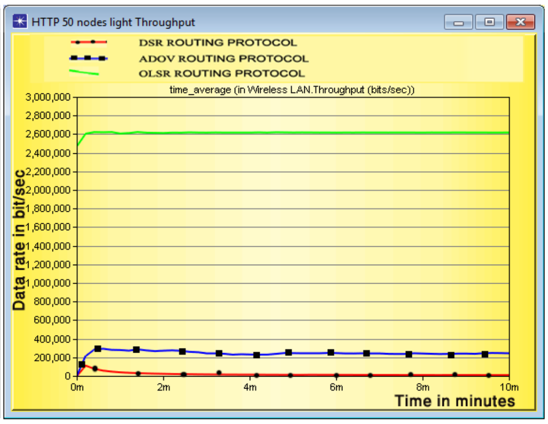


Figure 4.22 Throughput AODV DSR &OLSR 50 node using HTTP low load

The OLSR protocol shows a constant and low delay around 0.00035 sec, likely due to its mechanism of dividing the network to smaller networks, which conserves time in finding the shortest path. When using DSR protocol, we observed a very high latency compared to the other protocols, of around 0.002 sec. AODV protocol showed a lower delay than DSR routing, of around 0.0005 sec. At the beginning of the simulation it gradually decreased to reach a point in the middle, between DSR and OLSR routing protocols after which it remains constant.

Figure 4.20 shows throughput results for low HTTP load using AODV, DSR and OLSR routing protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results showed OLSR protocol gained the highest results when compared to the other protocols. At the first seconds of the simulation it increases to 56,000 bit/sec and became constant around 43,000 bit/sec. This is due to its mechanism of sharing information and dividing the network to multi relay points, which help in sending and receiving data. When using DSR and AODV protocols we observed very low results, in which they remained constant around 9,000 bit/sec.

A second scenario was set to measure the performance delay and throughput of HTTP server, low load, with increasing the number of nodes to 50, to check if the HTTP server using AODV, DSR, and OLSR routing protocols will be affected when increasing network size. We used the same parameters as the first scenario: AODV, DSR and OLSR routing protocols with node speed of 10 m/s

Figure 4.21 shows delay results for low HTTP load (fewer than 50 nodes) using DSR, AODV and OLSR protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. OLSR shows an approximate constant delay of around 0.0003 sec, while DSR had a higher delay than OLSR but still a lower result than AODV, while DSR initially decreased to reach 0.0005 sec and then remained constant. AODV gained the highest delay between other protocols, in which it gradually decreased to 0.005 sec and remained constant.

Figure 4.22 shows throughput results for low HTTP load using AODV, DSR and OLSR routing protocols with a node speed of 10 m/s. The graph is shown in time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results for the OLSR protocol show it gained the highest results compared to other protocols. At the first few seconds of the simulation, it increases to 2,800,000 bit/sec, likely due to its mechanism of sharing information and dividing the network to multi relay points, which help in sending and receiving data. DSR had the lowest results, which remained constant at 9,000 bit/sec. The AODV protocol showed low results, in which it remained constant around 200,000 bit/sec, in finding the shortest path.

A third scenario was set for low HTTP loads using DSR, AODV and OLSR protocols with a node speed of 10 m/s. We increased number of nodes to 100 to check if delay and throughput behavior of low HTTP loads will be affected when using a very big network size.

Delay results returned for HTTP low load using AODV, DSR and OLSR routing protocols in a large network size. The graph is shown in the time average form, where the x-axis denotes time in seconds and the y-axis in minutes. Results show that AODV has the highest delay compared to the other protocols. In the first seconds of the simulation, it increases to 0.009 sec, while OLSR and DSR had a lower delay, in which DSR decreased to 0.003 sec and remained constant, while OLSR remained constant at 0.0005 sec.

Throughput results for low HTTP load using AODV, DSR and OLSR routing protocols with a node speed of 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results for OLSR protocol revealed the highest results compared to other protocols. In the first seconds of the simulation, it increased to 18,400,000 bit/sec, due to its mechanism of sharing information and dividing the network to multi relay points, which help in sending and receiving data. DSR had the lowest results, which remained constant at 26,300 bit/sec. AODV protocol began the simulation with moderate results, due to the protocol's procedure in finding the shortest path to the destination. It then increased to remain in the middle with a low result at 183,664 bit/sec.

We can conclude when measuring performance for HTTP low load in small, medium and large network sizes, the OLSR protocol showed the lowest delay and highest throughput compared to other protocols. AODV and DSR showed a very high delay and very low throughput performances.

C. Database analysis results:

Another scenario was created for database traffic for 10, 50, and 100 nodes using AODV, DSR and OLSR, using high load with a transaction size of 32768 bytes and low load with a transaction size of constant value 16 bytes.

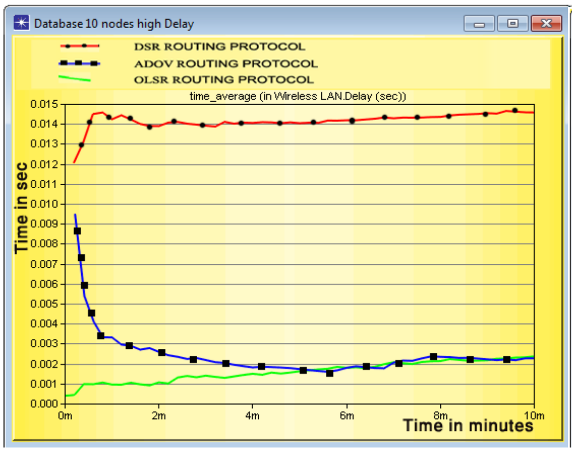


Figure 4.23 Delay AODV DSR &OLSR 10 node using DB high load

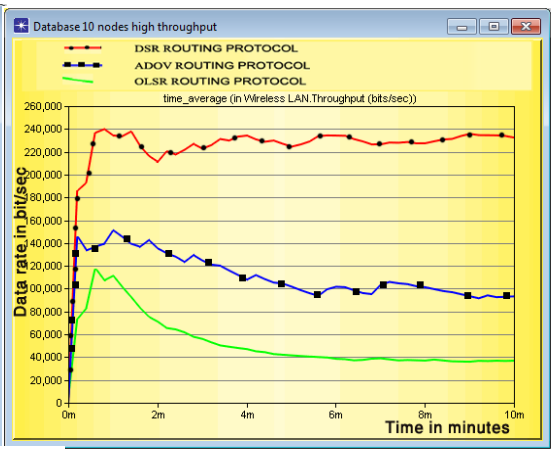


Figure 4.24 Throughput AODV DSR &OLSR 10 node using DB high load

Database is considered a wireless node that provides services. When a node demands to fetch a transaction, select, update, and delete a query between a node and a Database server in an adhoc network, it will try to find the shortest path between the source and the Database server by using route request messages and route reply messages, using the procedure of the protocols that have been set, either AODV or DSR or OLSR. After finding the shortest path, a control connection will be established for sending and receiving commands, which will allow a data connection to be established for the required data transactions.

❖ **Database high load using 10m/s**

Figure 4.23 shows delay results for database with high load using AODV, DSR and OLSR routing protocols in a small network size (10 nodes). The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. Results from the DSR protocol show a more aggressive increment in delay than other protocols, to reach 0.014 sec. OLSR and AODV protocols show a lower delay than DSR, due to its mechanism in broadcasting a route request message to the whole network and waiting until a response message returns with the destination address. The OLSR routing protocol shows a constant delay, since OLSR depends on a routing table that uses routes saved in this table, which will lead to a lower latency. AODV begins with a high delay, and then decreases to 0.02 sec, where it joins the OLSR results.

Figure 4.24 shows throughput results for high database load using AODV, DSR and OLSR routing protocols with a node speed of 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results shows a very rate when using the DSR routing protocol, which gained the highest results, at around 240,000 bit/sec. AODV protocol also showed high throughput result, at around 100,000 bit /sec, while OLSR showed the

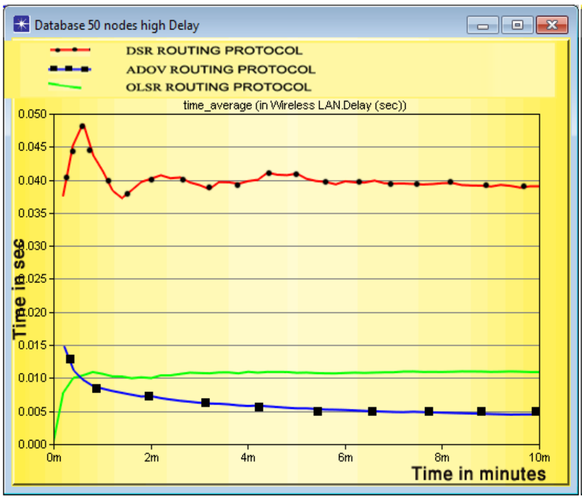


Figure 4.25 Throughput AODV DSR &OLSR 50 node using DB high load

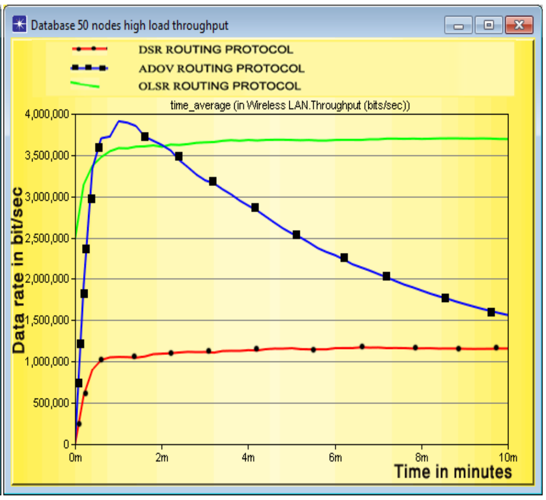


Figure 4.26 Delay AODV DSR &OLSR 50 node using DB high load

lowest result, between the other protocols, at 40,000 bit/sec.

A second scenario was set, to measure the performance delay and throughput of database servers, high load, with increasing the number of nodes to 50, to check if database server, high load, using AODV, DSR and OLSR routing protocols would be affected when increasing network size. We used the same parameters as in the first scenario with AODV, DSR and OLSR routing protocols and node speed 10 m/s.

Figure 4.25 shows delay results for database high load using AODV, DSR and OLSR routing protocols in small network size (50 nodes.) The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. AODV results showed the lowest delay, at around 0.005 sec, while the OLSR protocol also showed a low result, but higher than OLSR, at around 0.010 sec. DSR showed the highest delay due to its mechanism in carrying the whole path through the network at 0.40 sec.

Figure 4.26 shows throughput results for high database load using AODV, DSR and

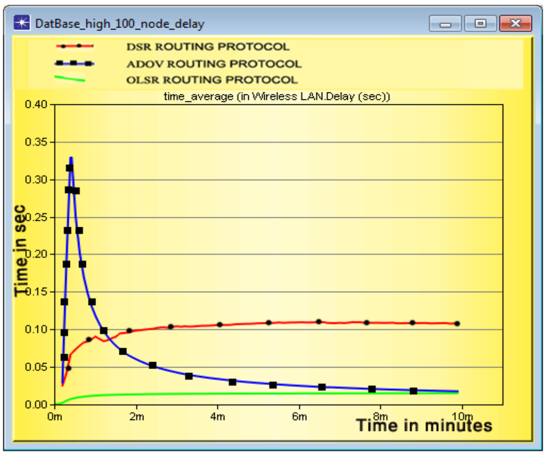


Figure 4.27 Delay AODV DSR &OLSR 100 node using DB high load

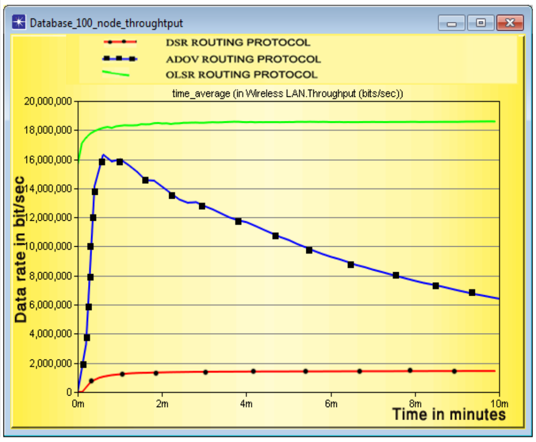


Figure 4.28 Throughput AODV DSR &OLSR 100 node using DB high load

OLSR routing protocols with a node speed of 10 m/s for 50 nodes. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. OLSR protocol shows a very high result, at around 3,600,000 bit/sec, while AODV gradually increased to reach an even higher result than OLSR, at around 3,800,000 bit/sec, then it decreased to reach 1,800,000 bit/sec, near the DSR results, which gained the worst results of all the protocols, at around 1,100,000 bit/sec..

A third scenario was set, to measure the performance delay and throughput of a database server high load with increasing the number of nodes to a big network size (100 nodes,) to see if database server high load using AODV, DSR and OLSR routing protocols would be affected. We used the same parameters as the first scenarios with AODV, DSR and OLSR routing protocols and node speed of 10 m/s.

Figure 4.27 shows the delay results of database high results for big network size using AODV, DSR and OLSR routing protocols. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. Results show, when using OLSR protocol, a very low delay of around 0.01 sec. This is likely caused by its nature in up to date maintenance and collecting information on the network, which leads to a low latency.

The AODV routing protocol also showed a low delay result. At the beginning of the simulation, delay increased to 0.032 sec, as it broadcasts a route request message to the whole network and waits until a response message returns with the destination address. It then decreased to a point very similar to that of OLSR, at around 0.01 sec. DSR showed a very high delay performance.

Figure 4.28 shows throughput results using database high load results for big network size using AODV, DSR and OLSR routing protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. When using the OLSR routing protocol, we observed a very high result, in which it remained constant around 18,000,000 bit /sec. This high result is caused by OLSR's process in dividing the network into smaller networks by the multi relay point. When using the AODV routing protocol, it gradually increased to a very high result at around 17,000,000 bit/sec, then decreased to 7,000,000 bit/sec. This decrease is caused by AODV's procedure. DSR routing protocol gained very low results, at 1,000,000 bit/sec when compared with other protocols.

We can conclude for small, medium and large network sizes, database high load using AODV and OLSR showed a good delay performance. When using DSR routing protocol we saw a high delay result, and when measuring throughput results, it shows very low results. When using the AODV routing protocol, we observed good results, while OLSR showed is the best performance when compared with the other protocols.

- ❖ **Database low load using 10m/s**

Another scenario was created to analyze database server, but with low load traffic, using AODV, DSR and OLSR routing protocols in small, medium and large networks with a node speed 10 m/s.

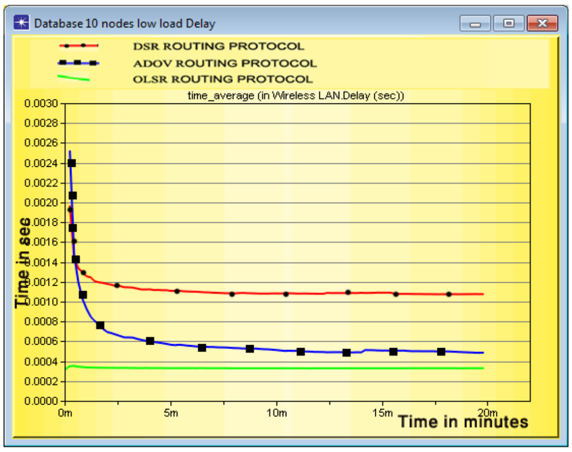


Figure 4.29 Delay AODV DSR &OLSR 10 node using DB low load

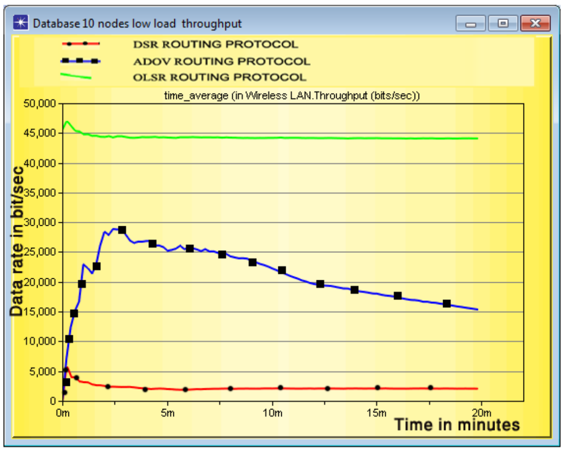


Figure 4.30Throughput AODV DSR &OLSR 10 node using DB low load

Figure 4.29 shows delay results of database low load for small network sizes using AODV, DSR and OLSR routing protocols, with a node speed 10 m/s. When using DSR protocol, we saw a very high delay of around 0.011 sec, due to its procedure of finding the shortest path.

OLSR showed the lowest delay compared to the other protocols at around 0.0003 sec. This is since OLSR depends on a routing table, which will lead to a lower latency. When using the AODV routing protocol, we observed a high delay which remained between the other protocols at 0.0005 sec. A At the beginning of the simulation, AODV and DSR routing protocols reached a very high delay which then decreased, a behavior caused by the protocol's procedure in finding the shortest path.

Figure 4.30 shows throughput results of database low load in small network sizes using AODV, DSR and OLSR routing protocols with a node speed 10 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. OLSR routing protocol data shows a very high result at around

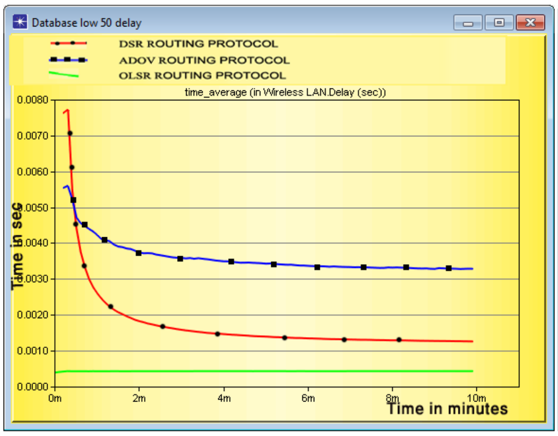


Figure 4.31 Delay AODV DSR &OLSR 50 node using DB low load

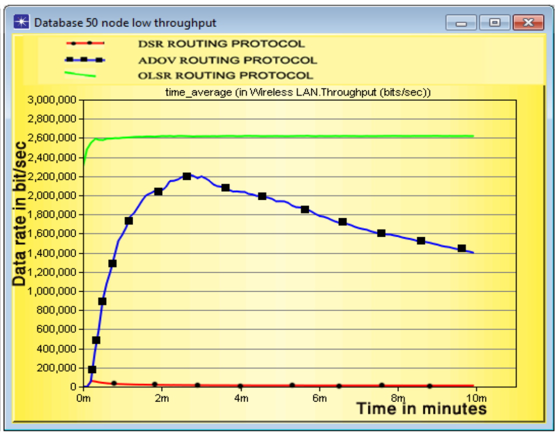


Figure 4.32 Throughput AODV DSR &OLSR 50 node using DB low load

45,000 bit/sec. AODV routing protocol gradually increased to reach 28,000 bit/sec, then decreased to reach an acceptable point at 16,000 bit/sec. DSR results gained 2,500 bit/sec, the worst results among the other protocols.

A second scenario was created for database low traffic load using AODV, DSR, OLSR routing protocols and a speed of 10 m/s, but in this scenario we increased the number of nodes to 50, to check if doing so will affect the performance of the network. The scenario model was executed for 10 minutes.

Figure 4.31 shows delay results of database server low load for 50 nodes, where the x-axis denotes time in minutes and the y-axis in seconds. We can observe a constant delay of around 0.005 sec. When using the OLSR routing protocol, due to its nature in up to date maintenance and collecting information of the network, we could observe low latency.

AODV routing protocol showed the highest delay when compared with other protocols. As shown in the simulation results, it gradually decreased to 0.035 sec, and then remained constant. DSR routing protocol showed an acceptable delay when compared with OLSR routing protocol's delay performance which began with a very high delay then decreased to 0.0015 sec.

Figure 4.32 shows simulation throughput results for database low load traffic metrics in medium network sizes, consisting of 50 nodes, and a speed of 10 m/s, using AODV, DSR, OLSR protocols. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput data showed that the OLSR routing protocol gained the highest performance results when compared with the AODV and DSR routing protocols, reaching 2,600,000 bit/sec. This is most likely to do with its neighbor sensing and flooding mechanism. AODV showed a good performance, in which it increased to 2,000,000 bit/sec, then decreased to 1,400,000 bit/sec and remained constant. The DSR routing protocol showed a very low throughput performance around 50,000 bit /sec.

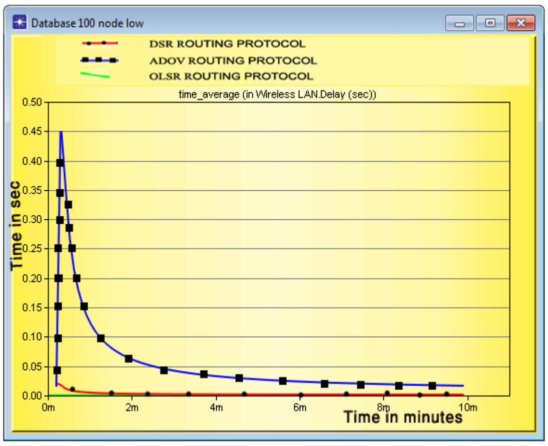


Figure 4.33 Delay AODV DSR &OLSR 100 node using DB low load

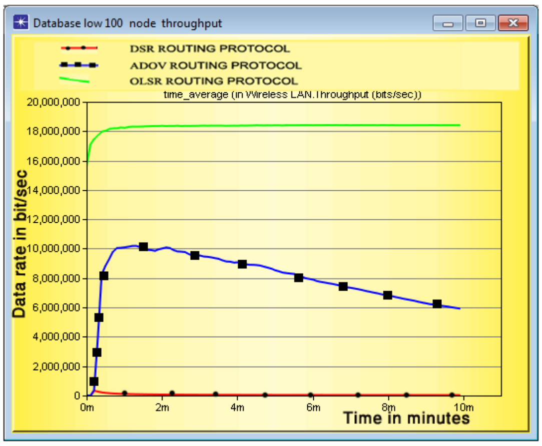


Figure 4.34 Throughput AODV DSR &OLSR 100 node using DB low load

A third scenario was set for database low load using AODV, DSR and OLSR routing protocols and node speed of 10 m/s, but with a big network size consisting of 100 nodes, to observe how increasing number of nodes to huge network size will affect the performance of database low load traffic.

Figure 4.33 shows delay results of database server low load using AODV, DSR and OLSR routing protocols and node speed of 10 m/s on a big network size, consisting of

100 nodes. The x-axis denotes time in minutes and the y-axis in seconds. We can observe the delay behavior when using AODV protocol, in which it gradually begin increasing at the beginning of the simulation to reach 0.45 sec, then it decreases sharply and continues decreasing until it reaches 0.02 sec. This behavior is caused by AODV protocol's procedure in finding the destination by sending route request message and keep waiting until a response message returns, but it still exhibited a higher delay when compared with DSR routing protocol, which gained a constant delay at 0.0001 sec. DSR behaved in a better way, because it may have found in its cache a valid path to the destination, which caused a lower latency. The OLSR routing protocol showed a very low delay results in which it remained constant at 0.0001 sec, since OLSR depends on a routing table that uses routes saved in its table; this will lead to a lower latency.

Figure 4.34 shows throughput result for database low load using AODV, DSR and OLSR routing protocols and a nodes speed of 10 m/s on a big network size. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Results show a big difference between proactive and reactive routing protocols, in which when using the OLSR protocol, throughput results remained constant at 18,000,000 bit/sec, while when using AODV protocol at the beginning of the simulation it increases to reach 10,000,000 bit/sec then gradually decreased to 6,000,000 bit/sec and remained constant. DSR showed the lowest throughput result.

We can conclude when measuring performance for database low load in small, medium and large network sizes, the OLSR protocol showed the lowest delay and the highest throughput compared to the other protocols. AODV and DSR protocols showed a very high delay, and when checking throughput performance, AODV showed a good performance. Finally, when using DSR routing protocol we observed very low results.

4.3.2 Simulation part two

In the second part of the our study we measured throughput and delay performance of FTP, HTTP and database traffic with high and low loads in medium network size (consisting of 50 nodes). We used the same parameters as in simulation part one, but we increased speed nodes from 10 m/s to 30 m/s in 1000 *1000 m.

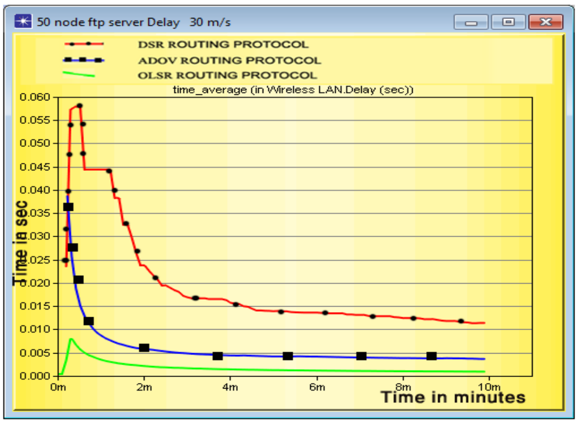


Figure 4.35 Delay AODV DSR &OLSR 50 node using FTP low load

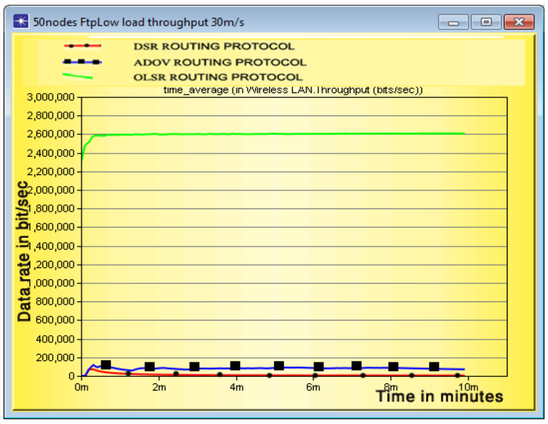


Figure 4.36 Throughput AODV DSR &OLSR 50 node using FTP low load

Figure 4.35, shows delay results of FTP server low load using AODV, DSR and OLSR routing protocols and a nodes speed 30 m/s on a medium network size (consisting of 50 nodes). The x-axis denotes time in minutes and the y-axis in seconds, and we can observe a constant delay when using OLSR routing protocol due to its nature in up to date maintenance and collecting information of the network, which causes low latency.

The AODV routing protocol shows an acceptable delay. As shown in the simulation results, it gradually decreased to 0.003 sec, and then remains constant. When using 10 m/s, delay performance for AODV was constant around 0.002 sec. When using DSR routing protocol in 10 m/s and 30 m/s we observed a very high delay when compared with other protocols at 0.004 sec.

When measuring throughput results of FTP server low load using AODV, DSR and OLSR routing protocols and a nodes speed 30 m/s on a medium network size (consisting of 50 nodes), we observed a huge gap between the reactive and proactive routing protocols. When using OLSR routing protocol, we saw a very good throughput results, it having remained constant at 2,600,000 bit/sec, likely associated with its procedure in updating information about the network and the participating nodes while AODV and DSR had far lower results, at around 100,000 bit/sec, as shown in figure 4.36.

We conclude there appears to be no difference in performance when increasing speed node to 30 m/s of FTP low load traffic in medium network size when using AODV, DSR and OLSR routing protocols.

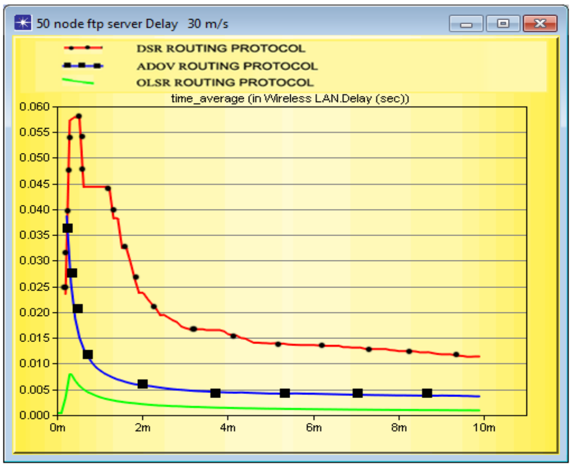


Figure 4.37 Delay AODV DSR &OLSR 50 node using FTP high load

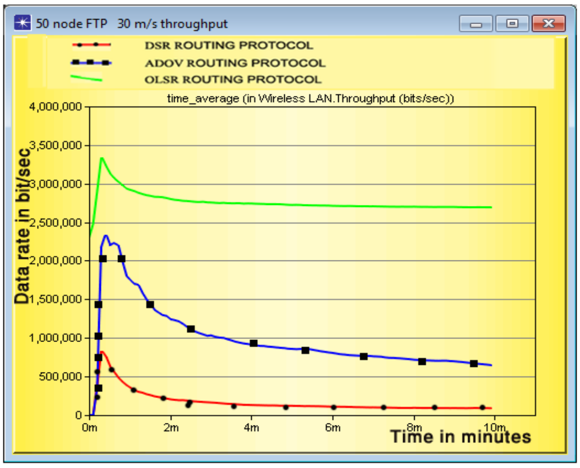


Figure 4.38 Throughput AODV DSR &OLSR 50 node using FTP high load

Figure 4.37 shows delay simulation results for FTP high traffic load using AODV, DSR, OLSR routing protocols with 50 nodes and a speed of 30 m/s. When using the DSR protocol, we observed an aggressive increment in delay than other protocols which then gradually decreased to 0.012 sec and remained constant. When using the AODV protocol, we saw a moderate delay, higher than OLSR, in which it became stable at 0.004 sec, which is not far off from the OLSR protocol which becomes stable at 0.001 sec. DSR and AODV reactive protocols show higher delays than proactive routing protocols, due to their mechanism in broadcasting a route request message to the whole network and waiting until a response message returns with the destination address. The OLSR routing protocol shows a constant delay, since OLSR depends on a routing table which uses routes saved in its table; this will lead to a lower latency.

We checked throughput results of FTP high traffic using AODV, DSR, OLSR routing protocols when increasing number of nodes to 50, and a speed of 30 m/s. We have shown our results in figure 4.38, the graph of which uses time average form where the x-axis represents time in minutes and the y-axis data rate in bit/sec. We found a huge gap between proactive and reactive results, due to OLSR protocol's characteristic periodic updating of information between other nodes in the network. OLSR acted in a very positive way, becoming stable around 2,800,000 bit/sec, while DSR showed a very low throughput result when compared with other protocols. DSR became constant at 200,000 bit/sec, while AODV acted in a better way. It initially increased to a point greater than 2000,000 bit/sec, then decreased to around 250,000 bit/sec When comparing performance results we couldn't find a significant difference in delay results.

When we compared throughput results, we found no difference in DSR and OLSR protocols, while AODV decreased from 581,853.4 to 250,000 bit/sec.

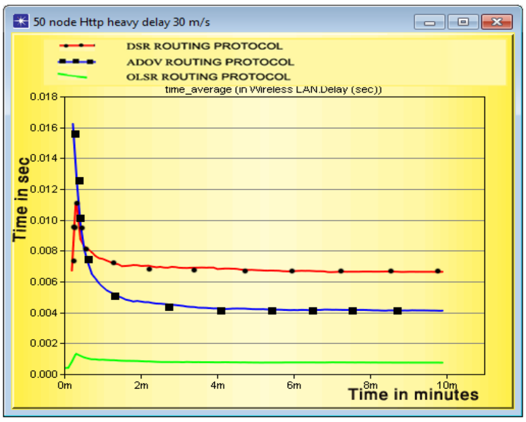


Figure 4.39 Delay of AODV DSR &OLSR 50 node using HTTP high load

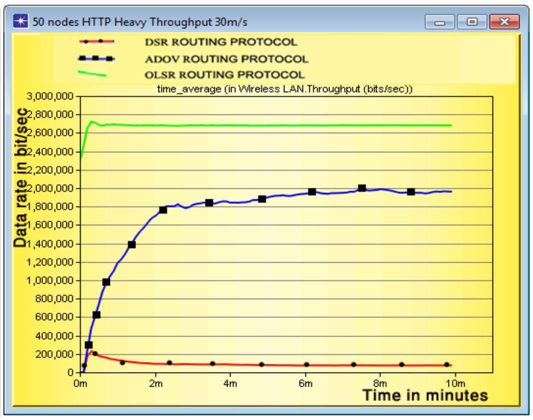


Figure 4.40 Throughput of AODV DSR &OLSR 50 node using HTTP high load

In figure 4.39, we analyzed delay results for high HTTP load with 50 nodes using DSR, AODV and OLSR protocols with a node speed 30 m/s. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. OLSR shows an approximately constant delay around 0.001 sec, while AODV showed a higher delay than OLSR, but still lower than DSR. AODV initially decreased to 0.004 sec and then remained constant. DSR gained the highest delay compared to other protocols in which it gradually decreased to 0.007 sec and remained constant. This high result is due to DSR's mechanism in carrying the whole path along the entire network. When comparing DSR delay results with 10 nodes and 50 nodes, we didn't find a huge difference, in which DSR using 10 nodes stabilized around 0.0060 sec, while when using 50 nodes it was stable around 0.007 sec. When we compared delay results for AODV, we found that it gained around 0.0010 sec when using 10 nodes and 0.004 sec when using 50 nodes. OLSR showed the same delay results when using 10 and 50 nodes.

In figure 4.40 we analyze throughput results for high HTTP loads using DSR, AODV and OLSR protocols with a node speed of 30 m/s, and a network size of 50 nodes. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. OLSR showed very high throughput results because of its mechanism in updating information about its neighbors and dividing the network. In the first seconds of the simulation, it increased to 2,700,000 bit/sec and then remained constant. AODV also had a very good throughput; in the first two minutes of the simulation, it increased from zero to 2,100,000 bit/sec, then decreased to 2,050,000 bit/sec and remained constant. DSR showed very weak results, in which it increased to 100,000 bit/sec and remained constant.

We couldn’t find any differences when we measured high HTTP loads throughput and delay performance using DSR, AODV and OLSR protocols with a node speed 30 m/s and 10 m/s in a medium network size, consisting of 50 nodes.

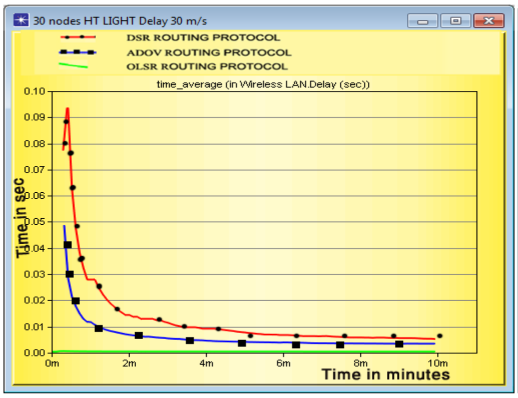


Figure 4.41 Delay of AODV DSR &OLSR 50 node using HTTP low load

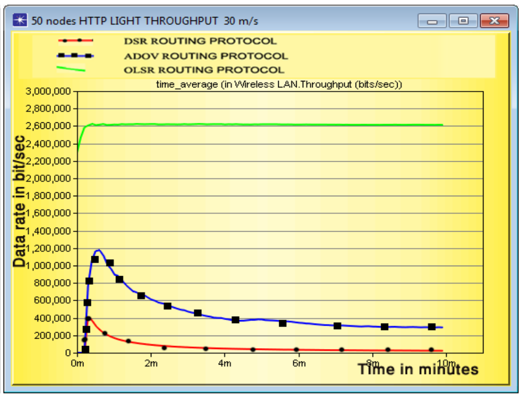
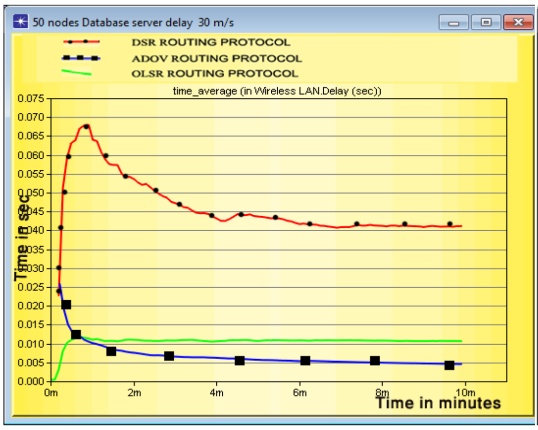


Figure 4.42 Throughput of AODV DSR &OLSR 50 node using HTTP low load

In figure 4.41 we analyze delay results for low HTTP low load with 50 nodes using DSR, AODV and OLSR protocols, with a node speed 30 m/s. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. OLSR shows an approximately constant delay around 0.0003 sec, while DSR had a higher delay than OLSR but still showed lower results than AODV. DSR initially decreased to 0.0005 sec, then remained constant. AODV gained the highest delay compared to other protocols, in which it gradually decreased to 0.005 sec and remained constant thereafter. Figure 4.42 show throughput results for low load HTTP using AODV, DSR and OLSR routing protocols with a node speed 30 m/s. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput result for the OLSR protocol show it gained the highest results compared to the others. In the first seconds of the simulation, it increases to 2,600,000 bit/sec due to its mechanism of sharing information and dividing the network to multi relay points which help in sending and receiving data. DSR had the lowest results, in which it remained constant at 9,000 bit/sec. The AODV protocol showed lower results, in which it was constant at around 500,000 bit/sec in finding the shortest path. We couldn't find any difference in performance of HTTP low load when increasing node speed to 30 m/s.

When measuring HTTP low load using DSR, and OLSR protocols, we found identical results with node speeds of 30 m/s and 10 m/s in a medium network size. AODV throughput performance became better when increasing speed to 30 m/s, in which its constant value was 250,000 bit/sec for 10 m/s. When using 30 m/s, it was 500,000 bit/sec.



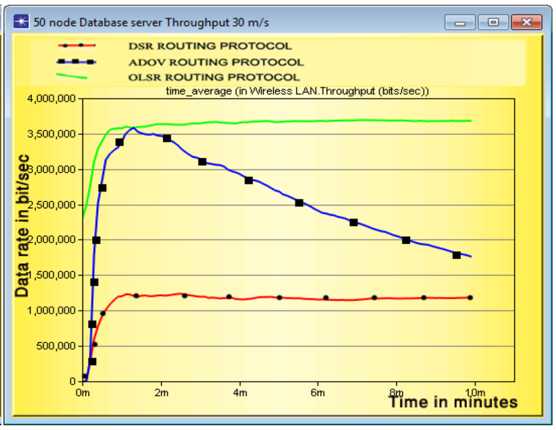


Figure 4.43 Delay of AODV DSR &OLSR 50 node using DB high load

Figure 4.44 Throughputs of AODV DSR &OLSR 50 node using DB high load

Figure 4.42 shows delay results of database high load using AODV, DSR and OLSR routing protocols with medium network sizes (consisting of 50 nodes) with node speed 30 m/s. The graph is shown in the time average form, where the x-axis denotes time in seconds and y-axis in minutes. Results shows AODV had the shortest delays of around 0.005 sec, while OLSR protocol showed a similarly low result, but higher than OLSR at around 0.010 sec. DSR showed the highest delay, due to it mechanism in carrying the whole path through the network 0.45 sec.

Figure 4.43 shows throughput results for high load database using AODV, DSR and OLSR routing protocols, with a node speed of 30 m/s for 50 nodes. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. OLSR protocol showed very high results, around 3,600,000 bit/sec. On the other hand, AODV gradually increased to reach an even higher result of around 3,800,000 bit/sec than OLSR; then it decreased to 1,800,000 bit/sec, near the DSR figures which gained the worst results at around 1,100,000 bit/sec.

We couldn't find any differences when comparing database high load using AODV, DSR and OLSR routing protocols in a medium network size, consisting of 50 nodes, with node speed 30 m/s or with a node speed of 10 m/s.

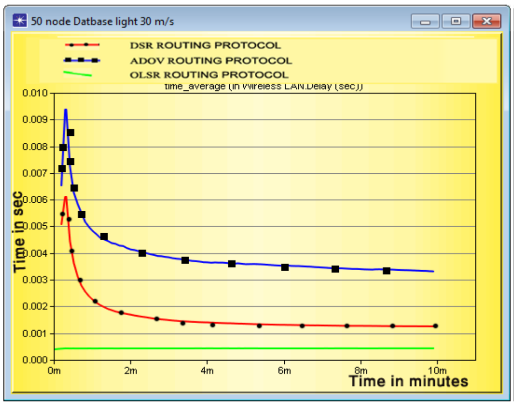


Figure 4.45 Delays of AODV DSR &OLSR 50 node using DB low load

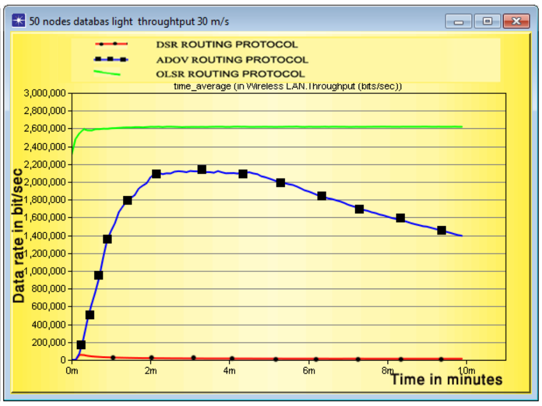


Figure 4.46 Throughputs of AODV DSR &OLSR 50 node using DB low load

Figure 4.45 shows delay results of database servers with low load for 50 nodes, with a node speed 30 m/s. The x-axis denotes time in minutes and the y-axis in seconds. We can observe a constant delay at around 0.005 sec, when using OLSR routing protocol. This is due to its nature in up to date maintenance and collecting information of the network, which causes low latency.

AODV routing protocol shows the highest delay when compared with other protocols. As shown in the simulation results, it gradually decreased to reach 0.035 sec and then remained constant. When using DSR routing protocol we observed an acceptable delay when compared with the OLSR routing protocol delay performance, which began with a very high delay which then decreased to 0.0015 sec.

Figure 4.46 showed simulation throughput results of database low load traffic metrics in medium network size consisting of 50 nodes and a speed of 30 m/s, using AODV, DSR, and OLSR protocols. The graph is shown in the time average form, where the x-axis represents time in minutes and the y-axis data rate in bit/sec. Throughput results show that the OLSR routing protocol demonstrated the highest performance results compared to AODV and DSR, having reached 2,600,000 bit/sec. This is due to its neighbor sensing and flooding mechanism. AODV showed a good performance in which it increases to 2,000,000 bit/sec, then decreased to 1,400,000 bit/sec and remained constant. The DSR routing protocol showed a very low throughput performance of around 50,000 bit /sec.

We couldn’t find any significant differences when comparing database low load using AODV, DSR and OLSR routing protocols in medium network size consisting of 50 nodes, with node speed 30 m/s and a node speed 10 m/s.

In conclusion, increasing node speed cannot affect FTP, HTTP, and database traffic throughput and delay performances in various 10, 50, and 100 node network sizes and at various loads.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this work, we performed an analytical study on MANET routing protocols AODV, DSR and OLSR. We measured delay and throughput using FTP, HTTP, and database application loads on small, medium and large network sizes with two different node speeds of 10 m/s and 30 m/s.

Our simulation results focused on analyzing each application FTP, HTTP, and database using each protocol at the same parameters, of a simulation work space 1000*1000 m and network sizes, data rates and simulation times.

We first analyzed FTP high load, and concluded that in small, medium and large network sizes, high delay is noticed in DSR routing protocols compared to AODV and OLSR. This is attributable to the nature of DSR in carrying the whole path along the network, which makes DSR routing packet larger than others. large routing overhead packets in the payload of the packets in DSR. OLSR was the best compared to the other protocols. In small and medium sized networks, AODV gained moderate end to end delay, but good throughput results.

When measuring the protocols using FTP low load, we found that DSR had the worst results in all sizes of network. AODV was similar to OLSR in small sizes, but in medium and large sized networks, AODV didn’t have good results like OLSR. When comparing results of FTP high and low load, the AODV protocol behaves better with FTP low load than FTP high load. In all numbers of nodes, delay appeared continuous, increasing in FTP high load more so than in low load; throughput showed better results in FTP low loads than high load.

We also observed increasing delays when the number of nodes increased. DSR Protocol in small networks had a higher delay when using FTP low load than FTP high load, while OLSR protocol was better with low load than high load when measuring for delay. Throughput gained better result with high loads. In general, OLSR had the best results compared to the others when using FTP high and low load. Results for http high load using various simulations for AODV, DSR and OLSR protocols at 588 sec, showed that when a node demands an http page, the source node will first find the shortest path from the source to the destination http, then one control connection will be opened between the source and http server. AODV gained the lowest results in end to end delay compared to other protocols, but not the best throughput results in smaller network sizes. In medium and large network sizes, OLSR had the best results, and DSR has the worst results between all protocols.

HTTP low load results, using various simulations for AODV, DSR and OLSR protocols at 588 sec are as follows. In all network sizes, OLSR gained the best results compared to other protocols, while DSR gained the worst results. AODV performed in a very good way in medium and large network sizes, while AODV results were very close to the OLSR results.

When comparing results of HTTP high and low load, AODV protocol had a higher delay with high pages, but also good throughput results. In small networks, we can observe that AODV has the highest throughput in HTTP high load. The DSR protocol had an increasing delay with high pages (due to DSR’s nature of carrying all paths), but

	ADOV			DSR	OLSR	
	Small	Medium	Large	Small Medium Large	Small Medium Large	
FTP High	Good	Good	Can	Weak	Best	
FTP Low	Good	Can	Can	Weak	Best	
HTTP high	Best	Good	Good	Weak	Good	Best
HTTP low	Can	Good	Good	Weak	Best	
DB high load	Good	Good	Good	Weak	Best	
DB low load	Good	Good	Good	Weak	Best	

Table 5.1 Final result for simulation study

Also good throughput results. OLSR protocol is better with low load than high load, when measuring delay. On the other hand, throughput gained better results with high loads.

Identical results were found for database low and high load using various simulations for AODV, DSR, OLSR protocols at 588 sec. OLSR performed best, while DSR returned the worst results compared with the other protocols, and AODV had good results.

When comparing results of database high and low load, the AODV protocol was found to have a higher delay with high loads, but also good throughput results. In small networks, observed that AODV had the highest throughput in DB high load.

DSR protocol had an increasing delay with DB transactions. OLSR protocol was better with DB load than high load; when measuring delay, throughput had gained better results with high load.

A second simulation was performed, using medium network sizes for 30 m/s nodes speed. AODV, DSR and OLSR routing protocols using, FTP, HTTP and Database applications with high and low loads.

Results show identical data even when nodes speed were increased to 30 m/s. OLSR had the best results, while DSR had the worst results between the protocols, and AODV gained moderate results.

5.2 Finally

In this thesis, we analyzed delay and throughput for proactive and reactive protocols with different numbers of nodes and different speed and modes. Results showed DSR has the worst results in throughput and delay, AODV didn't gain the best results in throughput but it didn't gain the worst results in delay. As a result, AODV can be used in small and medium network sizes. OLSR performed the best, suggesting it is the best choice to be used in large network sizes.

5.3 Future Work

Future work from this thesis could be conducted by focusing on protocols performances using different network conditions with the same applications, by changing the power capacities of nodes, mobility models and pause times. Another suggestion could be studying other applications using the same parameters of this study, to make a big image of applications and protocols performances, in efforts to continue to improving MANET's performance.

5.4 References:

[1] Das, S.,& Perkins, C.,& Royer, E. (Feb 2001). Performance Comparison of Two On-Demand Routing Protocols for Adhoc Networks. Personal Communications, IEEE. VOL. 8, 1, pp. 16 - 28. Tel Aviv, Israel: IEEE.

[2] Feeney, L. M.,& Ahlgren,B.,& Westerlund,A., & Dunkels, A. (2002). Spontnet: Experiences in Configuring and Securing Small Adhoc Networks. Networked Appliances, 2002. Liverpool. Proceedings. 2002 IEEE 5th International Workshop (pp. 102 - 106). IEEE.

[3] Zapata, M. (July 2002). Secure Adhoc On-Demand Distance Vector Routing. Vol 6 (Issue 3), 106-107.

[4] Mittal,P.,& Singh,P.,& Rani ,S. (2013). Performance Analysis Of AODV, OLSR, GRP and DSR Routing Protocols with DATABASE LOAD in MANET. IJRET: International Journal of Research in Engineering and Technology (pp. 2321-7308). IJRET.

[5] Gowrishankar, S.,& Basavaraju,.T.,& Singh,M.,& Sarkar,S. (November 18-19, 2007). Scenario Based Performance Analysis of AODV and OLSR in Mobile Adhoc Networks. Proc. of the 24th South East Asia Regional Computer Conference. Vol.15, pp. 8.1-8.6. Bangkok, Thailand: International Journal of the Computer.

[6] Gagangeet,A., & Sandeep, K. (March 2013). Comprehensive Evaluation of AODV, DSR, GRP, OLSR and TORA Routing Protocols with varying number of nodes and traffic applications over MANETs. In G. A, & S. K (Ed.), IOSR Journal of Computer Engineering. 9, pp. 54 -61. IOSR.

[7] Mwanza, J.,&Myirenda, B. (2009, January). Performance Evaluation of Routing Protocols in Mobile Adhoc Networks (MANETs).

[8] Baraković, S.&, Kasapović,S., Baraković, & Baraković, J. (2010). Comparison of MANET Routing Protocols in Different Traffic and Mobility Models. Tel for Journa, Vol. 1 (No. 1), 8-14.

[9] Vincent, R., Centibots: communication among robots. Personal Communication, University of Washington, Seattle, June 2004.

[10] Khatri,P.,& Rajput, M.,& Shastri, Al.,& Solanki, K. (2010). Performance Study of AdhocReactive Routing Protocols. Journal of Computer Science, vol.6 (10), 1130-1134.

[11] Shrestha,A.,& Tekiner,F. (February 2013). Investigation of MANET routing protocols for mobility and scalability. Proceedings on International Conference on

Recent Trends in Information Technology and Computer Science 2012 ICRTITCS , 9-14, New York, USA: Foundation of Computer Science.

[12] Rajeshkumar, G,&. Valluvan, K. (April 2013). A Comparative Study of Secure Intrusion-Detection Systems for Discovering Malicious Nodes on MANETs. International Journal of Computer Applications,Published by Foundation of Computer Science, Vol 67 (No.18), 1-5.

[13]Johnson, D, & Maltz,D, & Broch,J. (April 2003). DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Adhoc Networks. Proc, of the First ACM SIGCOMM Asia Workshop, (pp. 115-124). Asia.

[14] Modeler, O. (n.d.). OPNET Modeler 14.0, OPNET Modeler Documentation by OPNETTechnologies.Versions:14.Retrieved from http://www.OPNET.com/products/modeler/home.html

[15] Gamal, A.&, Mammen, J.&, Prabhakar, B.,& and Shah, D. (2004). Throughput-Delay Trade-off in Wireless Network. Proc of IEEE INFOCOM 2004. USA: IEEE.

[16] Boukerche, A., &,Das, S., & Fabbri, A. (2001). Analysis of randomized congestion control with DSDV routing in adhoc wireless networks. Journal of Parallel and Distributed Computing (JPDC), VOL 6, 967–995.

[17] http://www.comptechdoc.org/independent/networking/protocol/protlayers.html.

[18] Ramanathan, R., & Hain, R., An ad hoc wireless testbed for scalable,adaptive QoS support. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pages 998 – 1002, September 2000.

[19] Lee, S. (2000). Routing and Multicasting Strategies in Wireless Mobile Adhoc Networks. Los Angeles: University of California.

[20] Khokhar.L.& Guizani.M. (2008). A survey of Secure Mobile Adhoc Routing Protocols. in Proc. IEEE, 78-93.

[21] Camp,T, & Boleng, J, & Davies, V. (2002). A Survey of Mobility Models for Adhoc Network Research. in Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Adhoc Networking: Research, Trends and Applications, Vol. 2 (No. 5), 50-58.

[22] Clausen, T. (September 2013). The optimized link state routing protocol evaluating through experiments and simulation, mind pass center for distributed system, International Journal of Computer Science and Telecommunications, Vol 4 (No. 9), 120-126.

[23] Perkins, C.,& Royer, E. (1999). AdhocOn-Demand Distance Vector Routing. WMCSA '99 Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications (pp. 90-97). USA: IEEE Computer Society Washington, DC.

[24] Berndt, L. (2000). A Quick Guide to AODV Routing. USA: NIST National Institute of standard and Technology.

[25] Perkins, C, & Royer, E., & Das, S. (2003). Adhoc On-Demand Distance Vector (AODV) Routing. IEEE, 101-106.

[26] Dhulipala, S.,& Chandrasekaran,R, & Prabakaran, R.(May 2009). Timing Analysis and Repeatability Issues of Mobile AdhocNetworking Application traffics in Large Scale Scenarios. International Journal of Recent Trends in Engineering (IJRTE), Vol.1 (Issue 3), 463-467.

[27] Tafti, V.,& Gandomi A. (2010). Performance of QoS Parameters in MANET Application Traffics in Large Scale Scenarios. 2010 World Academy of Science, Engineering and Technology., 857-860.

[28] Khan, N, & Ahmed, R.,& Aziz, T. (December 2011). Analysis of the Interaction between Routing Protocols and MANET Parameters. International Journal of Computer Applications (0975 – 8887, Vol. 35 (Issue 2), 13-20.

[29] Gupta, M.,& Kaushik S. (May 2012). Performance Comparison Study of AODV, OLSR and TORA Routing Protocols for MANETS. International Journal Of Computational Engineering Research, 40-46.

[30] Parulpreet,S.,&, Ekta, B.,& Gurleen, W. (May 2012). Evaluation of various traffic loads in MANET with DSR routing protocol through use of OPNET Simulator. international Journal of Distributed and Parallel Systems (IJDPS), Vol. 3 (No. 3), 75-83.

[31] Forouzan, B. (2010 by). TCP/IP Protocol Suites Book (Fourth Edition ed.). The McGraw-Hill Companies, Inc.

[32] Jubin, J.,& Tornow, J. (January 1987). The DARPA of Packet Radio Network Protocol. IEEE, Vol.75, No.1, 21-32.

[33] Singh, P., & Barkhodia, E.,& Walia, G. (March 2012). Performance Study of Different Routing Protocols (OLSR,DSR, AODV) Under Different Traffic Loads and with Same Number of Nodes in MANET using OPNET. IEEE, Vol. 3 (No.1), 155-157.

[34] Nadia Q.,& Fatin, S.,& Hamid, A. (February, 2009). Mobile Adhoc Networking Protocol's Evaluation through Simulation for Quality of Service. IAENG International Journal of Computer Science, 51-54.

[35] Timcenko, V.&, Stojanovico, M., Rakas, S. (2009). MANET Routing Protocols vs. Mobility Models:Performance Analysis and Comparison, International Journal of Recent Trends in Engineering, 500-507.

[36] Hsu, J.,& Bhatia, S, & Takai, M.& Bagrodia, R. (2010). Performance of Mobile adhoc networking routing protocols in realistic scenarios. Culver: U.S Army CECOMRDEC.

[37] Broch, J.&, Maltz D.,& Johnson D.,& Chun Hu, Y., Jetcheva, J. (October 25–30, 1998). A Performance Comparison of Multi-Hop Wireless Adhoc Network Routing Protocols. Proceedings of the Fourth Annual ACM/IEEE International ,1-13 Dallas, Texas, USA.: ACM.

[38] Park, V, & Corson, M. (1998). A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing. Proceedings of ISCC ’98, 30 June - 2 July in Athens, Greece,1-7. Athens, Greece: IEEE.

[39] Jorg, D. (2003). Performance Comparison of MANET Routing Protocols in Different Network Sizes. Institute of Computer Science and Applied Mathematics, Computer Science. Switzerland: University of Berne, Switzerland.

[40] Clausen, T., Jacquet, P., Viennot, L. (Sep 2002). Comparative study of CBR and TCP performance of MANET routing protocol. IEEE, 1-8.

[41] Adam, G.,& Kapoulas, V.,& Bourasand, C.,& Kioumourtzis, G. (2011). Performance Evaluation of Routing Protocols for Multimedia Transmission Over Mobile Adhoc Network. IEEE, Technical program at IFIP WMNC', 2011, 1-6.

[42] Kumar, A. (, June 2008). Performance Comparison Study of Wireless Mobile Adhoc Network Routing Protocol. IJCSNS International Journal of Computer Science and Network Securit, Vol.8 (Issue.6).

[43] Mbarushimana, C. and Shahrabi, A. (May 2007). Comparative study of reactive and proactive routing protocols performance in mobile adhoc networks. Proc of the 21st International Conference on Advanced Information Networking and Applications Workshops, 679–684.

[44] Ijaz, M. (March, 2009). Transmission control Protocol (TCP) performance evaluation in MANET. BLEKINGE: Blekinge Institute of Technology.

[45] Usop, N., &Abdullah, A.,& Abidin A. (July 2009). Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment. IJCSNS International Journal of Computer Science and Network Security, VOL.9 (No.7), 50-56.

[46] Das, S. C. (October 1998). Comparative performance evaluation of routing protocols for mobile, adhoc networks. In 7th Int. Conf. on Computer Communications and Networks (IC3N), 153–161.

[47] Hoebeke, J.,& Moerman, I.,& Dhoedt, B.,& Demeester, P. (June,1993). An Overview of Mobile AdhocNetworks: Applications and Challenges. Belgium: Department of Information Technology(INTEC), University – IMEC vzw,.

[48] Simoneau, P. (2006). The OSI Model: Understanding the Seven Layers of Computer Networks. USA: G.

[49] Kaur,K.&Kaur,S.&Singh,V. (March,2014). Throughtput Analysis of Proactive and Reactive MANET Routing Protcols. International Journal Research in Managment &Technology, Vol.3 (Issue 3), 1-5.

[50] Ekta, & Gill,N. (June 2014). Manets Routing :Simulation Analysis for Aodvand Dsr. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3 (Issue 6), 6976-6982.

[51] Force., I. E. (n.d.). Mobile Ad-hoc Networks (MANET) - WORKGROUP. Retrieved from http://www.ietf.org/html.charters/manet-charter.html.

[52] Templin, F.,& Ogie,Rr., & Lewis, M. Technical Report (February 2004). TBRPF RFC3684, experimental edition.

[53] Maltz, D, & Broch, J, & Johnson,D., Experiences designing and building a multi-hop wireless ad hoc network testbed. Technical Report CMU-CS-99-116, School of Computer Science, Carnegie MellonUniversity, 1999.

[54] Nijem,S.,&Kafri,N.,(July 22-25, 2013). Comparison of Different Applications using Proactive, Reactive Manet Routing Protocols Under Different Application Modes, Nodes and Speed, Proceedings of The 2013 World Congress in Computer Science, Computer Engineering,and Applied Computing,164-169 , Las Vegas, Nevada, USA,.

http://weblidi.info.unlp.edu.ar/WorldComp2013-Mirror/p2013/MSV.html