Investigate Performance of IEEE 802.11b Using OPNET

Hayder M. Jasim

Submitted to the
Institute of Graduate Studies and Research
in partial fulfilment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Muhammed Salamah ______________________________________

2. Asst. Prof. Dr. Adnan Acan ______________________________________

3. Asst. Prof. Dr. Gürcü Öz ______________________________________

ABSTRACT

In recent years, the research community interest has increased about wireless ad hoc networks. IEEE 802.11b standard is one of the wireless network standards which is used in such network these groups' standards. The IEEE 802.11b standard defines the protocol and proper interconnections of data communication tools wireless local area network (LAN). It includes the physical and media access control (MAC) layers of the ISO seven-layer network model. Therefore, investigating its performance with different ad hoc network configurations is necessary.

Routing protocols like AODV, DSR, TORA and OLSR are famous studies that discourse the performance evaluation of routing protocols via different evaluation methods. Various methods and simulation environments grant different outcomes. Thus, there is a need to broaden the spectrum to account for effects not taken into consideration in a particular environment. There is a need to expand the spectrum to take into consideration the effects of file size, numbers of nodes and mobility that were neglected in a specific environment. During the time of the thesis, the performance of AODV, ad hoc routing protocol in OPNET was evaluated using several scenarios. Effectiveness of this approach with 802.11b standard in various wireless ad hoc network configurations was investigated. The simulation results were analyzed and compared.

In our simulation work, FTP traffic is used between a client and FTP server through other entire nodes with AODV protocol and 802.11b wireless standard. Our simulation results would also represent a situation where a MANET node can receive traffic from another node in the network. Furthermore, the nodes are randomly scattered in the network to provide the possibility of multi hop routes from the client to the server.

Keywords: Mobile Wireless ad Hoc Networks, Routing Protocol, Performance Evaluation, OPNIT simulator, AODV

ÖZ

Son yıllarda, araştırma topluluğunun ilgisi kablosuz özele amaca yönelik (ad hoc) ağlar hakkında artmıştır. IEEE 802.11b standardı, bu ağlarda kullanılan kablosuz ağ standartlarından biridir. IEEE 802.11b standardı very iletişim için gerekli kablosuz yerel alan ağında (LAN) protokolü ve uygun bağlantıları tanımlar. Standart, yedi katmanlı ISO ağ modelinde fiziksel ve ortam erişim denetimi (MAC) katmanlarını içerir. Bu nedenle, farklı ad hoc ağ yapılandırmalarında bu standartın performans araştırmak gereklidir.

AODV, DSR, TORA ve OLSR gibi yönlendirme protokollerinin farklı değerlendirme yöntemleri ile yönlendirme protokolleri performansın değerlendirme de kullanılan ünlü çalışmalardır. Çeşitli yönlendirme yöntemleri ve simülasyon ortamları farklı sonuçlar vermek. Bu yüzden performans ölçülerinde belirli bir ortam dikkate alınmak yerine farklı ortamlarda göz önünde bulundurmak gerekir. Farklı ortamlarda, dosya boyutu, düğüm sayısı ve düğümlerin hareketliliğinin etkilerini incelemeye gerek vardır. Bu tezde bilinen bir yönlerdirme protokolü olan AODVi, OPNET simulatörü kullanılarak farklı ağ senaryolarında değerlendirilmiştir. Çeşitli kablosuz ad hoc ağ yapılandırmalarında 802.11b standardının etkinliği araştırılmıştır. Elde edilen simülasyon sonuçları analiz edilip karşılaştırılmıştır.

Simülasyon çalışmalarında, AODV protokolü ve 802.11b kablosuz standart kullanarak işlemci ve sonucu düğümler arasında FTP trafiği oluşturulmuştur. Ayrıca simülasyonlarda kablosusun bir ad hoc güğümünün başka bir düğümün erişim alanında olmadığı durumlar da göz önüne alınmıştır. Ayrıca, düğümler ağda rasgele yerleştirilerde istemciden sunucuya çok hop yollarınoluşması sağlanmıştır.

Anahtar Kelimeler: Mobil Kablosuz Özel Amaca Yönekli Ağlar, Yönlendirme Protokolü, Performans Değerlendirme, OPNET simülatörü, AODV Yönlerdirme Protokolü

I dedicate this thesis to my father, mother, two brothers, sister, uncles, aunties, cousins and to all my friends.

ACKNOWLEDGMENTS

In the name of greatest All mighty ALLAH who has always bless us with potential knowledge and success.

I would like to express my sincere gratitude to my supervisor Asst. Prof. Dr. Gurcu Oz for her continuous support of my master study, for her patience, motivation, guidance, and knowledge. Her guidance helped me in all the time of study and writing of this thesis. I could not have imagined having a better supervisor for my master study.

And especially thankful to my parents and brothers and sister, for their support, effort, pain, and patience and to whom I own the success of my life.

Special thanks go to my twin brother Humman. Also I thank my friends Mohammed Namik, Ahmed Mahmoud, Hossam Nofal, Saif Anwer, Ahmed Salah, Mustafa Ibrahim, Liwaa Hussein, Anas Qasim, Ahmed Hani, Mohammed AL_sayed and Sinan Hazem for their help and support.

Also, my dearest thanks to my friend Ghassan A. Qas Marrogy for his help in the OPNET simulation program.

And not to forget to thanks all the people who help me out in my thesis such as Tanya Serenli Barkinay and Akile Serinkanl which they always there for me and supported me morally and psychologically.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
AODV	Ad hoc on Demand Distance Vector Routing
BSS	Basic Station Set
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear To Send
DCF	Distributed Coordination Function
DES	Discrete Event Simulation
DIFS	Distributed Inter-Frame Spacing
DSSS	Direct Sequences Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
IEEE	Institute of Electrical and Electronics Engineers
LLACKs	Link-large Acknowledgments
MAC	Medium Access Control
MANETs	Mobile Ad hoc Networks
MSDU	Mac Service Data Unit
MID	Multiple Interface Declaration
MPR	Multipoint Relays
NAC	Network Allocation Vector
NS 2	Network Simulation 2
OLSR	Optimized Link State Routing
OPNET	Optimized Network Engineering Tool

PRP	Proactive Routing Protocol
RERR	Route Error Message
RREP	Route Replay Packet
RREQ	Route Request Packet
RRP	Reactive Routing Protocol
RTS	Request To Send
SIFS	Short Inter-Frame Spacing
TCP	Transmission Control Protocol
TTL	Time to Life
UDP	User Datagram Protocol
WIMAX	Worldwide Interoperability for Microwave Access
WLSN	Wireless Local Area Network

Chapter 1

INTRODUCTION

Wireless networks are one of the most wide spread computer networks which utilize radio frequency channels to communicate between the nodes in the network without using any wire.

MANET stands for Mobile Ad hoc Network. It is a solid infrastructureless wireless network. A MANET can be formed either by mobile nodes or by both fixed and mobile nodes. Nodes randomly associate with each other forming arbitrary topologies. They act as both routers and usual nodes.

Ad hoc wireless network is one of the wireless networks that enable the users of the network to immediately communicate with each other. This means that ad hoc wireless networks do not require any routers or access points to be utilized in the network. Since there is no wire and no fixed router in this kind of networks, it is not difficult to enable the mobility in the network because the nodes will arbitrarily arrange themselves with respect to the topology changes. The transmission area of each node is limited. Hence, in order to reach a node that is out of a node’s transmission area, another node should be used as an intermediate node in order to forward the needed information. Since there is no any router or access point, every node inside the network can work as a router and accomplish the duty of forwarding data. Thus, there will be a multi hop wireless link between the sender and receiver.

Easy and fast deployment of wireless ad hoc networks and the decreased dependence on infrastructure makes this type of networks preferable in some areas. Besides being used as cell phones and for gaming purposes, wireless ad hoc networks can also be used in disaster areas or in search and rescue emergency operations. In our daily life, using wireless ad hoc network in taxis, stadiums and aircrafts is also possible. As for military purposes, these networks can be deployed on battlefield areas because they are good at mobility, fast and easy to setup.

Researches in this area have continued with emphasis on prominent studies on Ad hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA) and Optimized Link State Routing (OLSR) [1].

Routing protocols are classified either as reactive or proactive. Ad hoc routing protocols that are a combination of both reactive and proactive characteristics are referred to as hybrid. In this study, we considered one routing protocol and it is reactive: AODV with FTP application. We modeled MANET scenarios with varying message size for fixed and mobile scenarios and evaluated the performance of AODV with respect to real life experiments that we have compare the result of the simulations study and the real life experiments. The contribution of this study is taking a new perspective of using the AODV routing protocol with FTP application and various message size. Since most of the studies focus on keeping the message size constant and changing the routing protocols, whereas, in our case we keep the routing protocol constant, by using AODV routing protocol and change the message size. Also, in our mobility scenario, we consider the mobility speed of the nodes closed to human speed. However, the majority of the studies we make the mobility speed fast or very fast.

The organization of the thesis is as follows: Chapter 1 provides a general introduction. Chapter 2 introduces the routing protocol AODV in MANETs which is selected for investigation. Chapter 3 describes the simulation program OPNET. Chapter 4 presents the modeling of MANETs in OPNET which are presented earlier; simulation setup for different scenarios and the results of simulation are also discussed. Chapter 5 contains the conclusion and future work.

Chapter 2

OVERVIEW OF ROUTING PROTOCOLS

Wireless ad hoc network is a set of nodes that communicate with each other without the need to use router for centralized control. Source node can be any node in a wireless network, an intermediate node which work as a router, and a destination node. The main characteristics of a wireless ad hoc network can be changed with respect to the chosen routing protocol.

Wireless network has many routing protocols and we can mainly classify them as unicast, multicast and anycast. Unicast is a single connection between the client and the server. Unicast uses IP delivery methods like Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) which means that the packets will be send to a specific single destination host from a source node in a particular network. There is a one to one relationship between the source and the destination node. The second type is multicast which means sending a packet from a source node to a group of nodes in that network using networking technique of delivering the same packet simultaneously to a group of nodes. In this type of network, multicast address will be given to each node and in the same network more than one node can have the same multicast address. Consequently, when sending a packet to a unique multicast address, a group of nodes will receive this packet if they belong to that address. Multicast has a one-to-many association between network addresses and network end points. Finally, anycast means that a packet will be sent from a source node to the closest server or to the best localized server in the network. In anycast mechanism, in the network there is one or more server(s) and the goal is that the packet will be sent to the best server among all the other servers. And the word “best” can vary with respect to the anycast protocol that is selected. It may be the closest node, the least traffic involving server or any other things relying on the system used.

Wireless ad hoc networks have many routing protocols that are made for it. These protocols can be classified either as reactive or proactive [2]. Hybrid protocols are ad hoc routing protocols with a combination of both reactive and proactive properties.

2.1 Proactive Routing Protocols

Proactive routing protocols build and maintain routing data to the entire nodes. This is regardless whether the route is needed or not [3]. Every node saves the routing data for all the nodes in the network and that depends on the protocol used. The number of tables needed for maintaining the routing data can change. The most important thing is the tables are periodically updated. This is because the information transmission from a source node to any destination node continues even when the data flow does not exist. This kind of routing protocols has its advantages and disadvantages. One of its main advantages is that nodes can easily receive routing data and it’s easy to build a session. The disadvantages are: keeping excessive information in the nodes for route maintenance and when there is a failure in a specific link it is slow to restructure. OLSR (Optimized Link State Routing) [4] [5] is an example of a proactive routing protocol. It has working mechanisms that are unicast and proactive i.e. exchanging topology information will be done in frequent manner with other nodes in the network [6]. This protocol is the improvement of the conventional link state protocol developed for mobile ad hoc network and is also applied in WiMAX Mesh. Minimizing the size of the control packets, which are the OLSR accountabilities by reducing the desired control packets transmission number. OLSR primary goal is to regulate the control traffic overhead in the network with the Multipoint Relays (MPRs) assistance [7]. The main idea behind the OLSR protocol is the MPR concept. It is essentially a node's one-hop adjacent in the network.

2.2 Reactive Routing Protocols

Reactive protocols are known also as on demand protocols which have the capability of finding a route from a source node to a destination node(s) when a source node wants to send a packet. Generally, this means that before sending the original data to the destination, a route discovery mechanism is activated to discover the route that is going to be used for the information to be sent on it. Furthermore, reactive protocols have two types of protocols. The first one of reactive protocols works by combining the whole route address with the original data after discovering the best route and sending the entire packet. The intermediate nodes don’t have to care about to which node they have to send the packet since that information is provided within the packet inside the data that our goal is to send from the source node. The other kind works by putting a routing table within each intermediate node. Every time an intermediate node receives a packet, the current node will take the decision where to send the packet by searching at the table inside it. The distinction of this mechanism comes from setting the next hop address in the packet rather than setting the whole route information. The advantage of this algorithm is that it gives us lower route overhead and the disadvantage of this algorithm is its high latency when discovering routes. Another disadvantage is that the potential of the network will be blocked up when flooding is immoderate [8]. Reactive routing protocols have many types of routing protocols such as AODV, TORA, LAR and DSR. In our study, we used AODV routing protocol.

2.3 Ad hoc On-demand Distance Vector Routing (AODV)

Ad hoc On-demand Distance Vector Routing (AODV) [9] [10] is a novel algorithm for the ad hoc operation networks. Every mobile node works as a special router and routes are gained as required i.e. on-demand with little or no dependence on periodic announcements. For a dynamic self-starting network, AODV algorithm is very suitable as desired by users wanting to make use of the ad hoc networks. AODV presents loop free routes even when reforming broken links. Since the protocol does not need global periodic routing announcements, the request on the overall bandwidth available to the mobile nodes is considerably less than in those protocols that do require such announcements.

A pure on-demand route acquisition system can be called to the AODV, these nodes do not rely on active paths or keep any routing information neither contribute in any periodic routing table exchanges. In addition, a node doesn’t have to detect and keep a route to another node until it wants to communicate. To keep the most new routing information between nodes, the destination sequence numbering method will be applied.

Every ad hoc node keeps a monotonically rising sequence number counter which is applied to replace stale cached routes.

2.3.1 AODV Basic Operations

This section clarifies every operation required in an AODV [9] network to create, delete and maintain routes.

2.3.1.1 Path Discovery

The operation of Path Discovery is started whenever a source node wants to contact with another node that has no routing information in its table. Each node keeps two isolated counters: sequence number of a node and a broadcast identification. The source node begins path discovery by broadcasting a route request (RREQ) packet to its neighbors.

The RREQ consists of the following fields:

- Source address
- Source sequence number
- Broadcast ID
- Destination address
- Destination sequence number
- Hop count

What uniquely identifies a RREQ is both source address and broadcast ID. Broadcast ID is increased whenever the source released a new RREQ. Every neighbor either returns a route reply (RREP) back to the source to satisfy the RREQ or re-broadcasts the RREQ to its own neighbors after raising the count number of hops. Notice that a node can receive the same route broadcast packet from different neighbors which means that the node will have multiple duplicate of the packet. The intermediate node may drop the received RREQ and will not rebroadcast it, if it already has received a RREQ with the same broadcast ID and source address.

2.3.1.2 Reverse Path Setup

RREQ contains two sequence numbers included in it: the sequence number of the source and the last sequence number of the destination known to the source. The sequence number of the source is utilized to preserve the updated information considering the reverse route to the source and the sequence number of the destination specifies how updating a route must be to the target so the source can accept it.

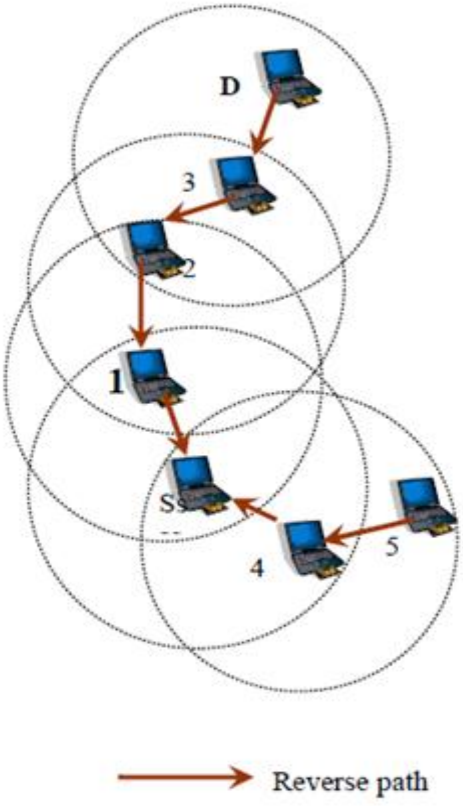


Figure 2.1: Reverse Path Settings [11]

Figure 2.1 shows that when the source node S decides that it wants a route to the target node D and there is no root available, instantly node S begins broadcasting RREQ (Route Request) message to its neighboring nodes in research of route to the target. The nodes 1 and 4 are start as neighbors to the node S that receives the RREQ message. And nodes 1 and 4 make a reverse link to the source from which they received RREQ. Since the nodes 1 and 4 are not aware of the link to the node D, they rebroadcast this RREQ to their neighboring nodes 2 and 5. As the RREQ travels from a source to different targets, it automatically adjusts the reverse path from every node back to the source as shown in Figure 2.1 above. If the node receives a RREP back to the node that created the RREQ, the reverse route is needed. Before broadcasting the RREQ, the making node buffers the RREQ ID and the originator IP address. In this way, when the node receives a packet again from its neighbors, it will not reprocess and re-forward the packet.

2.3.1.3 Forward Path Setup

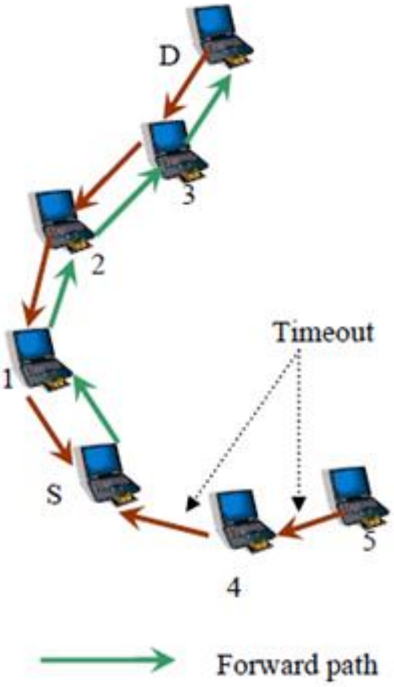


Figure 2.2: Forward Path Settings [11]

As shown in Figure 2.2 on forward path. A node will receive a RREQ that has a current route to the target or the target itself. The first check of the node will do when it receives the RREQ if it was received through a bi-directional link. If an intermediate node has a route entry for the desired destination, it decides whether the route is current by matching the sequence number of the destination in its own route entry to the sequence number of the destination in the RREQ. If the RREQ's destination sequence number is larger than that the one recorded by the intermediate node, the intermediate node should not utilize its recorded route to replay to the RREQ. Instead, the intermediate node re-broadcasts the RREQ. The response of the intermediate node can only happen when it has a route with a sequence number that is larger or equal to that in the RREQ. If it has a current route to the destination and if the RREQ has not been processed before, then the node unicasts a route reply packet (RREP) back to its neighbor from which it received the RREQ.

A RREP has the following information:

- Source address
- Destination address
- Destination sequence number
- Hop count
- Lifetime

By the time a broadcast packet reaches at a node that can provide a route to the destination, a reverse path has been created to the RREQ source. As the RREP goes back to the source, every node along the path establishes a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination, and saves the latest sequence number of the destination for the requested destination. Figure 2.2 shows the forward path setup as the RREP travels across the nodes 3, 2, 1 from the target D to the source node S. The RREP decides that nodes 4 and 5 are not in the path, and will timeout after active route timeout and will delete the reverse pointers from these nodes.

A node that receives a RREP will pass through the first RREP for a given source node heading to that source. If further RREPs are received, it refreshes its routing information and passes the RREP through only if the RREP has either a larger sequence number of the destination than the old RREP, or equal sequence number of destination with a smaller hop count. Now the source node S can start data transmission once it receives the first RREP, and can later refresh its routing information if it detects a better route.

2.3.2 Route Table Management

Route request expiration timer is a timer coupled with reverse path routing entries. The reason why the timer is used is to delete the reverse path routing entries from the nodes that do not rely on the path from the source to the destination. The expiration time rely on the ad-hoc network size. Another essential parameter that associates with routing entries is the route caching timeout or the time after which the route is considered invalid.

In every routing table entry, the active neighbors' address from which packets for the given destination are received is also preserved. If it establishes or relays one packet at least for that destination inside the most recent active timeout period then a neighbor is said to be active for that destination, this information is kept so that every active source nodes can be informed when a break happens in the link along the path to the destination. A route entry is called active, if any active neighbors are using it. And what keeps a route table entry for every destination of interest is the mobile node.

Every route table entry has the following information:

- Destination
- Next Hop
- Number of hops
- Sequence number for the destination
- Active neighbors for this route
- Expiration time for the route table entry

Every time a route entry is used to transfer data from a source to a destination, the entry timeout is reset to the current time plus active route timeout. If a fresh route is given to a mobile node, the mobile node compares the sequence number of the destination of the fresh route to the sequence number of the destination for the current route. The route with the largest sequence number is selected. If the sequence numbers are equal, then the fresh route is chosen only if it has a smaller metric to the destination.

2.3.3 Link Breakage

When a break happens in the link, the node must nullify the existing route in the routing table entry. The node must make a list of the influenced destinations and decides which neighbors can be influenced with this breakage. At last the node should submit the route error (RERR) message to the corresponding neighbors. The RERR message can be unicasted if one neighbor is only there or broadcasted if there are a lot of neighbors in need for that information. If broadcast is impossible, the hosts as well can recursively unicast the message to the neighbors that need the information.

2.3.4 Path Maintenance

Nodes movements that are not in an active path don’t influence the routing to that path’s destination. During an active session, if the source node moves, it can re-initiate the route discovery procedure to make a fresh route to the destination but when the destination or an intermediate node moves, a unique RREP is sent to the influenced source nodes. To guarantee symmetric links, a periodic hello messages can be applied, and also to discover any failures in the link. Instead, and with latency that is far less, applying link-layer acknowledgments (LLACKS) could discover these failures. A failure in the link is detected also if attempts to forward a packet to the next hop fail.

The moment that next hop becomes out of reach, the node upstream of the break spreads an undesirable RREP with a new sequence number and hops count of infinity to every active upstream neighbors. Those nodes then rebroadcast that message to their active neighbors and so on. This operation lasts until every active source nodes are informed.

When a broken link notification is received, source nodes can repeat the discovery operation if they still need destination route. To decide if a route is still required, a node may inspect if the route was used lately, and also check upper level protocol control blocks to see if connections stay open, applying the indicated destination whether the source node (or any other node along the old route) determines that it likes to reconstruct the route to the destination, it sends an RREQ with a sequence number of the destination of one larger than the old known sequence number, to guarantee that it constructs a fresh route and that no nodes response if they still consider the old route as valid.

2.3.5 Local Connectivity Management

Despite the fact that AODV is a reactive protocol it applies the Hello messages regularly to notify its neighbors that the route to the host is alive. The Hello messages are broadcasted with TTL and the value of it equals to 1, to ensure that the message won’t be forwarded further. It will update the host lifetime information in the routing table when the Hello message is sent and received by the host. If the host doesn’t obtain information from the host’s neighbor for letting hallo loss hello interval amount of time, after that the routing information inside the routing table is signed as lost information. This action creates the required RRER message to notify other hosts of the breakage in the link.

The local connectivity management with hello messages can be used to guarantee that only nodes with bidirectional connectivity are assumed to be neighbors. For this purpose, every hello sent by a node lists the nodes from which it has heard. Every node is examined to ensure that it uses routes only to neighbors that have heard the node’s hello message. To keep local bandwidth, such checking should be done only if explicitly configured into the nodes.

2.3.6 Local repair

When the breakage happened in the link occurs and if the destination from the host to breakage is less than or equals to a particulate amount of hops, the host can locally try to repair the link. To repair the link, the host increases the sequence number of the destination and broadcasts the RREQ message to the host. The TTL for the IP header should be calculated, so that local repair operation wouldn’t distribute over the network.

The host stands in for the RREP messages to its RREQ message for particulate amount of time. If the RREP message is not received, then it shifts the status of the routing table for the entry to invalid. If a host gets the RREP message, consequently the hop count metric is compared. If the hop metric from the message is larger than the old one then the RERR with the N field install is broadcasted. The N field in the RERR message shows that the host has locally reformed the link and the entry in the table shouldn't be canceled. The received RREP message is considered as original RREP message.

2.4 Properties of IEEE 802.11 Standards (WLAN)

802.11 refer to a family of specifications developed by the IEEE (Institute of Electrical and Electronics Engineers) for wireless LAN (WLAN) technology. 802.11 specify an over the air interface between a wireless client and a base station or between two wireless clients. The IEEE accepted the specification in 1997. There are different specifications of the 802.11 family such as 802.11a, 802.11b, 802.11e and 802.11g. In our study, we chose 802.11b for implementing. 802.11 applies to wireless LANs and provides 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS). The 802.11b applies to wireless LANS and provides 11 Mbps transmission (with a fallback to 5.5, 2 and 1-Mbps) in the 2.4 GHz band. The 802.11b uses a direct sequence spread spectrum for transmission data.

The system that has removable computers which can communicate to each other by radio can be expressed as a wireless LAN. These wireless LANs are of more several characteristics than traditional LANs and request specific MAC sub layer protocols.

Using traditional LAN protocol for WLAN is not that perfect because of the interference at the receiver, not at the sender. Traditional LAN protocol is utilized in WLAN and results in two types of problems called hidden station problem and exposed station problem [12].

2.4.1 Hidden Station Problem

Let's suppose that there are four stations and they are A, B, C and D as shown in Figure 2.3. The station A and B are within each other radio range and they can likely interfere with each other but station C can only interfere with stations B and D, but not station A.

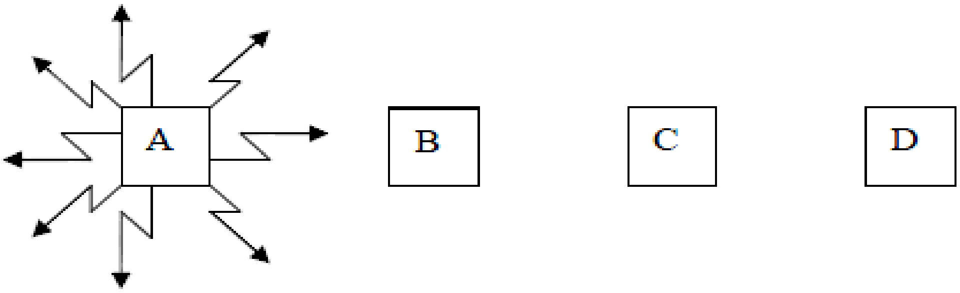


Figure 2.3: A Wireless LAN with Station A is Transmitting (Hidden Station Problem)

Now if the A station wants to transmit to station B as shown in the figure above. Station C feels the medium, but it will not hear station A because it is far from the range of station C, therefore; station C makes an incorrect decision that can transmit. If in this situation, station C begins transmitting, it will interfere at station B, which eliminates the framework of station A. This simple hidden station problem occurs when the station cannot discover a possible rival for the medium because the rival is far away.

2.4.2 Exposed Station Problem

First let’s suppose that station B is transmitting to station A as shown in the Figure 2.4. If station C feels the medium, station C will hear a continuous transmission and makes an incorrect decision not to send to the station D, in reality, transmission like that can be a reason for poor reception only to the area between station B and C, where neither of the intentional receivers exists. This state is called as the exposed station problem.

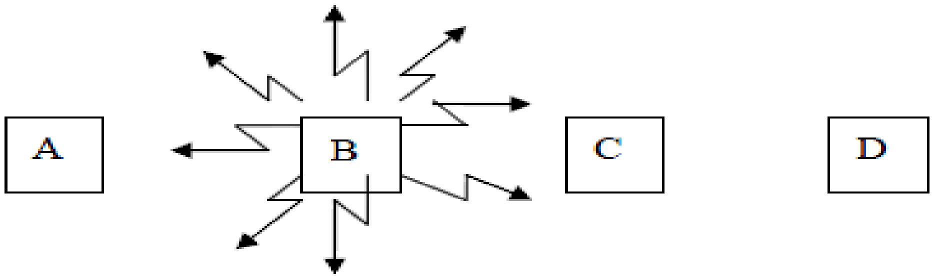


Figure 2.4: Wireless LAN with Station B is Transmitting (Exposed Station Problem)

2.4.3 Basic Access Method: CSMA/CA

The access mechanism that is utilized in WLAN is CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance) simultaneously with the Distributed Coordination Function (DCF). DCF utilized RTS (Request To Send) with CTS (Clear To Send) indicates to minimize the potential of collisions that happen because of the hidden station problem [12]. To avoid the exposed station problem, a station has to wait a randomly back off time between two sequential novel packet transmissions time. Next paragraphs will show how the mechanism works.

In wireless network, if the station decides to transmit information, it implements a carrier sense of the medium first of all. If the medium is idle, it waits for a period of time called DIFS (Distributed Inter-Frame Spacing), which means that the station has to wait a period of time before sensing the medium again. If in this satiation, the medium are idle again after this DIFS period, the package will be transmitted directly. In the receiver side, it also waits for a period of time called SIFS (Short Inter Frame Spacing). After the SIFS, the Acknowledgement ACK responses back to the transmission station. For all other stations that want to transmit during this time, the medium will be accessible; they will have to delay access [13].

On the other hand, if the transmitter which will send the data feels that the medium is occupied, it should wait a DIFS and then enters into the contention stage. In the contention stage, every station has to produce a random back off period and looks forward the time stage. After each elapsed period which is calculated in a slot, the back off amount is counted down, but however the medium becomes idle for transmission. If the medium becomes idle for sending the data next to the back off time stage, the station should wait for a DIFS as shown above on the top and sends the data if the medium is kept idle.

Virtual Carrier Sense: The station prepared for transmitting a packet will initially transmit a RTS (Request To Send) which is a small monitoring packet, this RTS contains the information of the server, destination and the period of the next bargain and at the destination it ought to reply with CTS (Clear To Send) which is a reply monitoring packet if the medium is idle, which contains the similar period information. The whole other stations which receive any of the RTS or the CTS will put their Virtual Carrier Sense signal named NAV (Network Allocation Vector) for the specific period. NAV is a utilized reference on how an extended medium be reserved. Figure 2.5 below shows all the operations that are mentioned above.

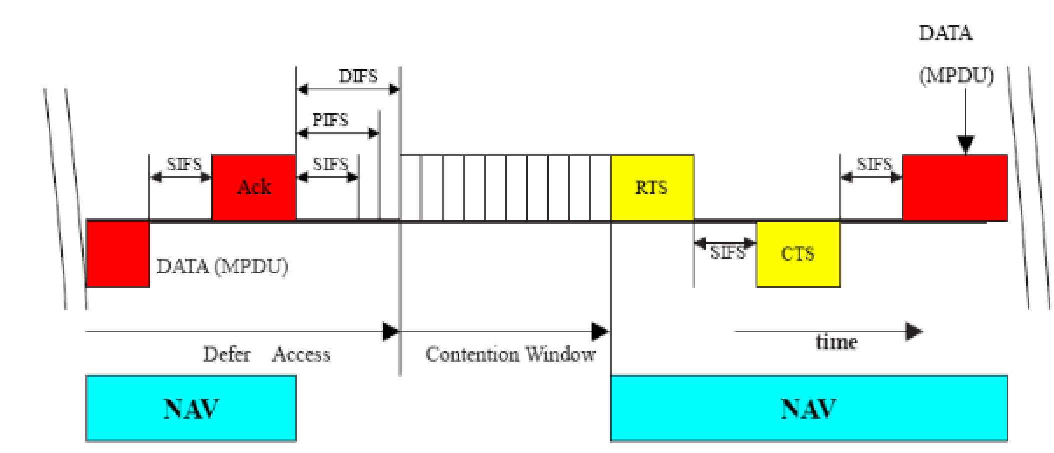


Figure 2.5: WLAN CSMA/CA Medium Access Scheme [14]

2.5 Main Directions to Investigate Wireless Ad Hoc Networks

Wireless ad hoc networks are investigated under two main directions. The first direction is related to the deployment of such networks in real-world outdoor environments, namely experimental studies. It requires large resources, long time and many participants to perform such studies. The lack of repeatability is another drawback of the experimental studies, that is, the behavior of the investigated network cannot be evaluated many times with exactly the same environmental effects. However, this approach provides very valuable information about actual characteristics of wireless ad hoc networks. And since there is no potential to have inaccurate or wrong assumptions about external influences, more realistic output data is obtained as well. A survey of existing real-world ad hoc test beds is given in [15].

The second direction is related to simulation modeling. Simulation modeling is an attempt to form a simplified abstraction of a real system at a digital computing environment, so that the system can be studied to investigate its behavior, under various conditions, and gain insights on how the system operates [16]. In order to clarify and solidify the definition of simulation modeling, descriptions of a real system, model of a system and discrete-event system simulation are provided in the following text.

A real system can be defined as, a set of elements that interact with each other to perform some common task. Considering dynamic systems, continuous and discrete systems are the main categories. In continuous systems, state variables, variables that are chosen to describe the behavior of the system, are continuously changing in time and taking continuous values. However, in discrete systems, state variables are changing at discrete moments of time and only take discrete values [16].

A simplified abstraction of a system that is detailed enough to allow the derivation of desired performance measures with sufficient accuracy is called a model of the system. Models can be static/dynamic, deterministic/stochastic and continuous/discrete.

Simulation modeling has other great advantages, such as: requirement for much fewer resources (participants, time and equipments) as compared to the experimental studies, manageability of time (time can be easily compressed/expanded), Competence to investigate behavior and characteristics of ad hoc networks with arbitrary large number of mobile nodes and any desired combination of parameters, being the only type of investigation possible, for most complex, real-world systems with stochastic elements, since they cannot be accurately described by a mathematical model for analytical evaluation, usability to answer “what if” questions, ability to test different modes of operation outside the real system, without disturbing ongoing operations, in the analysis of an existing system, and ability to check design variants before implementation in the design of new systems [17], [18].

Beside these advantages, simulation modeling also has its problems and challenges. Model building requires special training and experience, in order to be able to develop accurate models. Where in our model, choosing sufficiently realistic mobility model for node movement, specifying a general and feasible scheme of inter-node communication and determining reasonable performance metrics that reflect important characteristics of the network, as well as, how to derive these metrics from the raw simulation data, are some of the complexities that I face. After modeling and run of simulation, the obtained results can be difficult to interpret, as the output results are random. Accordingly, it is hard to determine if an observation is a result of randomness or system interrelationship [19].

2.6 Literature Review of the Existing Work

In [20], the authors examined the performance of AODV, DSDV and OLSR in different simulation parameters using NS2 simulator. The used performance metrics are packet delivery ratio, average delay and throughput. The simulation results show that using AODV protocol the packet delivery ratio has high value in smaller packet size. By increasing the packet size (256, 512, 1024, 2048 and 4096 bytes), the packet delivery ratio is decreased.

In [21] a performance comparison of protocols AODV, DSR and DSDV on CBR traffic using NS2 simulator was preformed. In this study, the performance metrics used are packet delivery ratio, average end to end delay and normalized routing load. The results obtained from the simulation study show that with AODV routing protocol the packet delivery ratio start with the high value and getting decreed by the increasing of sending rate.

In [22], they addressed the on-demand routing protocols by focusing on DSR protocol and AODV routing protocol. Ftp traffic are used with OPNET simulator to establish the simulation models of DSR and AODV routing protocols. The performance metric used in this study are routing discovery time, network delay, average number of hops and network throughput.

In [23], the authors evaluated the performance of AODV and OLSR in various performance metrics such packet delivery ratio, routing overhead and end to end delay.

NS2 simulator is used with CBR traffic. By increasing the data traffic rate, the packet delivery ratio of AODV routing protocol begin to decrease.

Table 1 summarizes a few works that have been done recently using OPNET simulation. Detailed simulation and parameters could be observed from this table. In some of the simulation, results have been drown with respect to time but here in this thesis the different number of nodes [9, 30, 60 and 90] and different file size [500, 1000, 1500, 2000, 2500, 3000, 4000 and 6000] is shown in the results. Which we have several simulation scenarios by OPNET simulator for fix and mobile nodes, where changing the number of nodes and the message size and fix the speed of nodes for the mobility scenarios similar to walking speed and then did the real life work to test the result of the simulation and compare them.

Chapter 3

TOOLS FOR MODELLING AND SIMULATION OF WIRELESS AD HOC NETWORKS

There are a lot of simulation packages available, but NS2 (Network Simulator 2) [24] and OPNET (Optimized Network Engineering Tool) [25] [26] are the most famous tools utilized in the wireless ad hoc networks for modeling and simulation. NS2 is a discrete event network simulator that is mostly applied for multicast routing protocols ad hoc networks. User offers the topology of the network out of the simulation interface. After that the program, using particular parameters, simulates the offered topology. Support for famous network protocols is counted as one of the main benefits of NS2. But the simulator has incomplete wireless MAC/PHY layer definition that modeling of obstacles is unaccomplished.

In this study, OPNET was chosen as the simulation program. Since the measurability and the efficiency of the simulation for this program is one of the best because of its tremendous characteristic like comprehensive graphical user interface and animation, as well as it has several of protocol and vender devices model with massive flexibility for checking and analysis. Moreover, it offers object oriented modeling and open source code model that offers easier understanding of the system.

In this chapter, we will demonstrate by details the OPNET simulator architecture in four sections; OPNET Architecture, MANET Model Architecture in OPNET, Configuring routing protocols in OPNET, and Taking results of Route.

3.1 OPNET Architecture

OPNET provides a comprehensive development environment for modeling and performance evaluation of communication networks and distributed systems. The package consists of a number of tools, each one focusing on particular aspects of the modeling task. These tools fall into three major categories that correspond to the three phases of modeling and simulation projects: Specification, Data Collection and Simulation and Analysis.

These phases are necessarily performed in sequence. They generally form a cycle, with a return to Specification following Analysis. Specification is actually divided into two parts: initial specification and re-specification, with only the latter belonging to the cycle, as illustrated in Figure 3.1

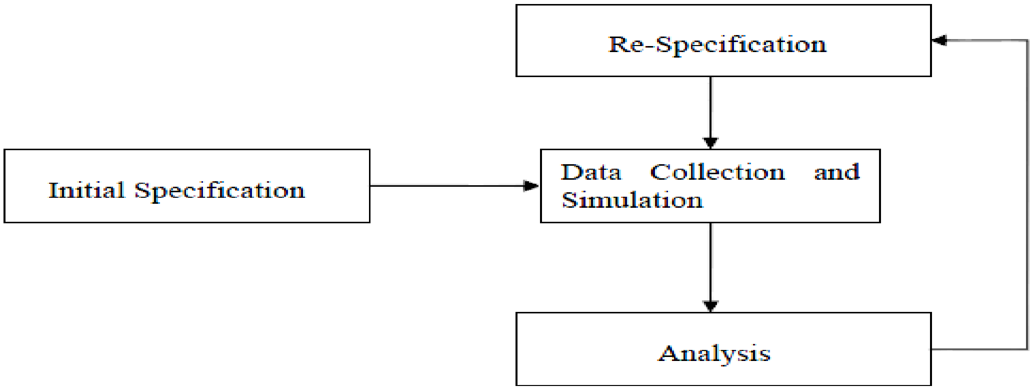


Figure 3.1: Simulation Project Cycle of OPNET

3.2 MANET Model Architecture in OPNET

The routing protocols such as AODV, DSR, TORA and OLSR are available to use in OPNET version17.1. ODPFv3 [27] for the MANET routing protocol are under development. This part explains model architecture, node models of MANET and all source, header and external files that are used by AODV process are shown in Figure 3.2.

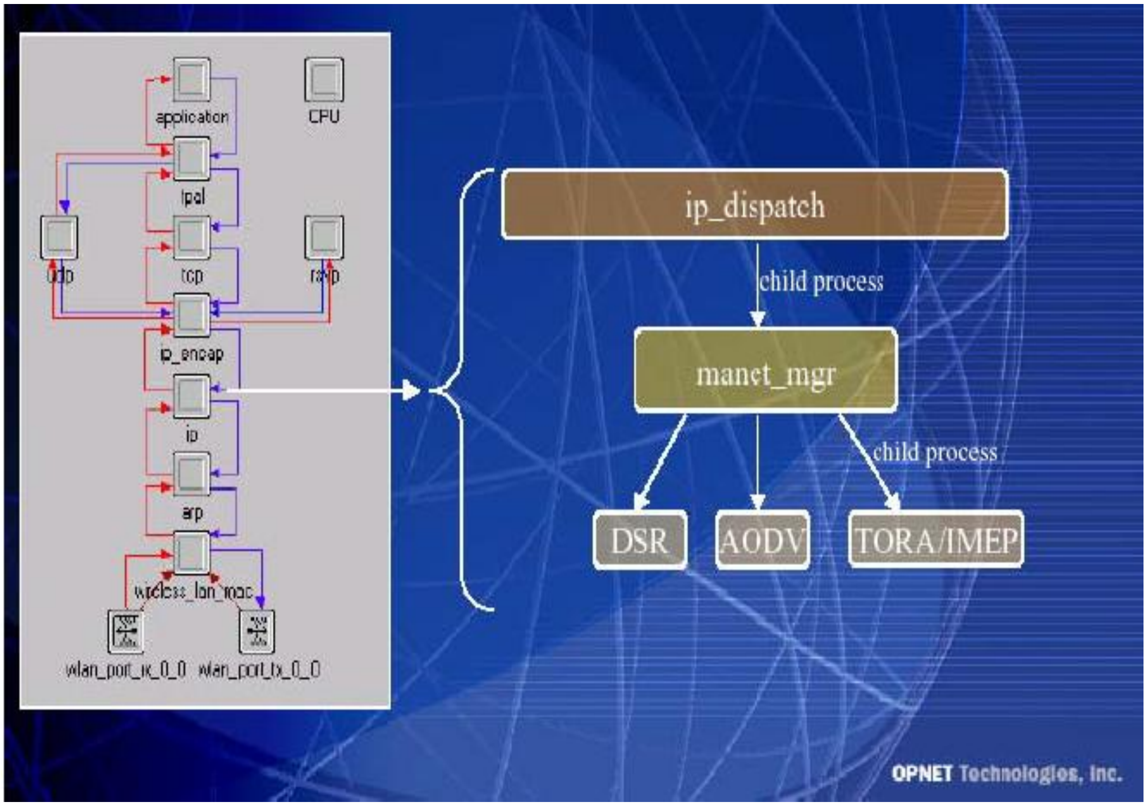
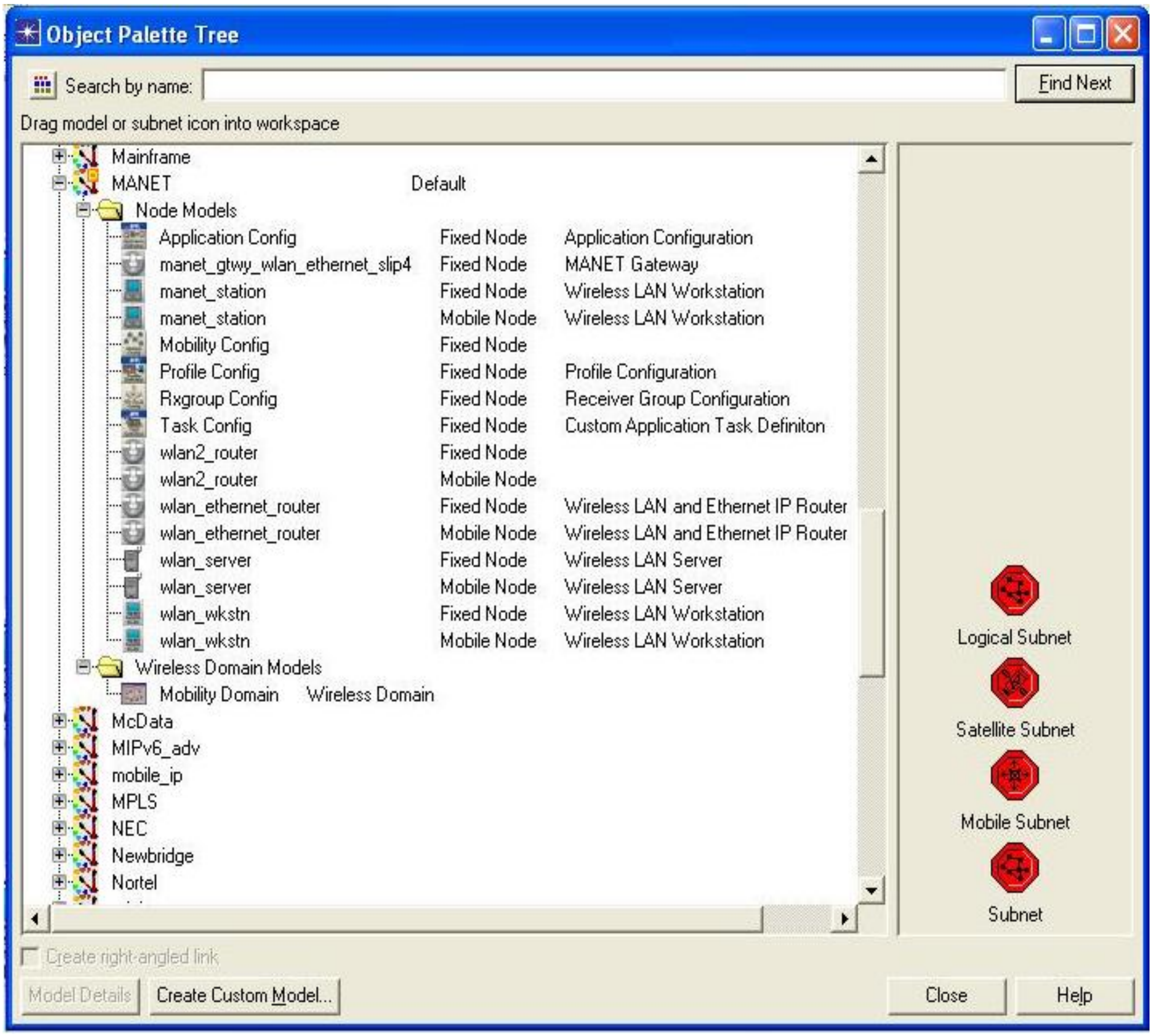


Figure 3.2: MANET Model Architecture [28]

The figure above illustrates the node model architecture of a MANET node. This MANET node could be a WLAN workstation operating in ad hoc mode. The position of ip_dispatch of the ip_encap process created a manet_mgr that manage whole ad hoc routing protocols in OPNET. The manet_mgr again creates a different specified process for the wanted ad hoc routing protocol as defined in the parameter.

3.2.1 Node Models in MANET

Whole nodes that used in MANET are included in the MANET object plate tree as shown in Figure 3.3. In our study, we used most of the nodes such as Application Config, Mobility Config, Profile Config, Rxgroup Config and Wlan_wkstn for both fix and mobility scenarios.



Mainframe		
MANET	Default	
Node Models		
Application Config	Fixed Node	Application Configuration
manet_gtwy_wlan_ethernet_slip4	Fixed Node	MANET Gateway
manet_station	Fixed Node	Wireless LAN Workstation
manet_station	Mobile Node	Wireless LAN Workstation
Mobility Config	Fixed Node	
Profile Config	Fixed Node	Profile Configuration
Rxgroup Config	Fixed Node	Receiver Group Configuration
Task Config	Fixed Node	Custom Application Task Definiton
wlan2_router	Fixed Node	
wlan2_router	Mobile Node	
wlan_ethernet_router	Fixed Node	Wireless LAN and Ethernet IP Router
wlan_ethernet_router	Mobile Node	Wireless LAN and Ethernet IP Router
wlan_server	Fixed Node	Wireless LAN Server
wlan_server	Mobile Node	Wireless LAN Server
wlan_wkstn	Fixed Node	Wireless LAN Workstation
wlan_wkstn	Mobile Node	Wireless LAN Workstation
Wireless Domain Models		
Mobility Domain	Wireless Domain	
McData		
MIPv6_adv		
mobile_ip		
MPLS		
NEC		
Newbridge		
Nortel		

Figure 3.3: MANET Object Palette Tree

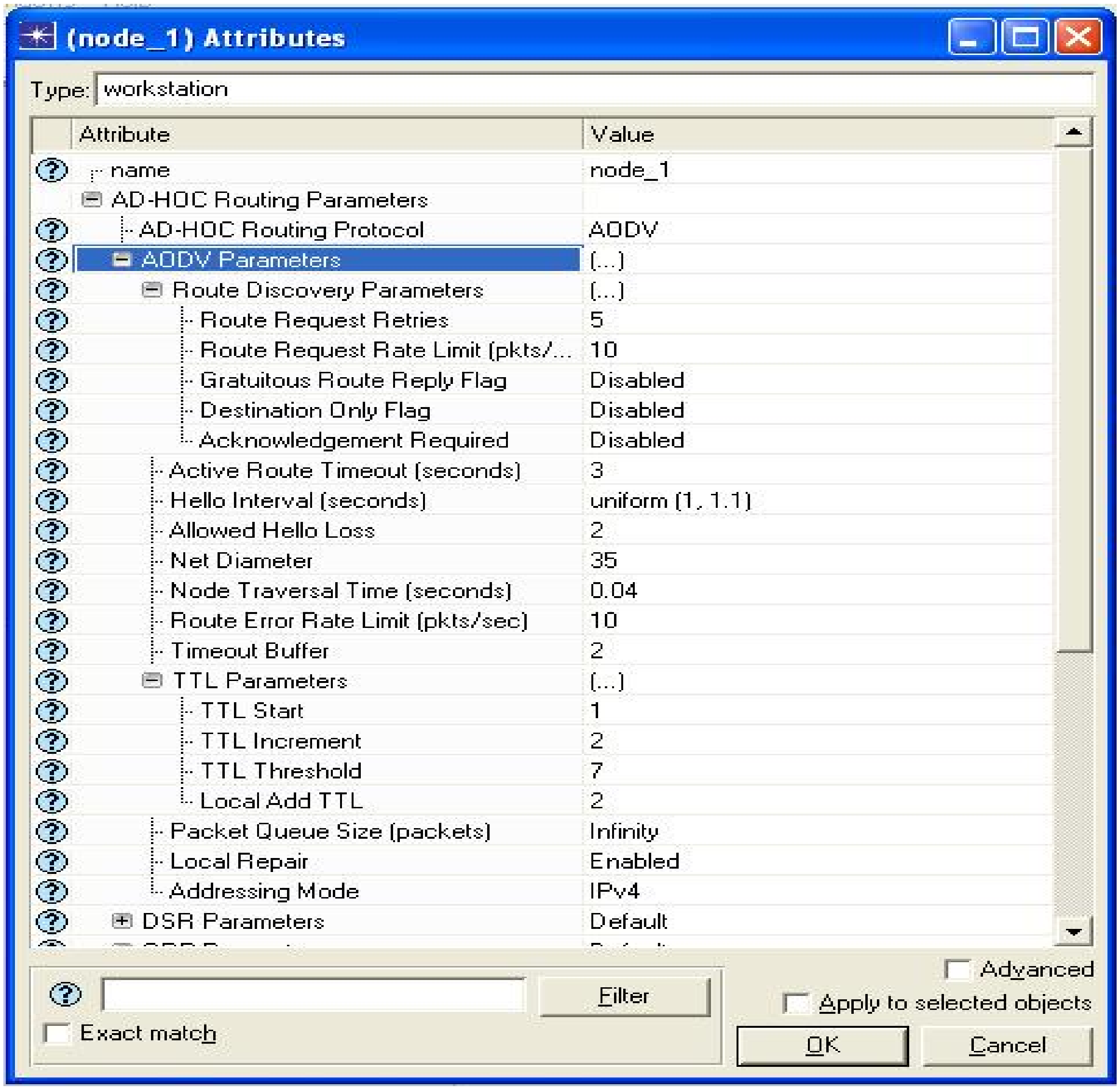
In MANET, wireless LAN workstations and servers nodes models could be utilized for generating application traffic like FTP, E-mail, and HTTP over TCP over IP over WLAN. For run the AODV, these nodes could be configured to run it as routing protocol. Where, the MANET stations models can be used for generating a raw of packets on IP in WLAN. They can be configured as a transit source or destination and also can be configured for running AODV as the routing protocol.

The application configuration was every profile and was built using various application definitions. For every application definition, we enable specifying the use of parameters like start time, duration and repeatability. We can have also two similar applications having different uses of the parameters; where we could use various names to correspond these as two distinguished application definitions. The mobility configuration to which these nodes can be applied to determine mobility profiles that individual nodes refer to model mobility. This node controls the nodes movement depending on the configured parameters.

Profiles configuration defines the activity patterns of a user or group of users in terms of the applications used over a period of time. You can have many various profiles running on a given LAN or workstation. These profiles can represent various user groups. For example, you can have an Engineering profile, a Sales profile and an administration profile to describe normal applications used for each employee group. The receiver group configuration node is applied to count the possible set of receivers with which a node can communicate. This utility node can extremely increase the speed of a simulation by eliminating receivers that do not match.

3.3 Configuring AODV in OPNET

When clicking the right click on any node put in the project modifier, a new window should appear to modify attribute values of different parameters. Figure 3.4 shows the configuration of AODV parameters.



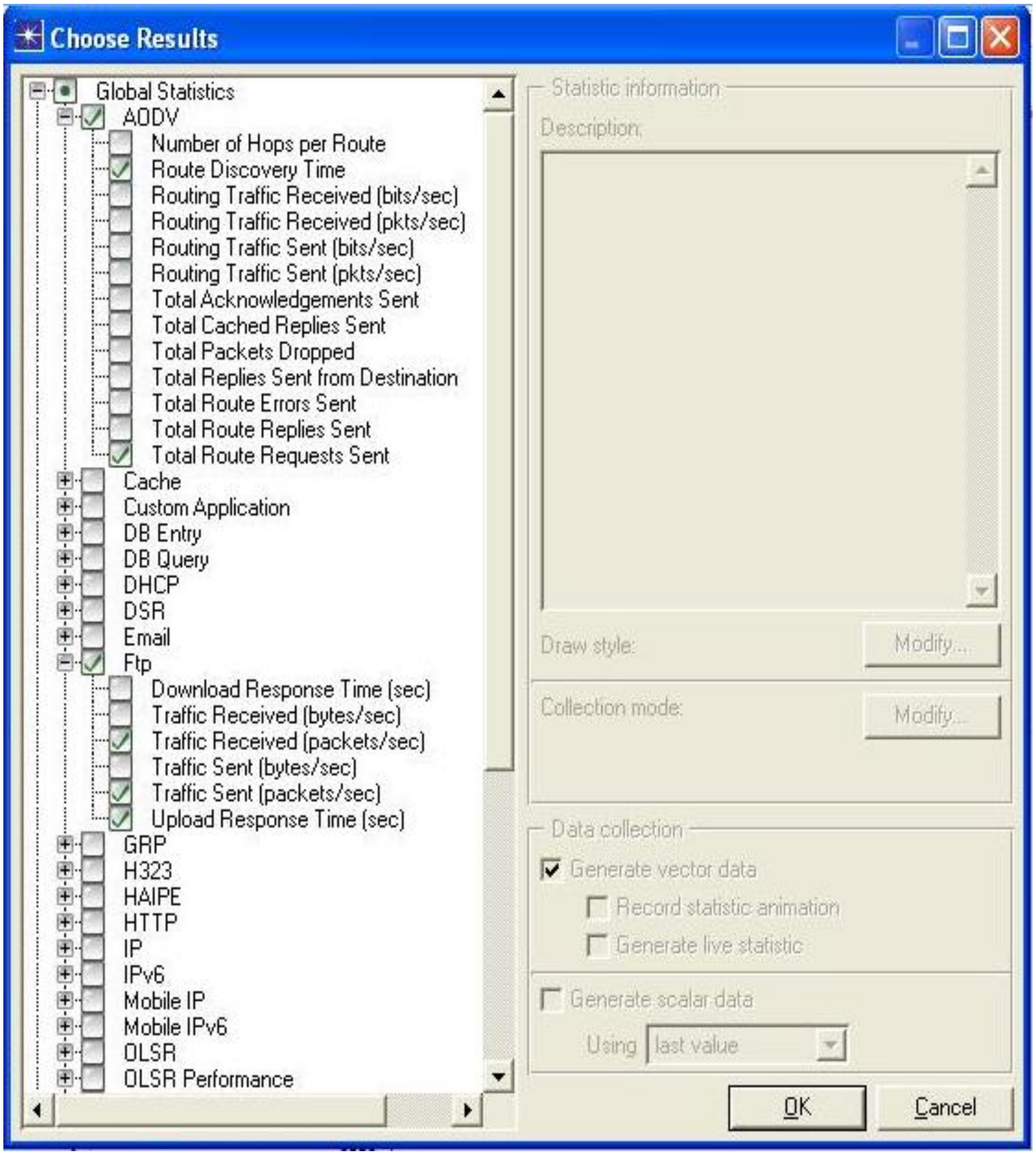
Type: workstation

Attribute	Value
name	node_1
⊟ AD-HOC Routing Parameters	
AD-HOC Routing Protocol	AODV
⊟ AODV Parameters	(...)
⊟ Route Discovery Parameters	(...)
Route Request Retries	5
Route Request Rate Limit (pkts/...	10
Gratuitous Route Reply Flag	Disabled
Destination Only Flag	Disabled
Acknowledgement Required	Disabled
Active Route Timeout (seconds)	3
Hello Interval (seconds)	uniform (1, 1.1)
Allowed Hello Loss	2
Net Diameter	35
Node Traversal Time (seconds)	0.04
Route Error Rate Limit (pkts/sec)	10
Timeout Buffer	2
⊟ TTL Parameters	(...)
TTL Start	1
TTL Increment	2
TTL Threshold	7
Local Add TTL	2
Packet Queue Size (packets)	Infinity
Local Repair	Enabled
Addressing Mode	IPv4
⊞ DSR Parameters	Default

Figure 3.4: Configuration of AODV in OPNET

3.4 Taking Results of AODV

By just right click on the project editor new editor will be opened to choose individual DES (Discrete Event Simulation) statistics, by going in that, we will have the choice to select various statistics that are going to be simulated. Figure 3.5 shows how to choose the statistics from OPNET project editor.



Choose Results

- ● Global Statistics
  - ☑ AODV
    - □ Number of Hops per Route
    - ☑ Route Discovery Time
    - □ Routing Traffic Received (bits/sec)
    - □ Routing Traffic Received (pkts/sec)
    - □ Routing Traffic Sent (bits/sec)
    - □ Routing Traffic Sent (pkts/sec)
    - □ Total Acknowledgements Sent
    - □ Total Cached Replies Sent
    - □ Total Packets Dropped
    - □ Total Replies Sent from Destination
    - □ Total Route Errors Sent
    - □ Total Route Replies Sent
    - ☑ Total Route Requests Sent
  - □ Cache
  - □ Custom Application
  - □ DB Entry
  - □ DB Query
  - □ DHCP
  - □ DSR
  - □ Email
  - ☑ Ftp
    - □ Download Response Time (sec)
    - □ Traffic Received (bytes/sec)
    - ☑ Traffic Received (packets/sec)
    - □ Traffic Sent (bytes/sec)
    - ☑ Traffic Sent (packets/sec)
    - ☑ Upload Response Time (sec)
  - □ GRP
  - □ H323
  - □ HAIPE
  - □ HTTP
  - □ IP
  - □ IPv6
  - □ Mobile IP
  - □ Mobile IPv6
  - □ OLSR
  - □ OLSR Performance

Statistic information

Description:

Draw style: Modify...

Collection mode: Modify...

Data collection

☑ Generate vector data

□ Record statistic animation

□ Generate live statistic

□ Generate scalar data

Using last value

OK Cancel

Figure 3.5: Choosing Statistics

Chapter 4

SIMULATION SETUP IN OPNET AND RESULTS

During this chapter, we will explain the simulation setup and modelling of network protocols with default parameters using a MANET model which is supported by OPNET 17.1. Moreover, the network scenarios are explained and a comparison of the results of the simulation is made.

4.1 Performance Metrics

In our simulation, route discovery time, total route request sent are used as performance metrics for the AODV routing protocol, upload response time and packet delivery ratio are used as performance metrics for the FTP application.

The route discovery time is the time to discover a route to a specific destination. It’s the time when a route request was out to discover a route to that destination until the time a route reply is received with a route to that destination. This statistic represents the time to discover a route to a specific destination by all nodes in the network.

Total route request sent is a statistic that represents the total number of route request packets sent by all nodes in the network during route discovery.

Packet delivery ratio metric consists of two parts, traffic received and traffic sent, which can be defined as follows: the traffic received is the average number of packets forwarded to all FTP applications by the transport layers in the network. The traffic sent is the average number of packets submitted to the transport layers by all FTP applications in the network. In our simulations there is one client and one server and other nodes are intermediates. The packet delivery ratio is found by the following equation:

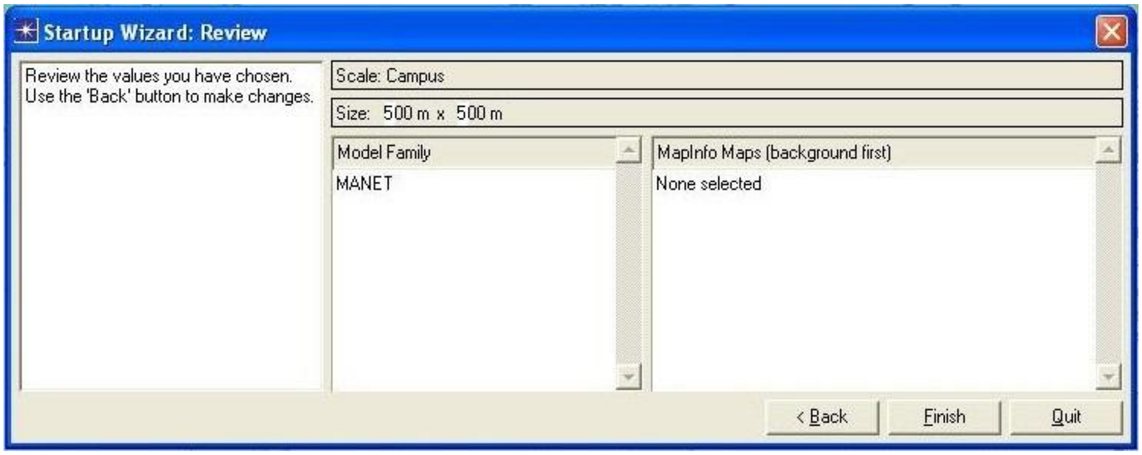
$$\text{Packet delivery ratio} = \frac{\text{Number of packet received by application layer}}{\text{Number of packet send from application layer}} \qquad (1)$$

Upload response time is the time to elapse between sending a file and receiving the response. The response time for responses sent from any server to an FTP application is included in this statistic.

4.2 Simulation Setup in OPNET

In the simulation program OPNET 17.1 which is used in this study, the program is supported the routing protocols such as AODV, DSR, OLSR, GRP and TORA. In our project, we choose the protocol AODV to apply it with the chancing the data size. All the nodes used in the simulation are set with IPv4 were auto configured. To complete the project, we have at least 60 sets of simulations details that should be designed and chosen carefully to make the project run. After the running of the simulation, and collecting statistical data, all of the scenarios are run for 300 seconds. In order to design a MANET with a routing protocol, someone should follow the following steps:

First of all, when we want to create our project we should name it. Then, the Scenario name must be a one that indicates what we work on; like in our case it’s named (9 fixable nodes in AODV). After naming, we create empty scenario and then select campus from the network Scale. In our project, we choose the area size to 500 by 500 square meters just like the area size that we work in real life. When we reach to model family list, we choose MANET by clicking on it. Figure 4.1 shows these steps.


Startup Wizard: Review

Review the values you have chosen.
Use the 'Back' button to make changes.

Scale: Campus

Size: 500 m x 500 m

Model Family
MANET

MapInfo Maps (background first)
None selected

Figure 4.1. Review of Startup Wizard

After we did these settings, we will get an empty screen in which we can put our nodes and complete the rest setting so we can run it at the end.

Now let’s start with node and how we made the setting:

4.3 Application Configuration

From the object palette, we can choose and integrate the application configuration on the campus network. After we open it, it will pop a window which includes a name if we want to name the application and a description table that specifies various parameters. A user profile is built using various application definitions. An application definition specifies an application with parameters. An application may have tasks and a task may have multiple phases. A phase can have many requests and responses. In our study, we have chosen the FTP application.

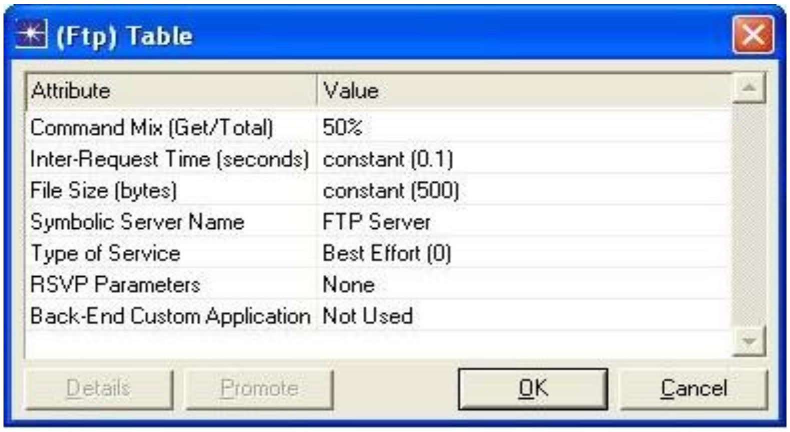
4.3.1 FTP Application

FTP is a file transfer protocol used to perform huge data transfer from server to user agents. Main objects of FTP include [29] file sharing promotion between computers, usage of remote systems through some applications; efficiently and reliably data transfers; they are designed specifically for application programs for utilization. The client always downloads one file per session in which the server may change for each session. In our work, we used FTP application with different file size, which is verging from 500 to 6000 bytes.

Also, when we double click on FTP it will show another window, from which we can control the same specific of, FTP such as Inter-request time and file size and others.

Inter-request time defines the amount of time between file transfers. The start time for a file transfer session is computed by adding the inter-request time to the time that the previous file transfer started. The unit used for measuring inter-request time is a second. In our simulation, the inter-request time has been set to 0.1 second.

File size defines the size of a file transfer, and the unit used for file size is a byte. In our simulation, file size is changed from 500, 1000, 1500, 2000, 2500, 3000, 4000 and 6000 bytes. This means that we need to run the same scenario eight times for each set of contract so we can collect all the data we need. The following Figure 4.2 shows these settings.



Attribute	Value
Command Mix (Get/Total)	50%
Inter-Request Time (seconds)	constant (0.1)
File Size (bytes)	constant (500)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

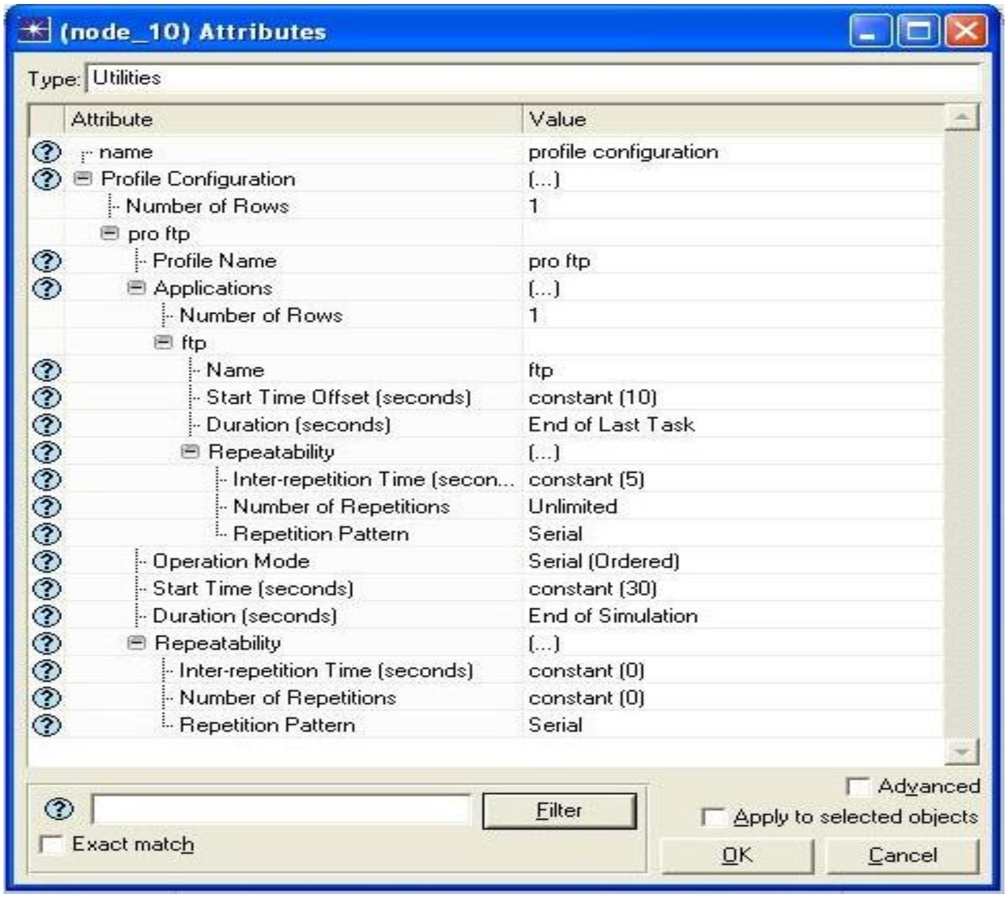
Figure 4.2: FTP Setting

4.3.2 Command Mix (Get/Total)

It is mean the percentage of file "get" commands to the total FTP commands. The remaining percent of the commands are FTP file "put" transactions.

4.3.3 Profile Configuration

From the object palette, we choose Profile configuration and drag it to the campus network. A profile configuration is a profile that describes user activity over a period of time. A profile consists of many different applications. For example, a "Human Resources" user profile may contain "Email", "Web" and "Database". We can specify various loading characteristics for the different applications. Each application is described in detail within the application configuration object. The profile created on this object is referenced by the individual workstation to generate traffic. The following Figure 4.3 shows the setting of profile configuration.



Type: Utilities

Attribute	Value
name	profile configuration
⊟ Profile Configuration	(...)
Number of Rows	1
⊟ pro ftp	
Profile Name	pro ftp
⊟ Applications	(...)
Number of Rows	1
⊟ ftp	
Name	ftp
Start Time Offset (seconds)	constant (10)
Duration (seconds)	End of Last Task
⊟ Repeatability	(...)
Inter-repetition Time (secon...	constant (5)
Number of Repetitions	Unlimited
Repetition Pattern	Serial
Operation Mode	Serial (Ordered)
Start Time (seconds)	constant (30)
Duration (seconds)	End of Simulation
⊟ Repeatability	(...)
Inter-repetition Time (seconds)	constant (0)
Number of Repetitions	constant (0)
Repetition Pattern	Serial

Figure 4.3: Profile Configuration Attribute

4.3.3.1 Start Time Offset

If the "Operation Mode" is set to "Simultaneous", this offset refers to the offset of the first instance of each application (defined in the profile), from the start of the profile.

If the "Operation Mode" is set to "Serial (Ordered)" or "Serial (Random)", this offset refers to the time from the start of the profile to the start of the first application. It also serves as the inter-application time between the ends of one application to the start of the next. If an application does not end (e.g., duration set to 'End of Profile), subsequent applications won't start. In our study, we set start time offset to constant (10). This means that application will start after (10) seconds and then will begin the profile at (30) seconds because we previously set it to (30) from start time as shown in Figure 4.4.

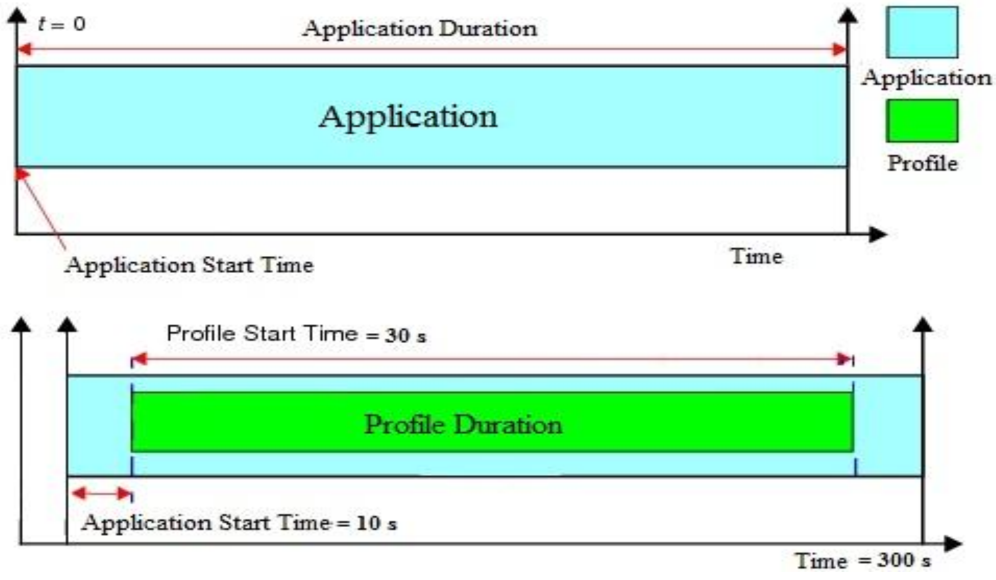


Figure 4.4: Application and Profile Starting

4.3.3.2 Duration

In the profile configuration, we have two types of duration. The first one is for the application, and it means the maximum amount of time allowed for an application session before it aborts. This is often used as a time out. When set to “End of Profile”, the application will end when the profile duration has expired. When set to “End of Last Task”, the application will end when the last task of the application has completed regardless of task completion times. In our thesis, we set to “End of Last Task”.

The second duration is for the profile, which means the maximum amount of time allowed for the profile before it ends. When set to “End of Simulation” the profile is allowed to continue indefinitely till the simulation ends. When set to “End of Last Application” the profile is allowed to continue till the last instance of an application running as part of this profile ends. If the application repeatability is unlimited, the profile will end when the simulation ends. Repeated profiles should not have their duration set to “End of Simulation”. In our study, we set it to “End of Simulation” as shown in Figure 4.4.

4.3.3.3 Inter-repetition Time

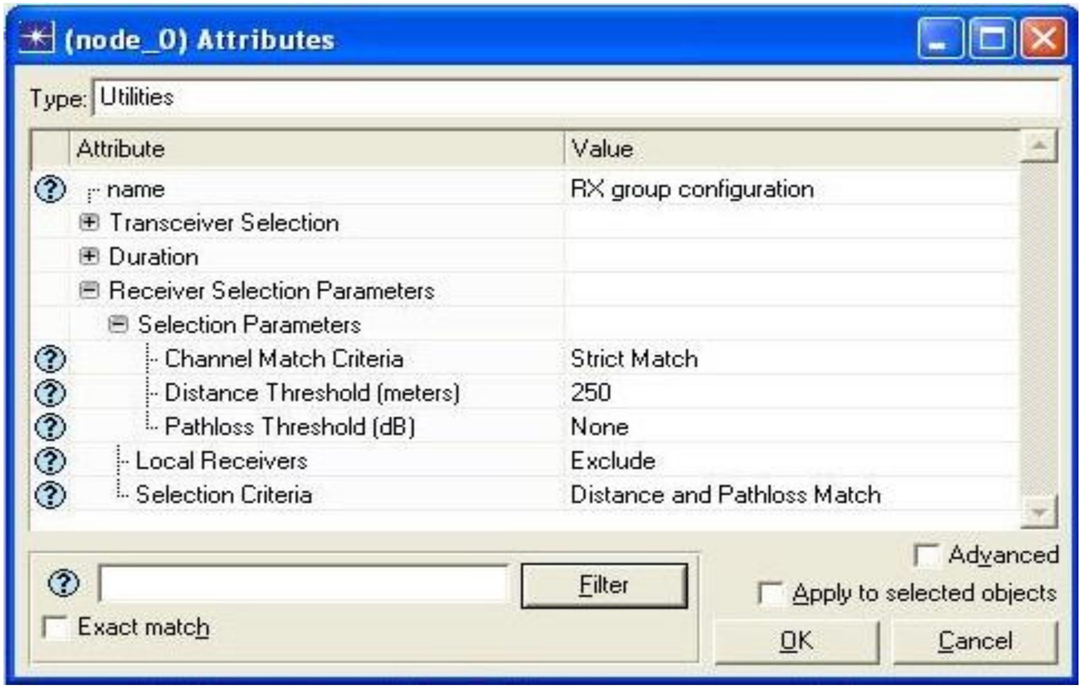
It is defined when the next session of the application will start depending on the Repetition pattern.

Serial - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session completed.

Concurrent - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session started. When set to concurrent, the mean outcome should not be zero. A mean of zero would cause sessions to be created at an infinite rate. In our study, we set it to constant (5) as shown in Figure 4.4.

4.3.4 RX group Configuration

From the object palette, we choose RX group configuration and drag it to the campus network. The main purpose of using RX group configuration is to control the distance of the transmission of each node in the network because we cannot control the transmission distance by the transmit power, since in our project, we already set the transmit power to the minimum value. In our study, we set the distance of the transmission for every node to 250 meter. The following Figure 4.5 shows this setup.



(node_0) Attributes

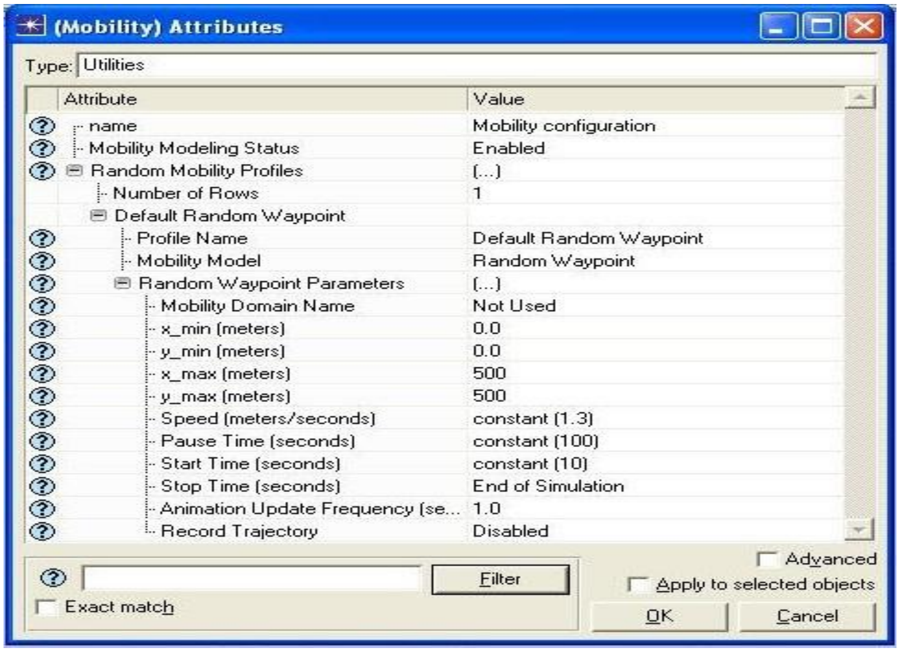
Type: Utilities

Attribute	Value
name	RX group configuration
⊞ Transceiver Selection	
⊞ Duration	
⊟ Receiver Selection Parameters	
⊟ Selection Parameters	
Channel Match Criteria	Strict Match
Distance Threshold (meters)	250
Pathloss Threshold (dB)	None
Local Receivers	Exclude
Selection Criteria	Distance and Pathloss Match

Figure 4.5: RX group Configuration Attribute

4.3.5 Mobility Configuration

The mobility profile defined in the mobility configuration can be specified to model the mobility over the nodes. Random waypoint mobility model is the algorithm that will be used to model mobility for this project and it's the only model provided by OPNET [30] [31]. Which is meaning a node picks random destination and move with the speed that set. After reaching the destination, it stops for a distributed pause time that given. This is repeated until simulation ends. In general, mobile nodes are moving randomly in a network and make random destinations. Furthermore, random mobility model is more suitable for simulation studies. Therefore, mobility configuration is chosen form object palette and dragged to the network. For our work, we put the size of moving area to (500) meter, the same size of our network. And, we make the moving speed measured in (meters/seconds) to (1.3), in order to be comparable to natural human movement. The following Figure 4.6 shows our setup.



(Mobility) Attributes

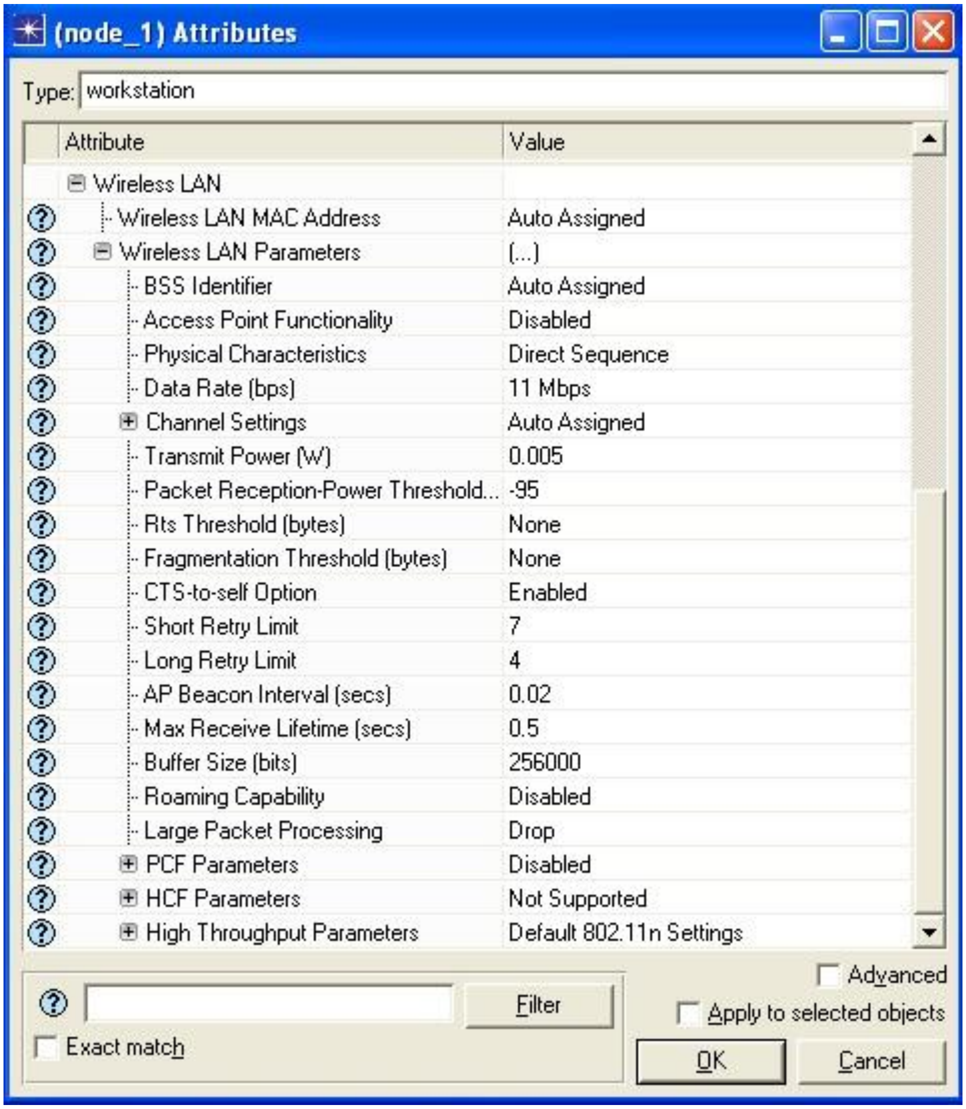
Type: Utilities

Attribute	Value
name	Mobility configuration
Mobility Modeling Status	Enabled
Random Mobility Profiles	(...)
Number of Rows	1
Default Random Waypoint	
Profile Name	Default Random Waypoint
Mobility Model	Random Waypoint
Random Waypoint Parameters	(...)
Mobility Domain Name	Not Used
x_min (meters)	0.0
y_min (meters)	0.0
x_max (meters)	500
y_max (meters)	500
Speed (meters/seconds)	constant (1.3)
Pause Time (seconds)	constant (100)
Start Time (seconds)	constant (10)
Stop Time (seconds)	End of Simulation
Animation Update Frequency (se...	1.0
Record Trajectory	Disabled

Figure 4.6: Mobility Configuration Attributes

4.3.6 Wireless LAN Workstation

Form object palette, we select WLAN wkstn and drag it on network. After we double click on the node we put, a new window will pop as shown in the following Figure 4.7.



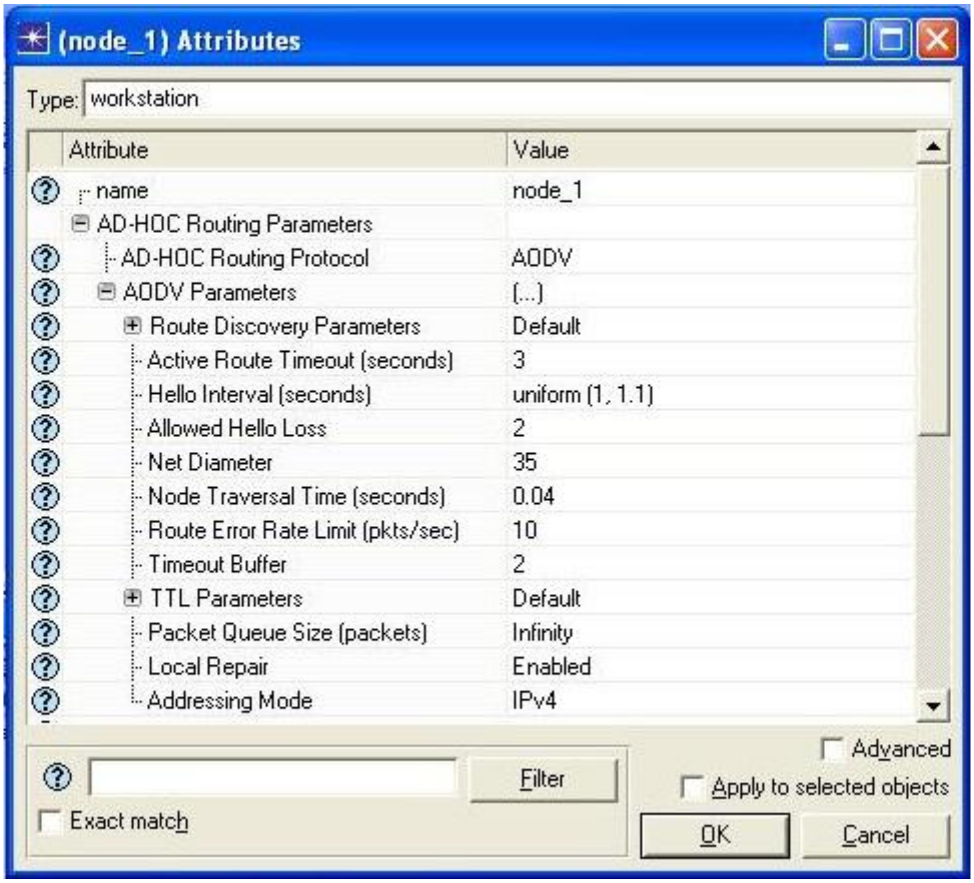
Type: workstation

Attribute	Value
⊟ Wireless LAN	
Wireless LAN MAC Address	Auto Assigned
⊟ Wireless LAN Parameters	(...)
BSS Identifier	Auto Assigned
Access Point Functionality	Disabled
Physical Characteristics	Direct Sequence
Data Rate (bps)	11 Mbps
⊞ Channel Settings	Auto Assigned
Transmit Power (W)	0.005
Packet Reception-Power Threshold...	-95
Rts Threshold (bytes)	None
Fragmentation Threshold (bytes)	None
CTS-to-self Option	Enabled
Short Retry Limit	7
Long Retry Limit	4
AP Beacon Interval (secs)	0.02
Max Receive Lifetime (secs)	0.5
Buffer Size (bits)	256000
Roaming Capability	Disabled
Large Packet Processing	Drop
⊞ PCF Parameters	Disabled
⊞ HCF Parameters	Not Supported
⊞ High Throughput Parameters	Default 802.11n Settings

Figure 4.7: Wireless LAN Workstation Attribute

Now, we will explain the setups we did on WLAN wkstn. The first thing we have done is determining the routing protocol, which for this project is AODV routing protocol.

For the specific parameters of AODV, we keep it set as a default; Figure 4.8 shows these parameters.



Type: workstation

Attribute	Value
name	node_1
⊟ AD-HOC Routing Parameters	
AD-HOC Routing Protocol	AODV
⊟ AODV Parameters	(...)
⊞ Route Discovery Parameters	Default
Active Route Timeout (seconds)	3
Hello Interval (seconds)	uniform (1, 1.1)
Allowed Hello Loss	2
Net Diameter	35
Node Traversal Time (seconds)	0.04
Route Error Rate Limit (pkts/sec)	10
Timeout Buffer	2
⊞ TTL Parameters	Default
Packet Queue Size (packets)	Infinity
Local Repair	Enabled
Addressing Mode	IPv4

Figure 4.8: AODV Routing Protocol Parameters

4.3.6.1 Physical Characteristics

Based on the value of this attribute, which determines the physical layer technology in use, the WLAN MAC will configure the values of the following protocols parameters as indicated in the IEEE 802.11 WLAN standard:

a) SIFS time.

b) SLOT time.

c) Minimum contention window size.

d) Maximum contention window size.

e) And any other parameter value derived from the values of these parameters (like DIFS).

The value of this attribute also determines the set of available data rates that can be used for the data packet transmissions of the WLAN MAC, which is configured under the sibling attribute "Data Rate".

All WLAN MACs that belong to the same BSS should have the same physical characteristics configuration; otherwise the simulation will terminate with an error message. The only exceptions to this rule are as follows:

1. WLAN MACs that deploy Direct Sequence Spread Spectrum (DSSS) 802.11b and Extended Rate PHY (ERP, 802.11g)

2. WLAN MACs that deploy Direct Sequence Spread Spectrum (802.11b), Extended Rate PHY (ERP, 802.11g) and High throughput PHY operating at 2.4 GHz frequency band (802.11b). In our study, we set Physical characteristics to direct sequence as it looks in Figure 4.8.

4.3.6.2 Data Rate (bps)

Data Rate identifies the data rate that will be used by the MAC to transfer data frames over the physical layer. The set of supported data rates depending on the deployed physical layer technology are specified in IEEE's 802.11, 802.11a, 802.11b, 802.11g and 802.11n standards. The value of the sibling attribute "Physical Characteristics" determines the deployed physical layer technology, and consequently, the set of the data rate values that can be configured under this attribute. For our study, we set the data rate which is measured in bytes to 11 Mbps as shown in Figure 4.7.

4.3.6.3 Buffer Size

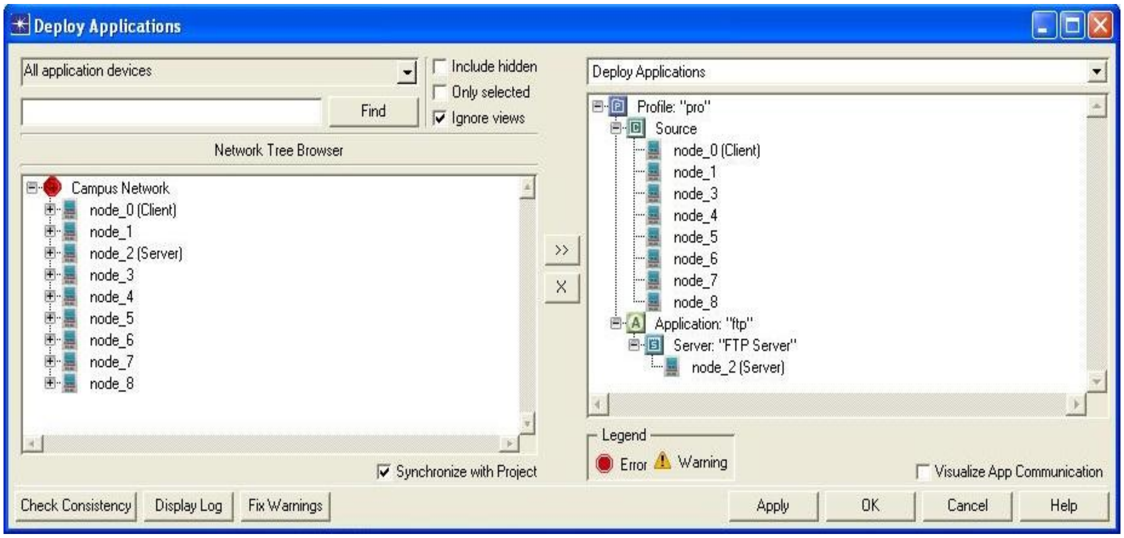
Specifies the maximum size of the higher layer data buffer in bits. Once the buffer limit is reached, the data packets arrived from higher layer will be discarded until some packets are removed from the buffer so that the buffer has some free space to store these new packets**,** we set the buffer size which measured in bits to 256000 as shown in Figure 4.7.

4.3.6.4 Large Packet Processing

The value of this attribute is used if MAC receives a higher layer packet whose size is larger than maximum allowed data size (2304 bytes/18432 bits). This may happen if MAC is running directly below a traffic source module rather than an IP layer; if the value is set to "Drop" then these large packets are dropped as shown in Figure 4.7.

4.3.7 Deploy Application

After we complete our simulation setup which is related to the nodes, profile, application and RX group configuration. , we cannot just after that runs the simulation to collect the data, we should first make the deployment of the nodes and select the one which will be the client to send the data to the request of FTP file and which one will be the server to send response, all other nodes should behave as a intermediate. We do all of that from “Deploy Defined Applications”. This attribute cannot be configured directly. To change the value of this attribute, use the utility, "Protocols / Applications / Deploy Defined Application". The following Figure 4.9 shows the deployment of the nodes.



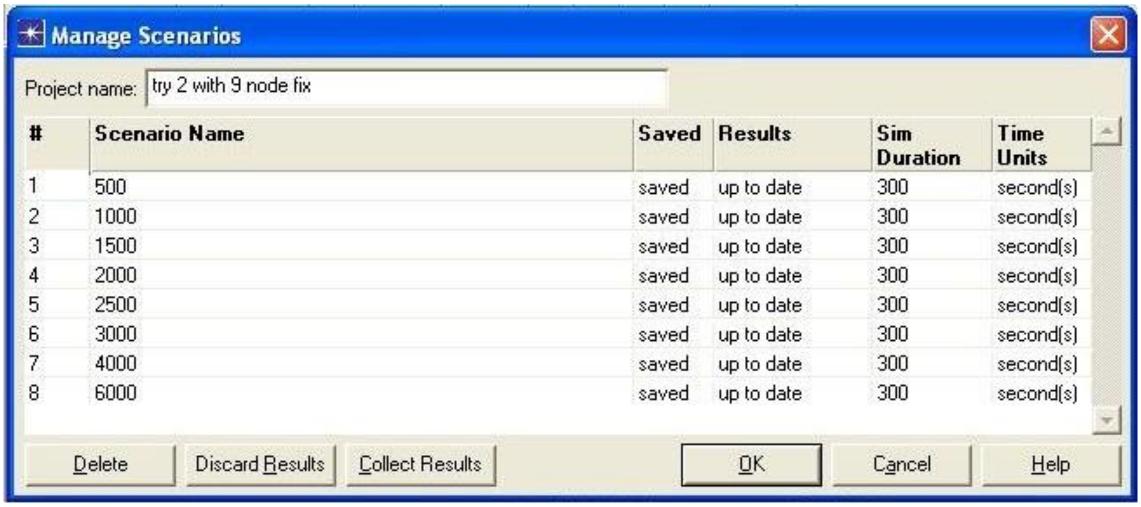
□ Include hidden
□ Only selected
☑ Ignore views
☑ Synchronize with Project
□ Visualize App Communication

```
Campus Network
  node_0 (Client)
  node_1
  node_2 (Server)
  node_3
  node_4
  node_5
  node_6
  node_7
  node_8
```

```
Profile: "pro"
  Source
    node_0 (Client)
    node_1
    node_3
    node_4
    node_5
    node_6
    node_7
    node_8
  Application: "ftp"
    Server: "FTP Server"
      node_2 (Server)
```

Figure 4.9: Deploy Application

Application Deployment dialog box helps in deploying the application in the network. To configure a profile or an application on a node or set of nodes select them in the network tree on the left hand side of the figure. Select the profile or application tier on the right hand side tree. Click the assign (>>) button to deploy the selected set of nodes to the selected tier. To remove the profile or the application from a node, just select it from the right hand side tree. Click the remove (x) button to remove the node from the tier. In the end, we can run our simulation from the manager scenario as shown in Figure 4.10

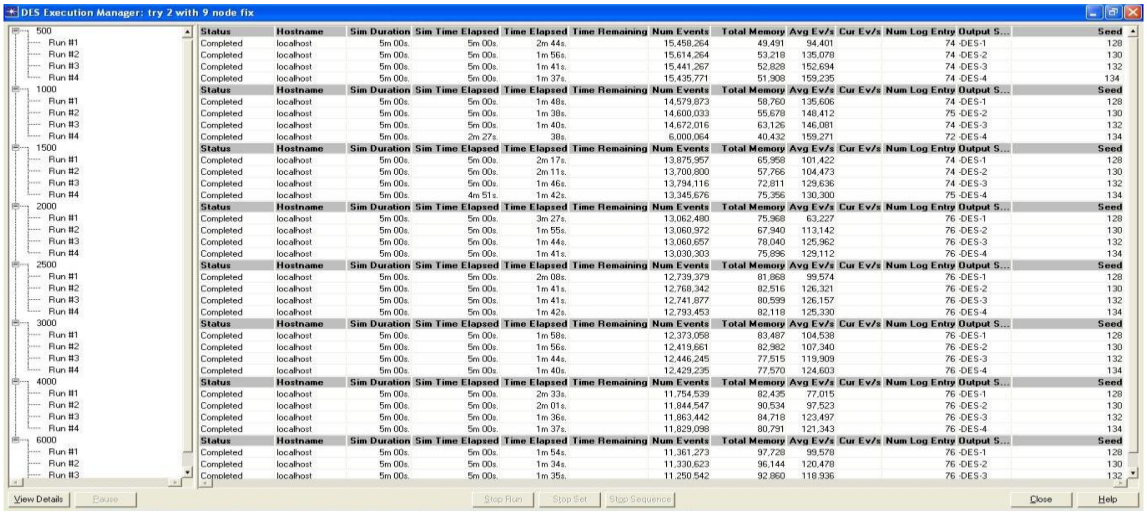


Project name: try 2 with 9 node fix

#	Scenario Name	Saved	Results	Sim Duration	Time Units
1	500	saved	up to date	300	second(s)
2	1000	saved	up to date	300	second(s)
3	1500	saved	up to date	300	second(s)
4	2000	saved	up to date	300	second(s)
5	2500	saved	up to date	300	second(s)
6	3000	saved	up to date	300	second(s)
7	4000	saved	up to date	300	second(s)
8	6000	saved	up to date	300	second(s)

Figure 4.10: Manager Scenarios

After that, we will click on OK button and run the eight scenarios which each one of them has four duplicate and that mean it will run four times for each data size as shown in Figure 4.11.



500	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	2m 44s.		15,458,264	49,491	94,401		74	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 56s.		15,614,264	53,218	135,078		74	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 41s.		15,441,267	52,828	152,694		74	-DES-3	132
Run #4	Completed	localhost	5m 00s.	5m 00s.	1m 37s.		15,435,771	51,908	159,235		74	-DES-4	134
1000	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	1m 48s.		14,579,873	58,760	135,606		74	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 38s.		14,600,033	55,678	148,412		75	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 40s.		14,672,016	63,126	146,081		74	-DES-3	132
Run #4	Completed	localhost	5m 00s.	2m 27s.	38s.		6,000,064	40,432	159,271		72	-DES-4	134
1500	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	2m 17s.		13,875,957	65,958	101,422		74	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	2m 11s.		13,700,800	57,766	104,473		74	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 46s.		13,794,116	72,811	129,636		74	-DES-3	132
Run #4	Completed	localhost	5m 00s.	4m 51s.	1m 42s.		13,345,676	75,356	130,300		75	-DES-4	134
2000	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	3m 27s.		13,062,480	75,968	63,227		76	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 55s.		13,060,972	67,940	113,142		76	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 44s.		13,060,657	78,040	125,962		76	-DES-3	132
Run #4	Completed	localhost	5m 00s.	5m 00s.	1m 41s.		13,030,303	75,896	129,112		76	-DES-4	134
2500	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	2m 08s.		12,739,379	81,868	99,574		76	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 41s.		12,768,342	82,516	126,321		76	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 41s.		12,741,877	80,599	126,157		76	-DES-3	132
Run #4	Completed	localhost	5m 00s.	5m 00s.	1m 42s.		12,793,453	82,118	125,330		76	-DES-4	134
3000	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	1m 58s.		12,373,058	83,487	104,538		76	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 56s.		12,419,661	82,982	107,340		76	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 44s.		12,446,245	77,515	119,909		76	-DES-3	132
Run #4	Completed	localhost	5m 00s.	5m 00s.	1m 40s.		12,429,235	77,570	124,603		76	-DES-4	134
4000	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	2m 33s.		11,754,539	82,435	77,015		76	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	2m 01s.		11,844,547	90,534	97,523		76	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 36s.		11,863,442	84,718	123,497		76	-DES-3	132
Run #4	Completed	localhost	5m 00s.	5m 00s.	1m 37s.		11,829,098	80,791	121,343		76	-DES-4	134
6000	Status	Hostname	Sim Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	Total Memory	Avg Ev/s	Cur Ev/s	Num Log Entry	Output S...	Seed
Run #1	Completed	localhost	5m 00s.	5m 00s.	1m 54s.		11,361,273	97,728	99,578		76	-DES-1	128
Run #2	Completed	localhost	5m 00s.	5m 00s.	1m 34s.		11,330,623	96,144	120,478		76	-DES-2	130
Run #3	Completed	localhost	5m 00s.	5m 00s.	1m 35s.		11,250,542	92,860	118,936		76	-DES-3	132

Figure 4.11: Discrete Event Simulation Execution Manager

4.4 Simulation with Different Ad hoc Network Scenarios and Results

The results of our simulation are gained through a number of scenarios. In our simulation study, there are a number scenarios established on the number of nodes, various data sizes and speed as performed with performance metrics route discovery time, total route request sent, download response time, upload response time and packet delivery ratio for AODV routing protocol.

4.4.1 Investigation of Different Number of Nodes and Message Size

In our first scenario, we prepare 9 fix nodes, where we get them from object palette of OPNET 17.1 and pasted all of them randomly in the workstation and choose AODV routing protocol for all of the nodes. After, we complete the setting of application configuration, profile configuration and RX group configuration from object palette to the workstation; the settings have to be done according to the requirements. The FTP is selected as traffic, and by default, it has low load with default set of 1000 bytes message size, medium load with 5000 bytes message size and high load with 50000 bytes message size. In our study we use various message sizes, which is ranging between low and medium load, FTP file size is set to eight values, starting from 500, 1000, 1500, 2000, 2500, 3000, 4000 and 6000 bytes. We replicate the run for all scenarios four times and collect the results which we used the average values.

We repeat all the steps above for the other scenarios but we change the number of the nodes, which will be 30, 60 and 90 fix nodes. After we complete running each of the simulation scenarios of the fix nodes, and save the result, we start all over again for the mobile nodes scenarios, for which we keep the same setting of data size with application configuration, profile configuration and RX group configuration. Also we add the mobility configuration inserted in to workstation from object palette. In our simulation, we set the speed to (1.3) meters/seconds so it will be similar to human moving speed. All our settings are illustrated in the tables below:

Table 2: General settings for our simulation

Characteristics	**Value**
Number of nodes	9, 30, 60 and 90
File (Message) size	500, 1000, 1500, 2000, 2500, 3000, 4000 and 6000 bytes
Protocol	AODV
Simulation run time	300 seconds
Simulation area size	500 meter * 500 meter

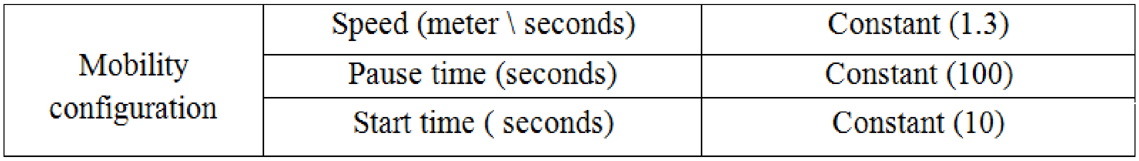
Table 3: Application configuration settings

Application configuration	FTP	Low Load
	Inter request time (seconds)	Constant (0.1)

Table 4: Profile configuration settings

Profile configuration	Start time offset (seconds)	Constant (10)
	Duration	End of Last task
	Start time (seconds)	Constant (30)
	Duration	End of Simulation

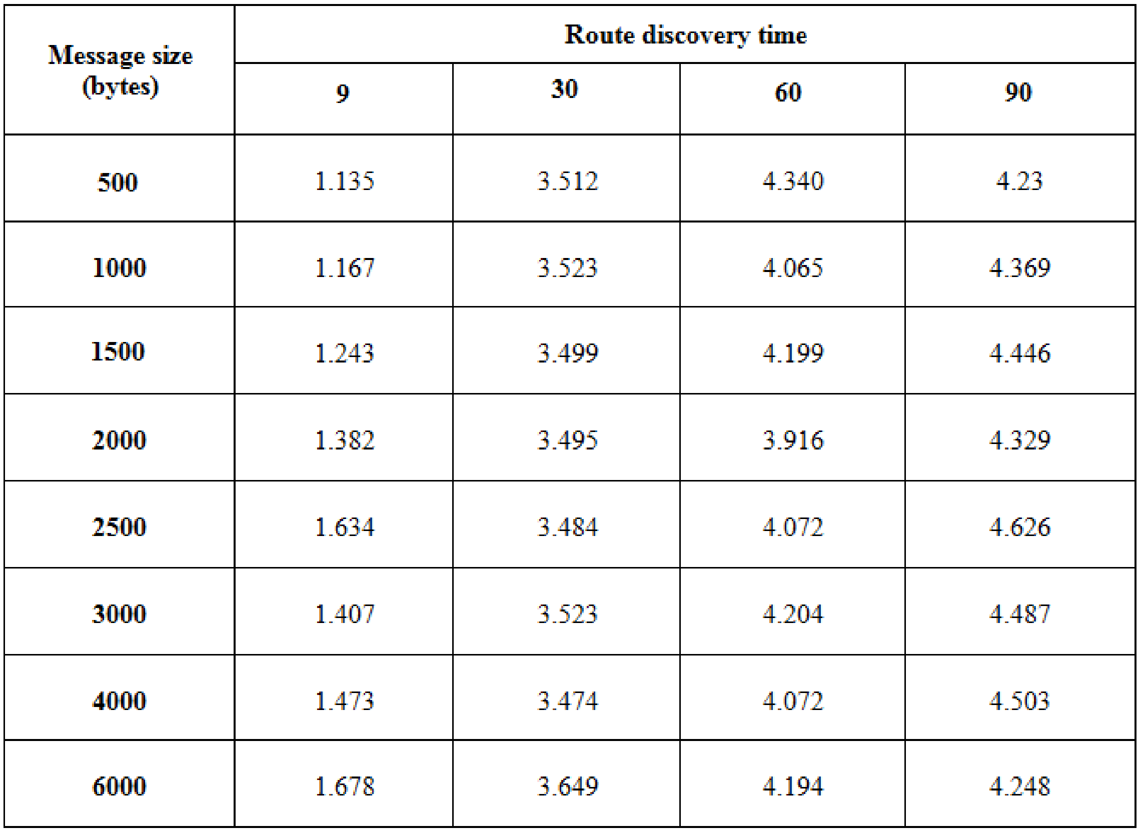
Table 5: Mobility configuration settings



Mobility configuration	Speed (meter \ seconds)	Constant (1.3)
	Pause time (seconds)	Constant (100)
	Start time (seconds)	Constant (10)

The results of our simulation are obtained in the shape of graphs which were displayed as sample mean of 4 runs to get the average value. We used network sizes such as 9, 30, 60 and 90 nodes to see the scalability of the network with performance metrics, like what it had been done in [32] [33].

Table 6: Simulation results of average route discovery time for fixed nodes using AODV routing protocol for FTP application.



Message size (bytes)	Route discovery time			
	9	30	60	90
500	1.135	3.512	4.340	4.23
1000	1.167	3.523	4.065	4.369
1500	1.243	3.499	4.199	4.446
2000	1.382	3.495	3.916	4.329
2500	1.634	3.484	4.072	4.626
3000	1.407	3.523	4.204	4.487
4000	1.473	3.474	4.072	4.503
6000	1.678	3.649	4.194	4.248

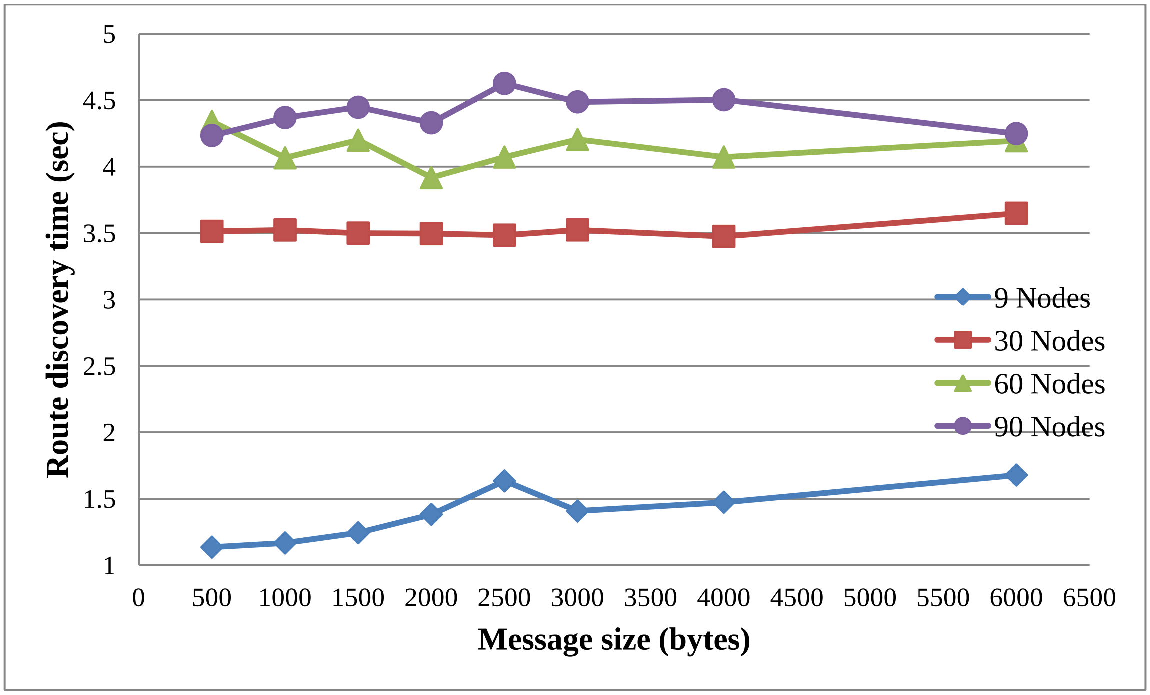
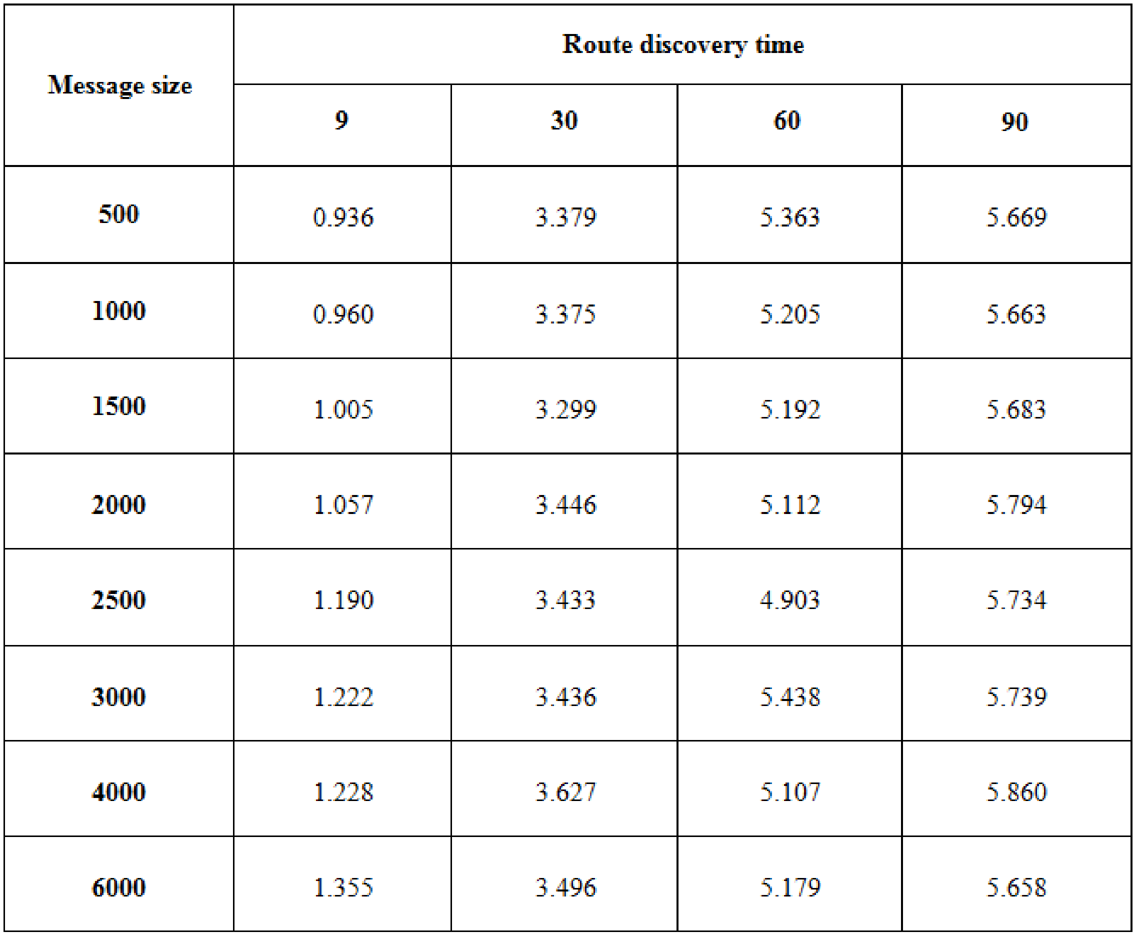


Figure 4.12: Route discovery time for fix nodes

Figure 4.12 shows the measurement of route discovery time for different number of nodes and message sizes. When we change message size from 500 bytes to 6000 bytes, it is observed that there is no effects of message size on the route discovery time since number of packets that are sent for route discovery time purpose are same for all message sizes and for route discovery mechanism is initiated by AODV protocol when there is packet to send, and for all message sizes same number of packets are transmitted for route discovery time. We observe that with number of nodes are increasing; route discovery time every time is increasing since more numbers of nodes are participating for discovering. For the number of nodes 30, 60 and 90, we have slight increased in the route discovery time, but with less node number such 9, route discovery time is quite low due to the less number of used intermediate nodes in order to arrive to the destination.

Table 7: Simulation results of average route discovery time for mobile nodes using AODV routing protocol for FTP application.



Message size	Route discovery time			
	9	30	60	90
500	0.936	3.379	5.363	5.669
1000	0.960	3.375	5.205	5.663
1500	1.005	3.299	5.192	5.683
2000	1.057	3.446	5.112	5.794
2500	1.190	3.433	4.903	5.734
3000	1.222	3.436	5.438	5.739
4000	1.228	3.627	5.107	5.860
6000	1.355	3.496	5.179	5.658

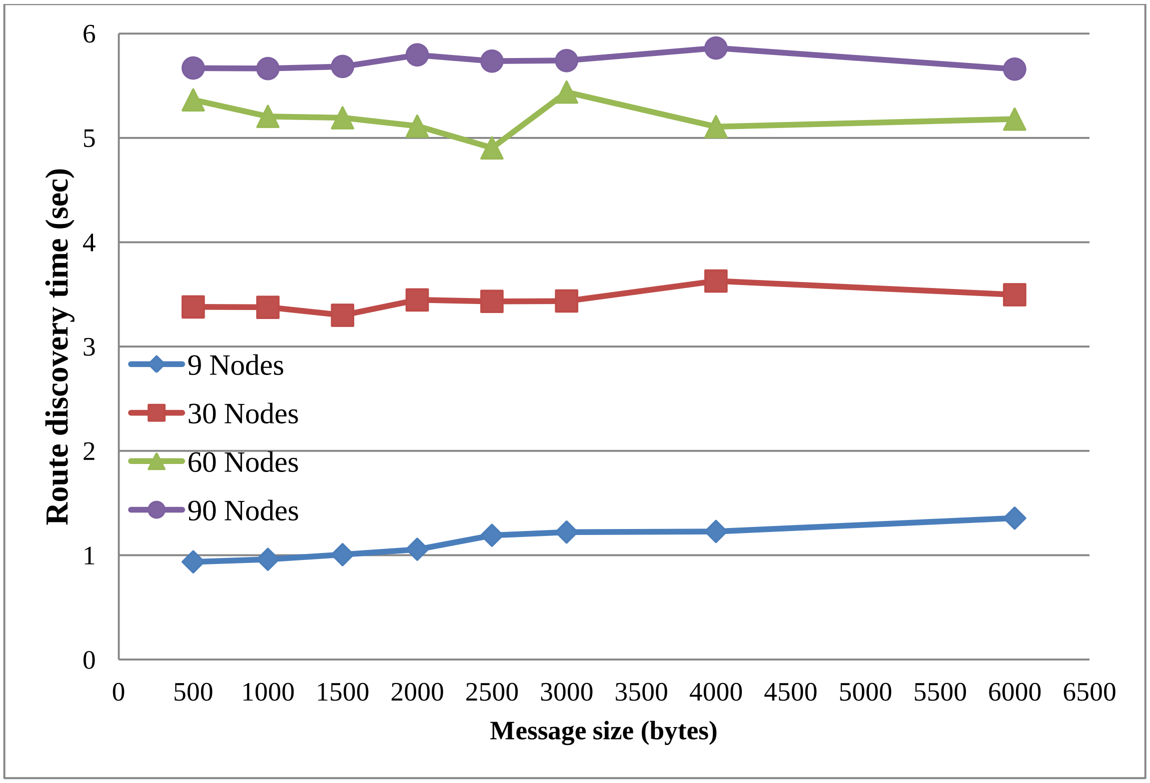


Figure 4.13: Route discovery time for mobile nodes

Figure 4.13 shows the measurement of route discovery time for mobile nodes for different number of nodes and message size. When we change message size from 500 bytes to 6000 bytes, there is no valuable effect on route discovery time. Here, the 9 mobile nodes have the lowest values of route discovery time due to the less number of used intermediate nodes in order to arrive the destination. When we compared mobile and fixed case, we observed that route discovery time is slightly increasing for large number of nodes (60 and 90) in mobile case due to the topology change in the network. We can say that for large number of nodes, discovering a route will take same time in a dynamic network.

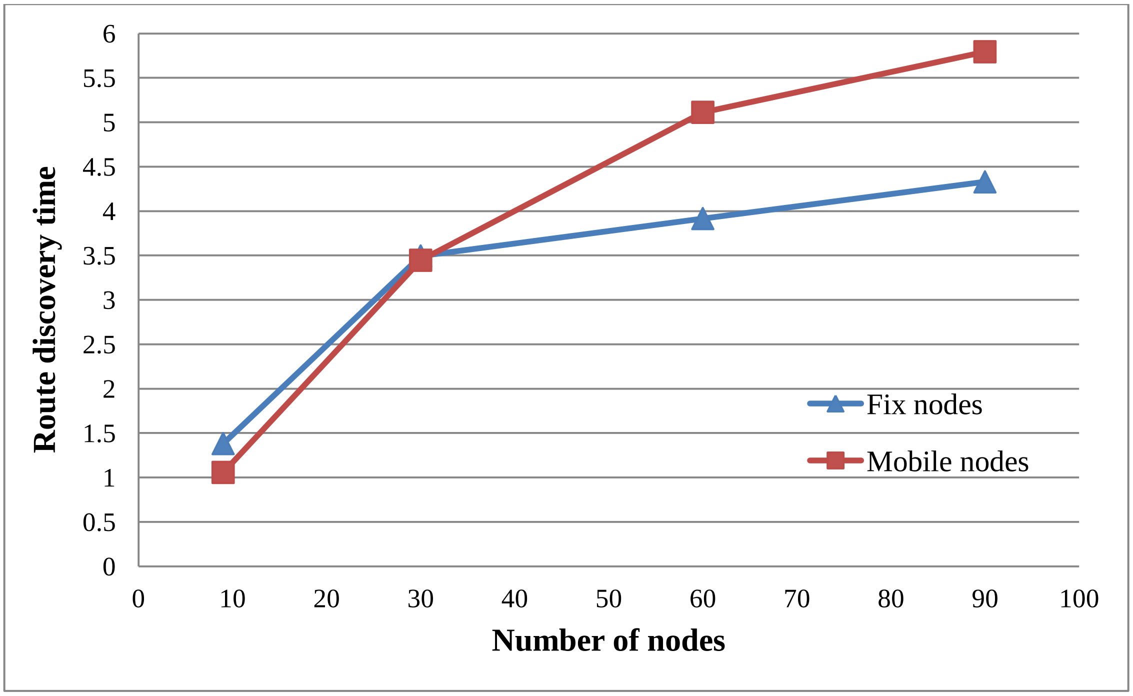


Figure 4.14: Route discovery time for fix and mobile nodes with 2000 bytes message size

Figure 4.14 shows the measurement of route discovery time for fix and mobile nodes for different number of nodes with 2000 bytes message size. For the reason of there is no effects on message size on route discovery time on both fix and mobile nodes, this measurement show the effects of number of nodes on route discovery time. Here, both of fix and mobile nodes have increase in values of route discovery time due to the increasing number of intermediate nodes between client and server. Also, the mobile scenario is having higher values in route discovery time from the fix scenario due to the topology change of the network.

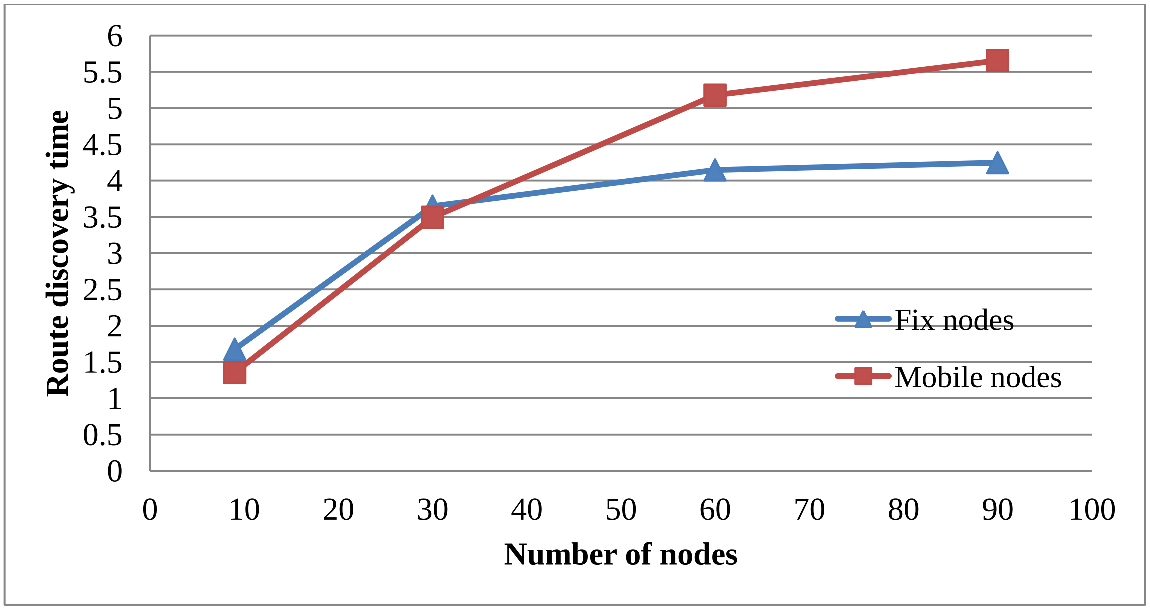
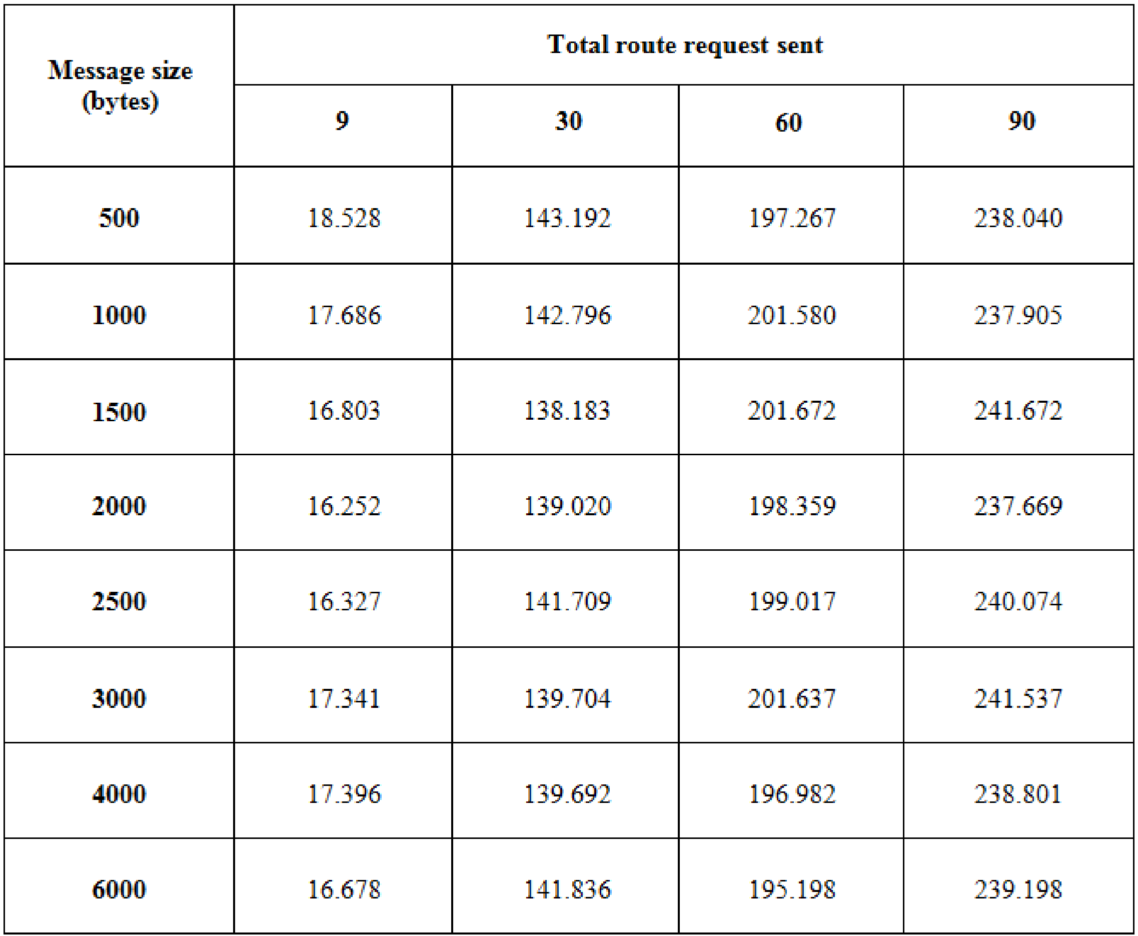


Figure 4.15: Route discovery time for fix and mobile nodes with 6000 bytes message size

Figure 4.15 shows the measurement of route discovery time for fix and mobile nodes for different number of nodes with 6000 bytes message size. Which there no different with changing message size on the route discovery time due to there is no effect of message size on route discovery time. This measurement shows the effects of number of nodes on route discovery time. Here, both of fix and mobile nodes have increase in values of route discovery time due to the increasing number of intermediate nodes between client and server. Also, the mobile scenario is having higher values in route discovery time from the fix scenario due to the topology change of the network.

Table 8: Simulation results of average total route request sent for fixed nodes using AODV routing protocol for FTP application.



Message size (bytes)	Total route request sent			
	9	30	60	90
500	18.528	143.192	197.267	238.040
1000	17.686	142.796	201.580	237.905
1500	16.803	138.183	201.672	241.672
2000	16.252	139.020	198.359	237.669
2500	16.327	141.709	199.017	240.074
3000	17.341	139.704	201.637	241.537
4000	17.396	139.692	196.982	238.801
6000	16.678	141.836	195.198	239.198

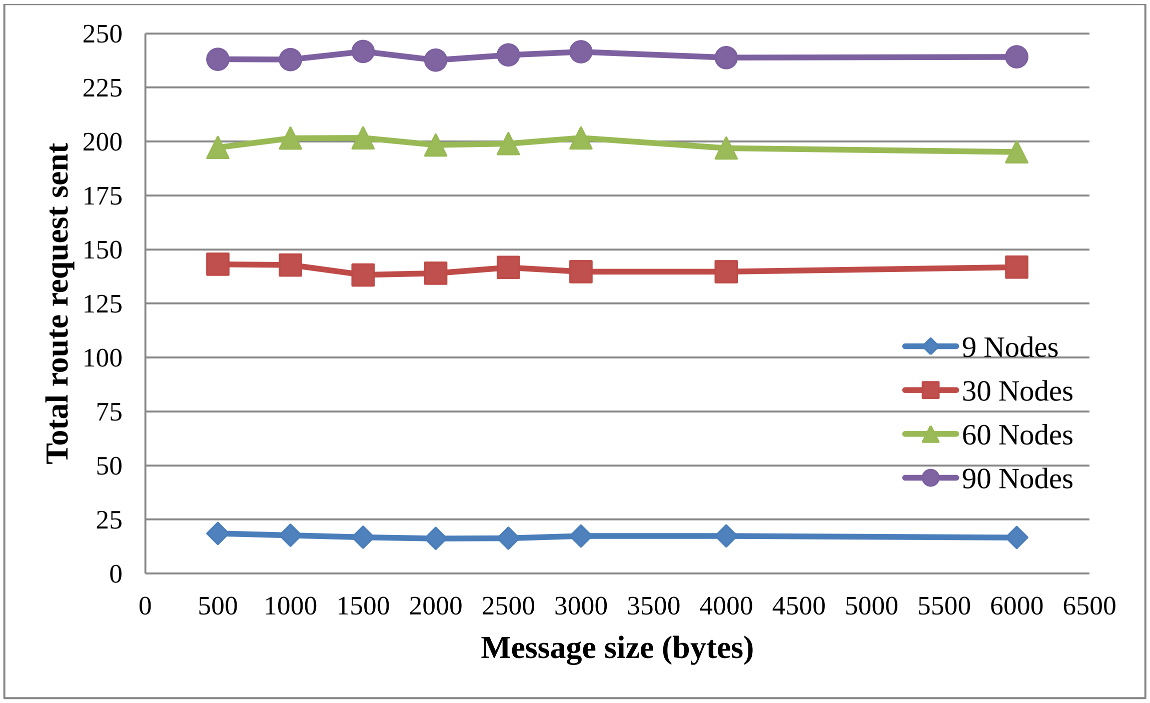
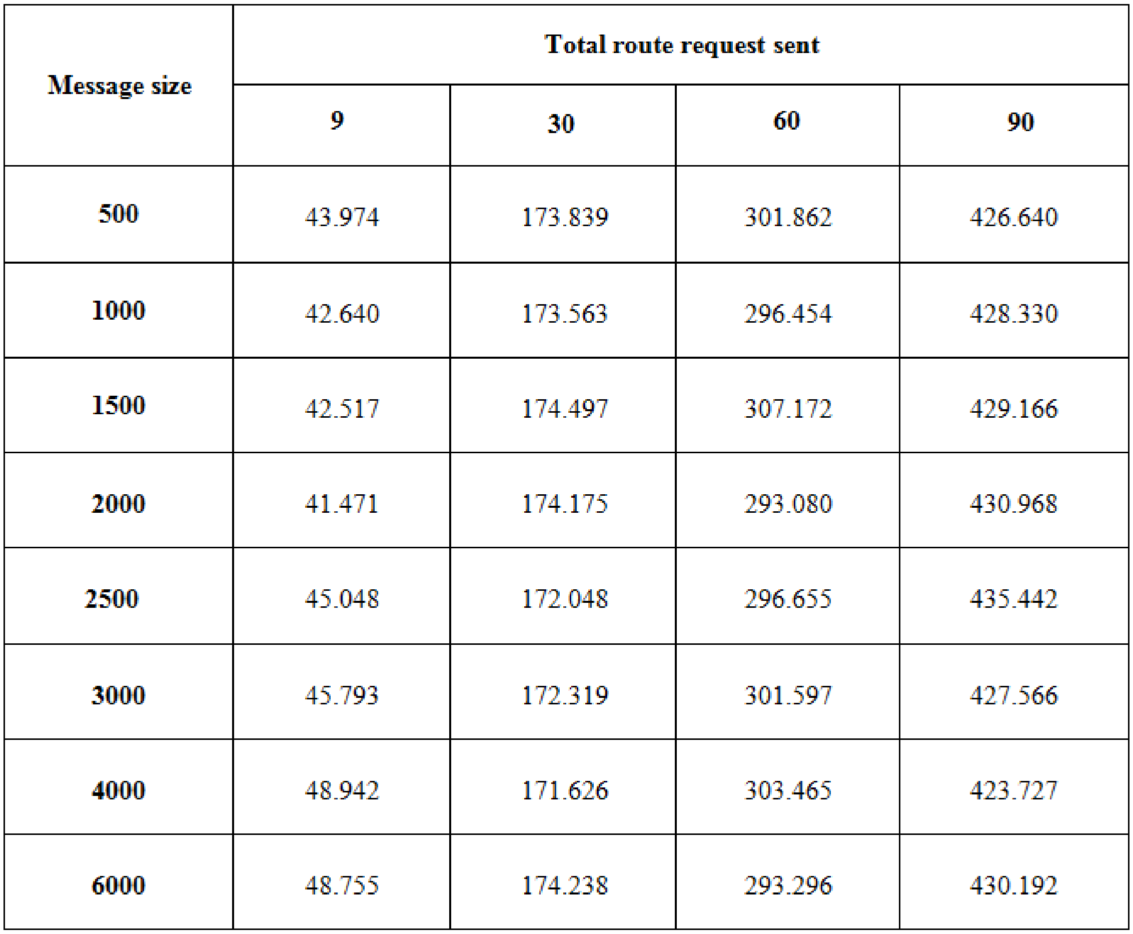


Figure 4.16: Total route request sent for fix nodes

Figure 4.16 shows the measurements of total route request sent for different number of nodes and message size. When we change message size from 500 bytes to 6000 bytes, there is no valuable effect on the total route request sent. By increasing the number of nodes 9, 30, 60 and 90, the total route requests sent are taking more values because of increasing the amount of request sent by intermediate nodes in flooding process, during discovery of route.

Table 9: Simulation results of average total route request sent for mobile nodes using AODV routing protocol For FTP application.



Message size	Total route request sent			
	9	30	60	90
500	43.974	173.839	301.862	426.640
1000	42.640	173.563	296.454	428.330
1500	42.517	174.497	307.172	429.166
2000	41.471	174.175	293.080	430.968
2500	45.048	172.048	296.655	435.442
3000	45.793	172.319	301.597	427.566
4000	48.942	171.626	303.465	423.727
6000	48.755	174.238	293.296	430.192

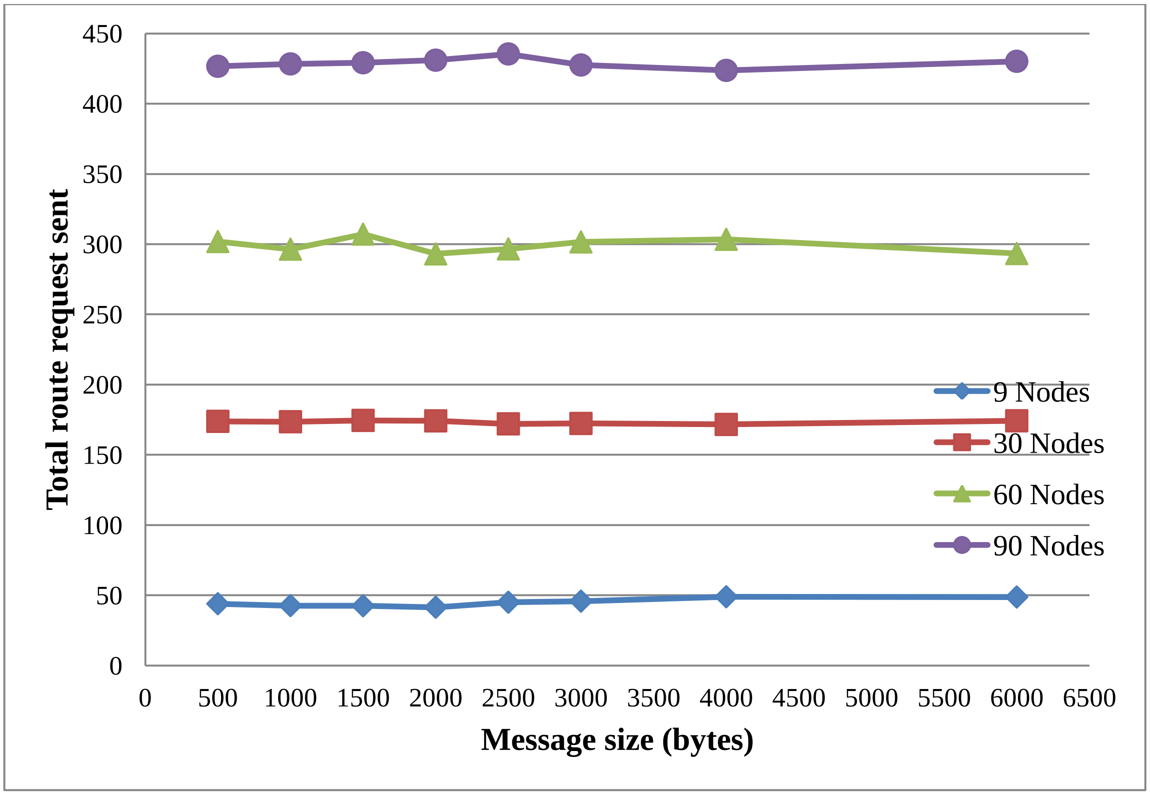


Figure 4.17: Total route request sent for mobile nodes

Figure 4.17, shows the measurement of total route request sent for different number of mobile nodes and message size. When we change message size from 500 bytes to 6000 bytes, there is no valuable effect on total route request sent. By increasing the number of nodes 9, 30, 60 and 90, the total route requests sent are taking more values because of increasing the amount of request sent by intermediate nodes in flooding process, during discovery of route.

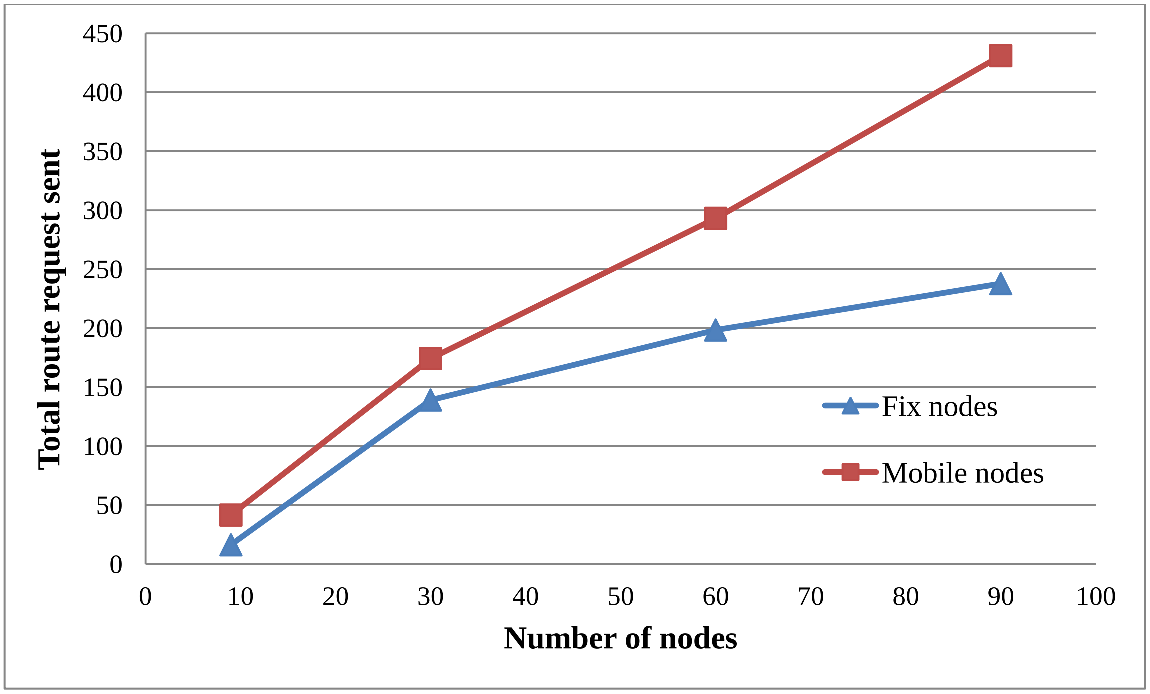


Figure 4.18: Total route request sent for fix and mobile node with 2000 bytes message size

Figure 4.18 shows the measurement of total route request sent for fix and mobile nodes for different number of nodes with 2000 bytes message size. Because of there is no effects on message size on total route request sent on both fix and mobile nodes. Here, both the fix and the mobile nodes are increasing in total route request sent by increasing number of nodes in the network, because of increasing the amount of request sent by intermediate nodes in flooding process, during discovery of route. In mobile nodes case, the values of total route request sent is higher due to more route request are been sent by nodes since breakage may occur in the route of a dynamic network.

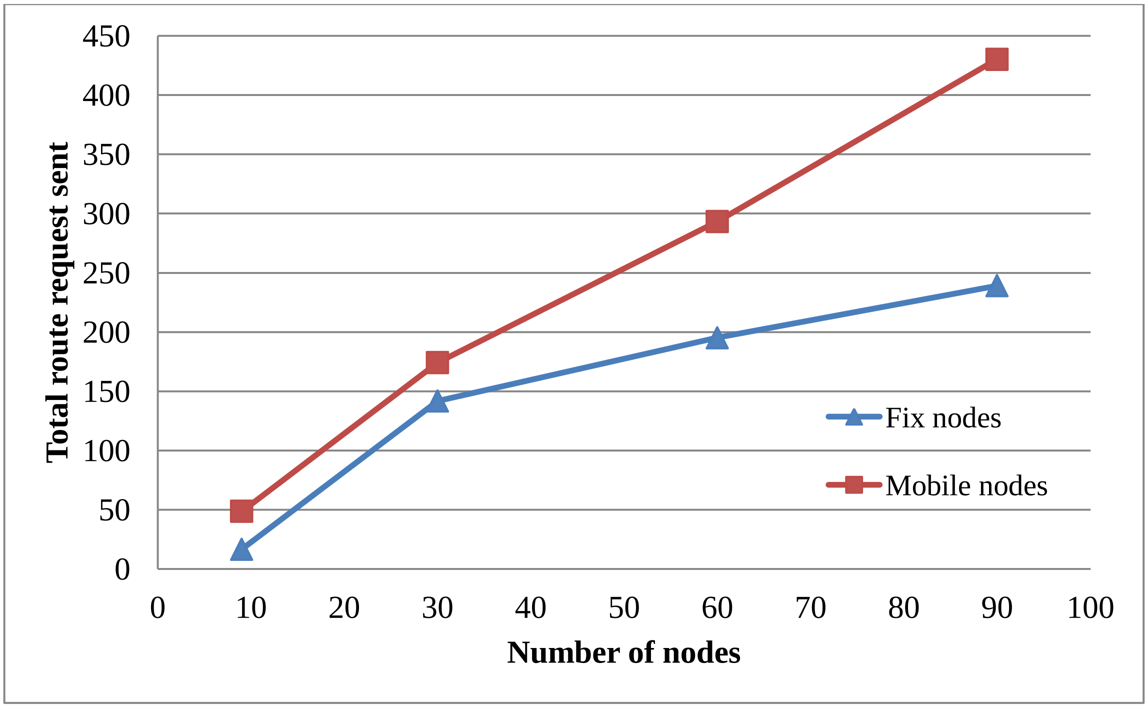
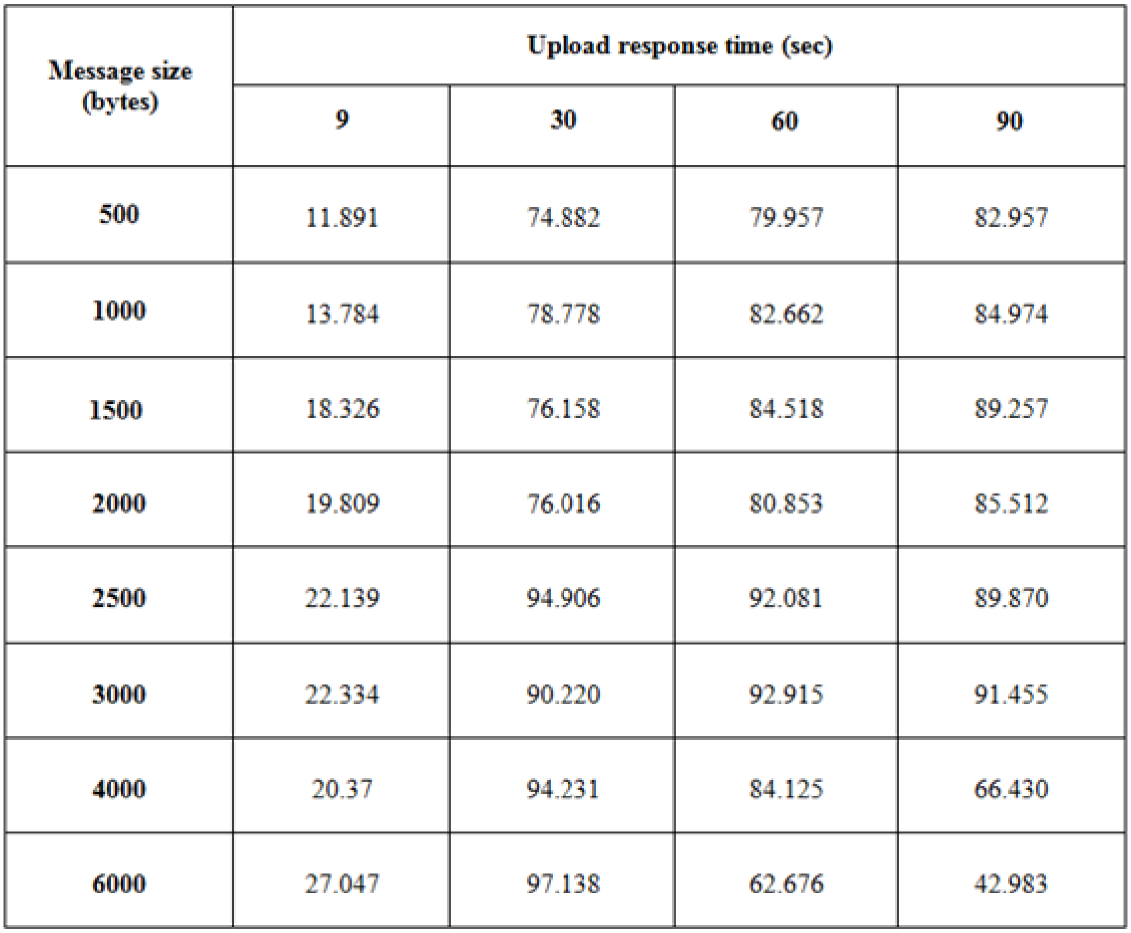


Figure 4.19: Total route request sent for fix and mobile node with 6000 bytes message size

Figure 4.19 shows the measurement of total route request sent for fix and mobile nodes for different number of nodes with 6000 bytes message size. Which there no different with changing message size on total route request sent due to there is no effect of message size on total route request sent. Here, both the fix and the mobile nodes are increasing in total route request sent by increasing number of nodes in the network, because of increasing the amount of request sent by intermediate nodes in flooding process, during discovery of route. In mobile nodes case, the values of total route request sent is higher due to more route request are been sent by nodes when breakage occur in the route.

Table 10: Simulation results of average upload response time (sec) for fixed nodes using AODV routing protocol for FTP application.



Message size (bytes)	Upload response time (sec)			
	9	30	60	90
500	11.891	74.882	79.957	82.957
1000	13.784	78.778	82.662	84.974
1500	18.326	76.158	84.518	89.257
2000	19.809	76.016	80.853	85.512
2500	22.139	94.906	92.081	89.870
3000	22.334	90.220	92.915	91.455
4000	20.37	94.231	84.125	66.430
6000	27.047	97.138	62.676	42.983

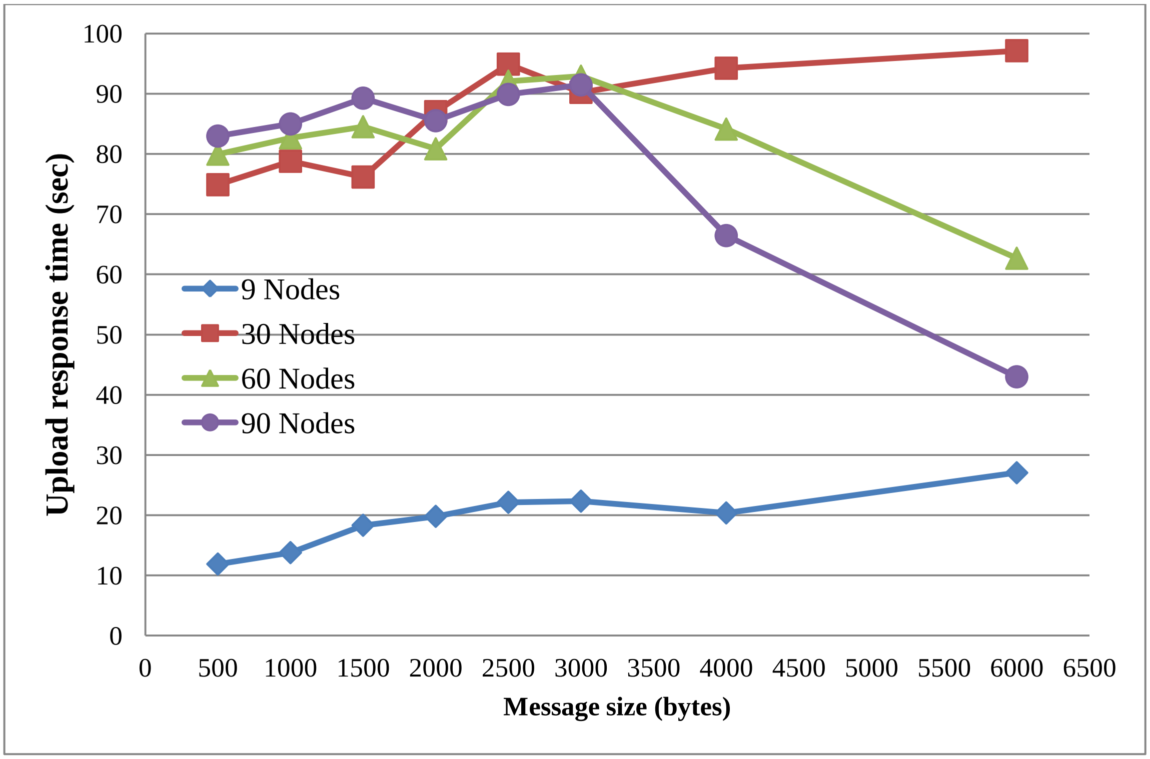
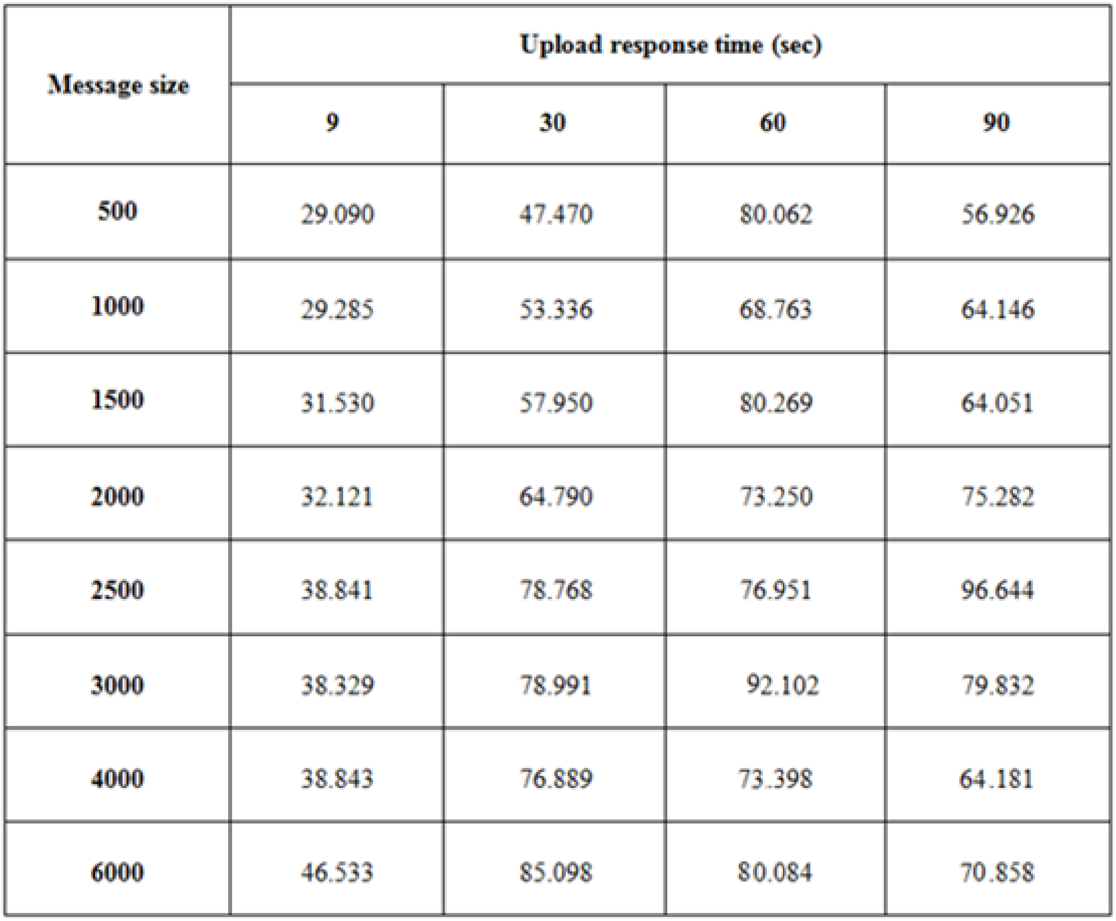


Figure 4.20: Upload response time (sec) for fix nodes

Figure 4.20 shows the measurement of upload response time (sec) for different number of nodes and message size, when the message size increased from 500 bytes until it reach 6000 bytes. When the number of nodes are low 9; client will receive the response file from the server in less time than other number of nodes. When nodes number increase (60 and 90), and the message size also increase such as 4000 and 6000, the network will be congested which packet dropping also increase. It will lead to lower value of the upload response time.

Table 11: Simulation results of average upload response time (sec) for mobile nodes using AODV routing protocol for FTP application.



Message size	Upload response time (sec)			
	9	30	60	90
500	29.090	47.470	80.062	56.926
1000	29.285	53.336	68.763	64.146
1500	31.530	57.950	80.269	64.051
2000	32.121	64.790	73.250	75.282
2500	38.841	78.768	76.951	96.644
3000	38.329	78.991	92.102	79.832
4000	38.843	76.889	73.398	64.181
6000	46.533	85.098	80.084	70.858

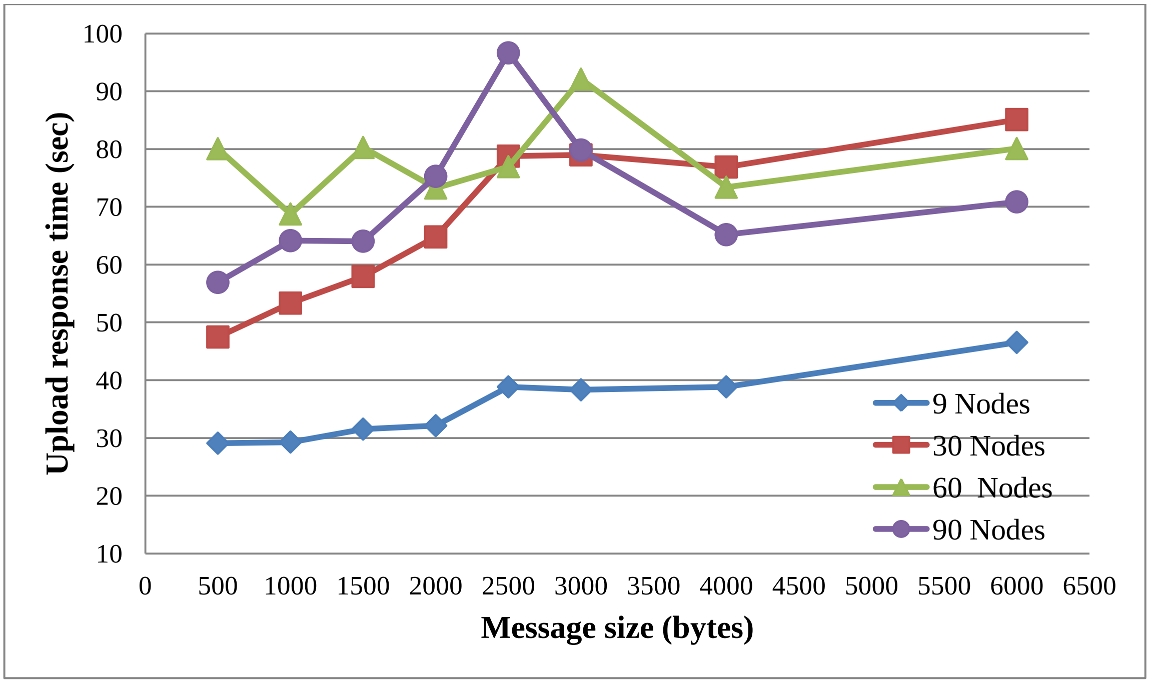


Figure 4.21: Upload response time (sec) for mobile nodes

Figure 4.21 shows the measurement in upload response time (sec) for mobile nodes, which we have different number of mobile nodes and message size, when the message size increased from 500 bytes until it reaches 6000 bytes. Here, there is a valuable effect of upload response time with message size. When the number of nodes are low; client will receive the response file from the server in less time than other number of nodes. When the nodes number increase in the network with the increase of message size, the upload response time will increase due to topology changing in the network.

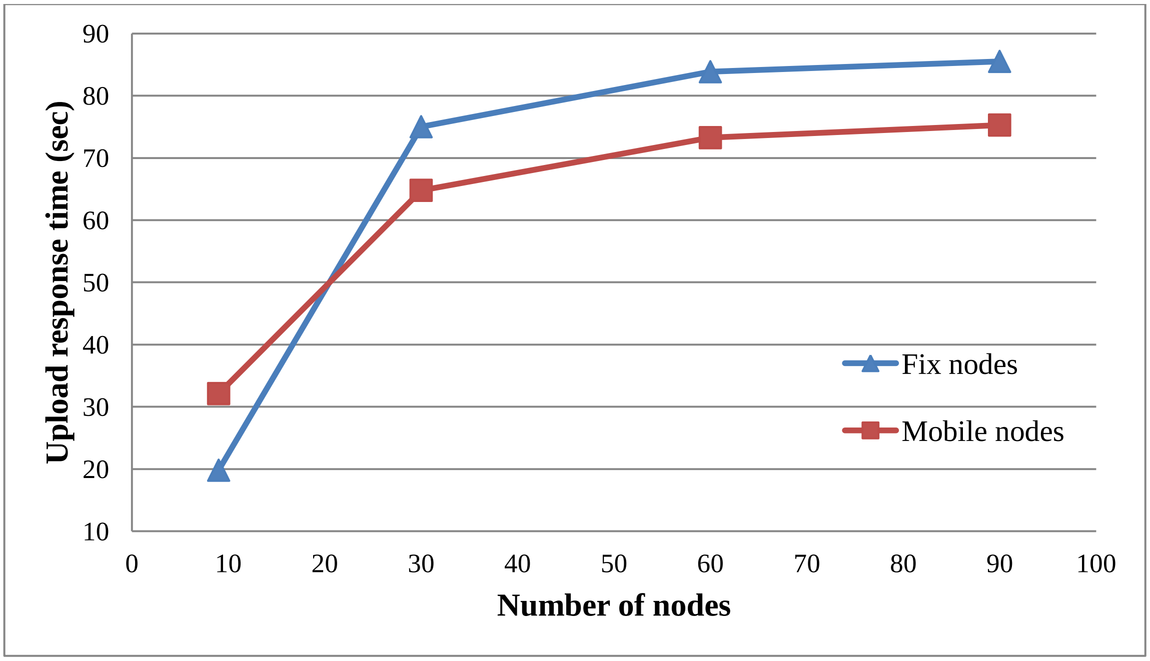


Figure 4.22: Upload response time (sec) for fix and mobile nodes with 2000 bytes message size

Figure 4.22 shows the measurement of upload response time (sec) for fix and mobile nodes for different number of nodes with 2000 bytes message size. Values of upload response time are increasing in both fixed and mobile nodes scenarios by increasing the number of nodes in the network due to the increasing of intermediate nodes in the network. In mobile scenario, the values of upload response time became less when the number of node is increasing due to topology changes.

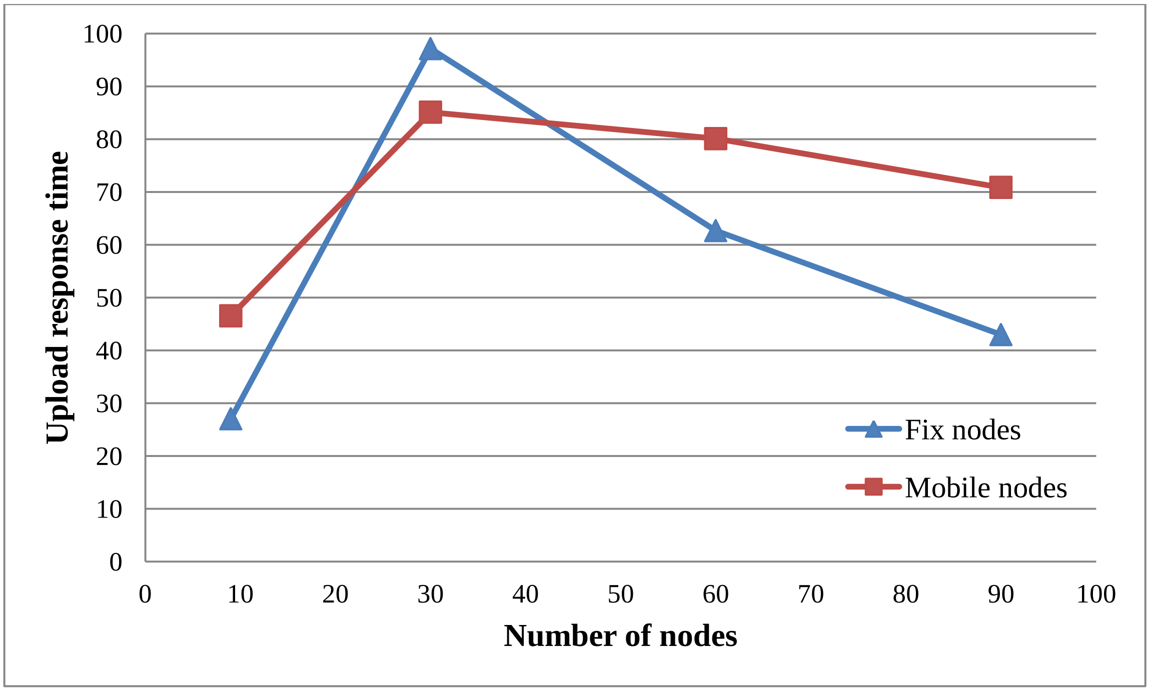
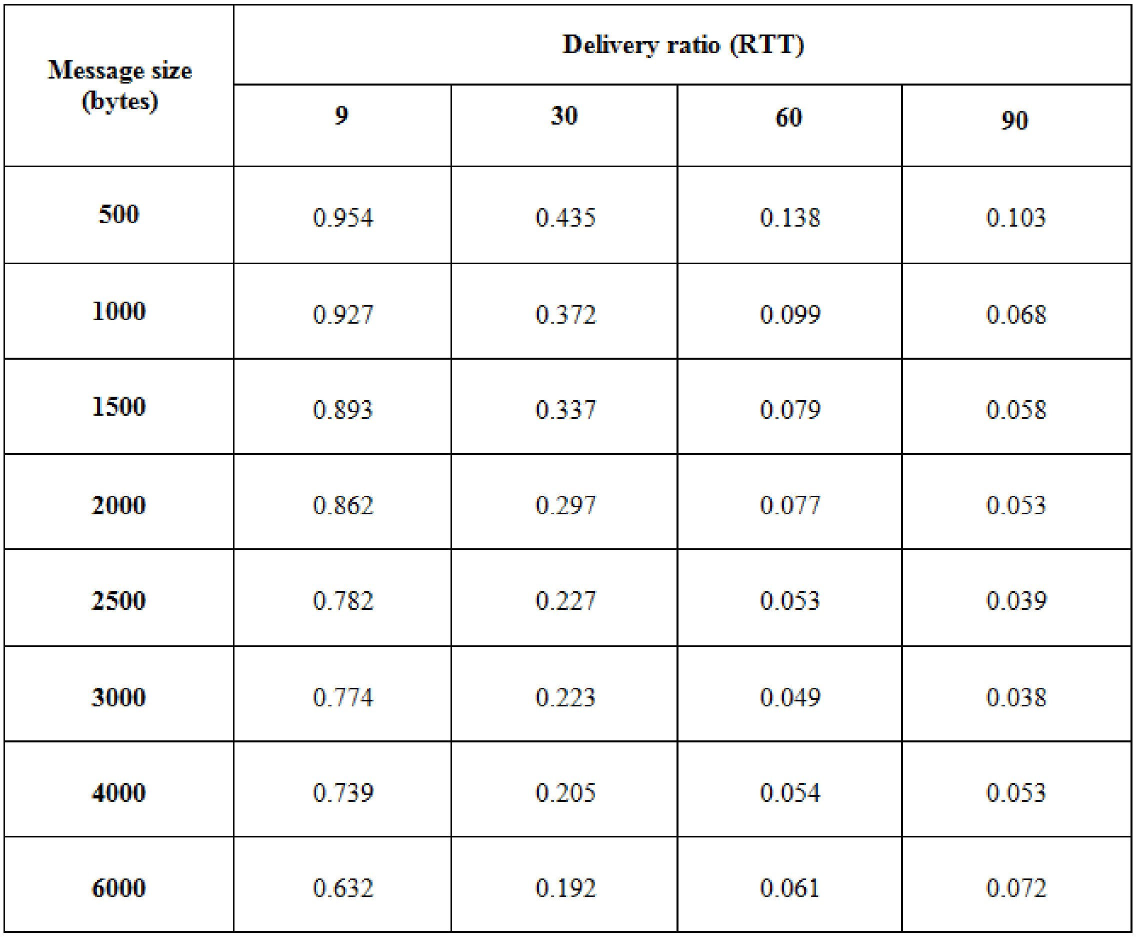


Figure 4.23: Upload response time (sec) for fix and mobile nodes with 6000 bytes message size

Figure 4.23 shows the measurement of upload response time (sec) for fix and mobile nodes for different number of nodes with 6000 bytes message size. Values of upload response time are increasing in both fixed and mobile nodes scenarios when number of nodes is low, but when the number of nodes is increase in the network, the network will be congested which packet dropping also increase. It will lead to lower value of upload response time.

Table 12: Simulation results of average delivery ratio for fixed nodes using AODV routing protocol for FTP application.



Message size (bytes)	Delivery ratio (RTT)			
	9	30	60	90
500	0.954	0.435	0.138	0.103
1000	0.927	0.372	0.099	0.068
1500	0.893	0.337	0.079	0.058
2000	0.862	0.297	0.077	0.053
2500	0.782	0.227	0.053	0.039
3000	0.774	0.223	0.049	0.038
4000	0.739	0.205	0.054	0.053
6000	0.632	0.192	0.061	0.072

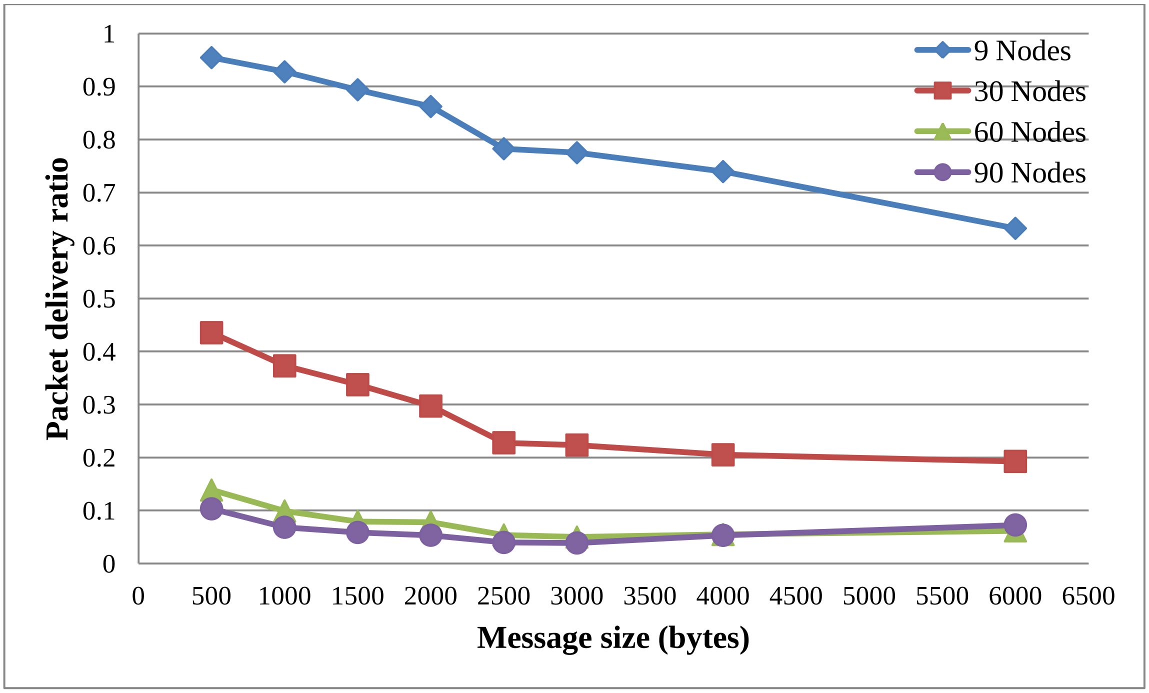
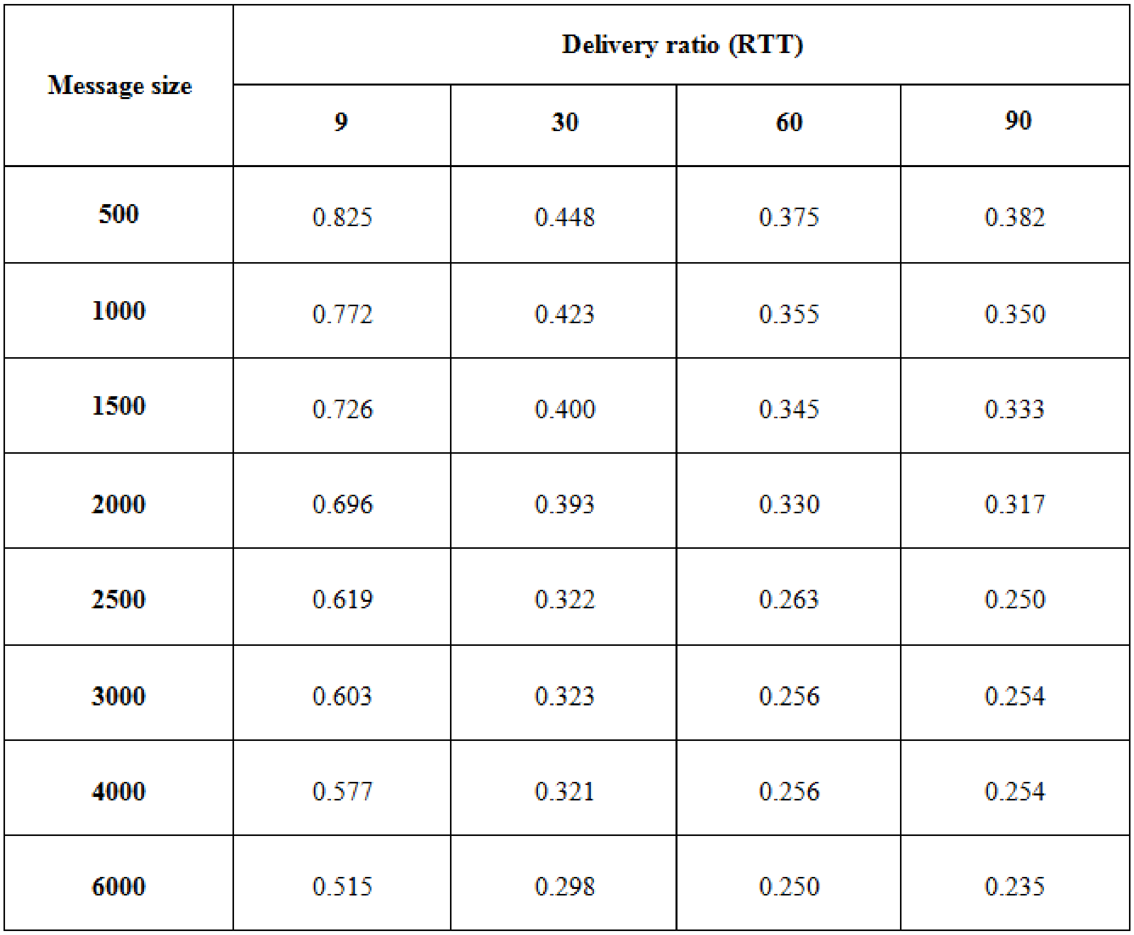


Figure 4.24: Packet delivery ratio for fix nodes

Figure 4.24 shows the measurement of packet delivery ratio for different number of nodes and message size. Packet delivery ratio is decrease when number of nodes in the network increased because packets require more time to reach the destination and the probability of packet loss is increasing. When we change message size from 500 bytes to 6000 bytes, packet delivery ratio is decreasing slowly; due to the amount of packets that arrive to the application layer is decrease.

Table 13: Simulation results of average delivery ratio for mobile nodes using AODV routing protocol.



Message size	Delivery ratio (RTT)			
	9	30	60	90
500	0.825	0.448	0.375	0.382
1000	0.772	0.423	0.355	0.350
1500	0.726	0.400	0.345	0.333
2000	0.696	0.393	0.330	0.317
2500	0.619	0.322	0.263	0.250
3000	0.603	0.323	0.256	0.254
4000	0.577	0.321	0.256	0.254
6000	0.515	0.298	0.250	0.235

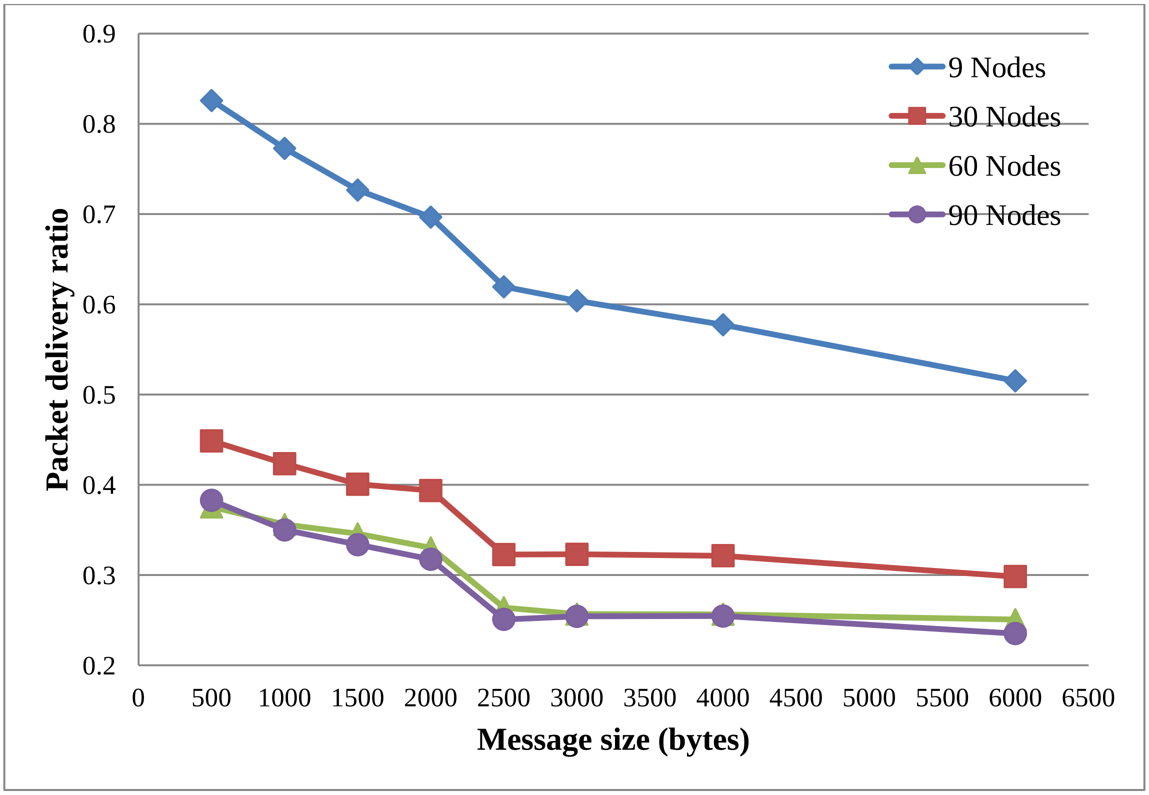


Figure 4.25: Packet delivery ratio for mobile nodes

Figure 4.25, shows the measurement of packet delivery ratio for mobile nodes, which we have different number of mobile nodes and message size. Packet delivery ratio is decrease when number of nodes in the network increased because packets require more time to reach the destination and the probability of packet loss is increasing. When we change message size from 500 bytes to 6000 bytes, packet delivery ratio is decreasing slowly; due to the amount of packets that arrive to the application layer are decrease.

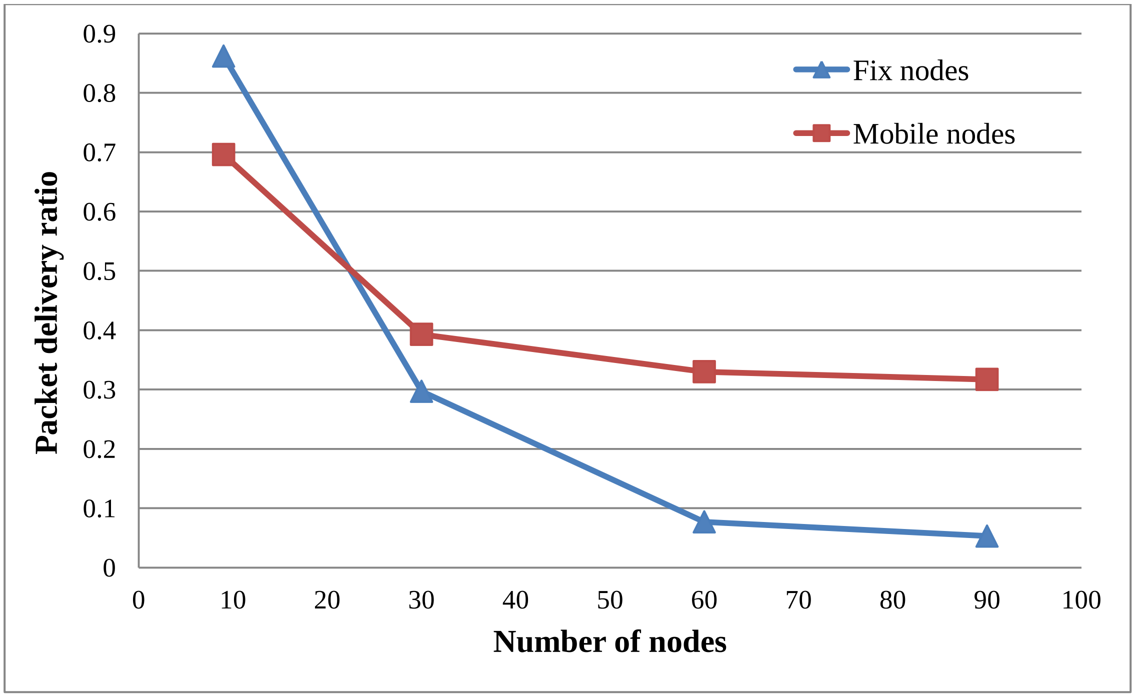


Figure 4.26: Packet delivery ratio for fix and mobile nodes with 2000 bytes message size

Figure 4.26 shows the measurement of packet delivery ratio for fix and mobile nodes for different number of nodes with 2000 bytes message size. Here, packet delivery ratio is decreasing when number of nodes is increasing. The reason is that many packets are lost on the way since they are visiting large number of nodes before reaching to server. In mobile nodes scenario, the packet delivery ratio are higher than fix nodes scenario due to the dynamic topology of the network. So, we can say that when number of nodes 30 or more, packet delivery ratio is better in a dynamic network.

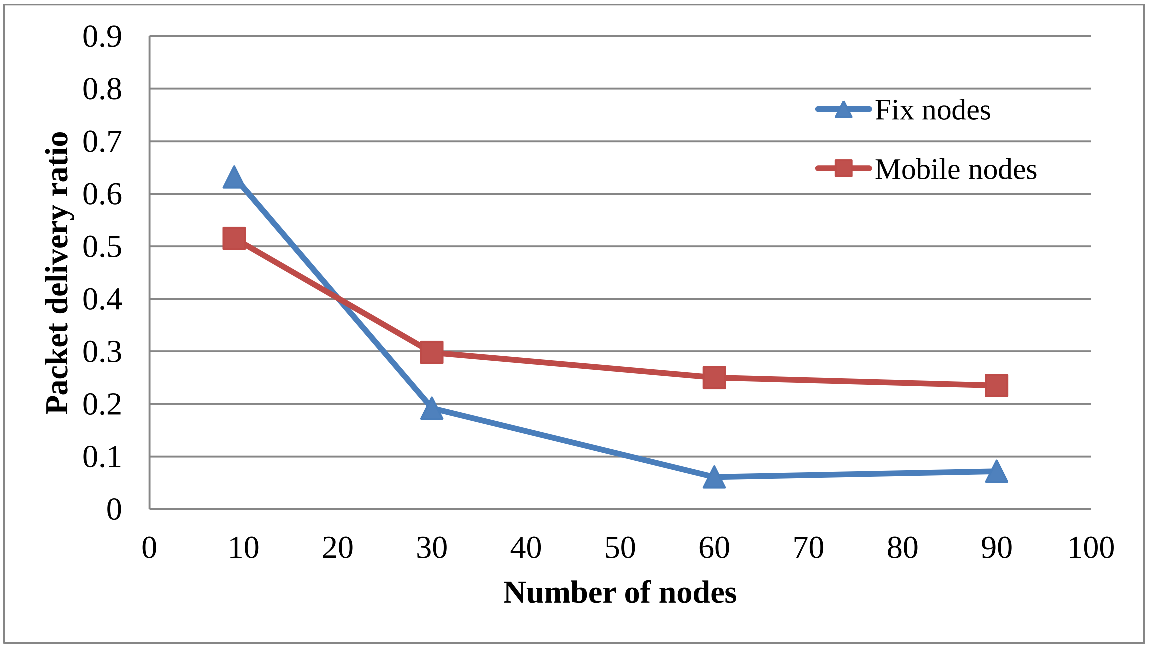


Figure 4.27: Packet delivery ratio for fix and mobile nodes with 6000 bytes message size

Figure 4.27 shows the measurement of packet delivery ratio for fix and mobile nodes for different number of nodes with 6000 bytes message size. Here, packet delivery ratio is decreasing when number of nodes is increasing. The reason is that many packets are lost on the way since they are visiting large number of nodes before reaching to server. In mobile nodes scenario, the packet delivery ratio are higher than fix nodes scenario due to the distance between server and client is changing due to the dynamic topology and more packet are arrived to the destination easily.

4.5 Real Life Experiments

In our real world study, we did a number of experiments with 9 nodes to investigate performance of real life network. We apply only one of the four different numbers of nodes that we done in the simulation, which are the 9 nodes since the other number of nodes are require a considerable resources and participants for applying in the real word. We set out area to the same size that we used in the simulation, which was 500 by 500 meters, where the nodes are distributed randomly in the area. In the real life work, we use pure flooding mechanism for sending the message between the client and server through the intermediate. Also the size of the transmitted message is set to the same message size that we used in our simulation. Where the message size is set to 500 in the beginning, then 1000, 1500, 2000, 2500, 3000, 4000, 6000 bytes, and each message was sent three times, after that we calculated the average of each one. In our real life work, we did two scenarios with the 9 nodes and they are fixed and mobile, the same as our simulation. Where the experimental program of the real life have a specified performance metrics, which we use delivery ratio and average round trip time to evaluate the performance of the real life. First, we start with the fixed nodes and send the message between the source and destination. Then, we repeat the run again with the same message size but we make the nodes mobile. The following tables and graphs show the comparison between the fixed and mobile nodes that we get from real life work.

Table 14: Real life work results for delivery ratio with 9 nodes.

Performance Metric	Message size (bytes)							
	500	1000	1500	2000	2500	3000	4000	6000
Delivery ratio (RTT) for fix nodes	0.852	0.733	0.329	0.298	0.187	0.096	0.045	0.012
Delivery ratio (RTT) for mobile nodes	0.876	0.813	0.272	0.257	0.180	0.125	0.108	0.035

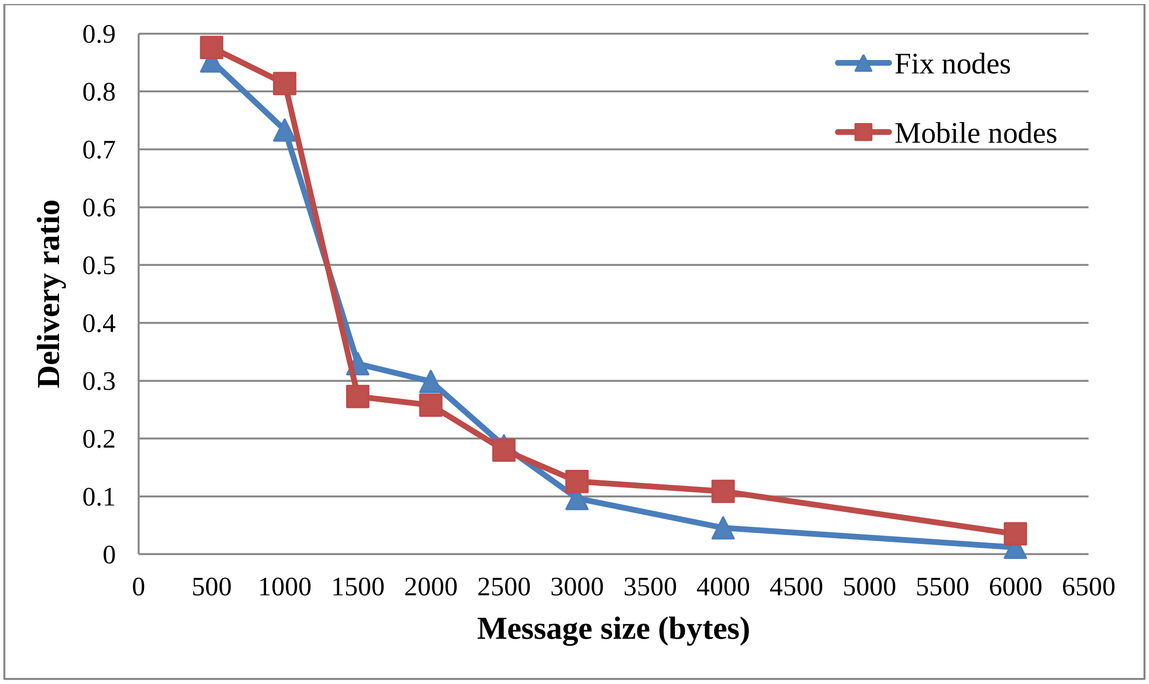


Figure 4.28: Delivery ratio for fix and mobile nodes

Figure 4.28 shows the different values between the fixed and mobile nodes of our real life work by delivery ratio with 9 nodes. Where, both scenarios show the same behavior when increasing the message size, the delivery ratio deceasing due to the increase of packet loss.

Table 15: Real life work results for average round trip time with 9 nodes.

Performance Metric	Message size (bytes)							
	500	1000	1500	2000	2500	3000	4000	6000
Average Round Trip Time (ms) for fix nodes	20.221	40.336	74.485	88.980	130.589	163.52	237.079	379.761
Average Round Trip Time (ms) for mobile nodes	8.171	25.908	77.513	82.284	101.880	119.336	163.691	263.047

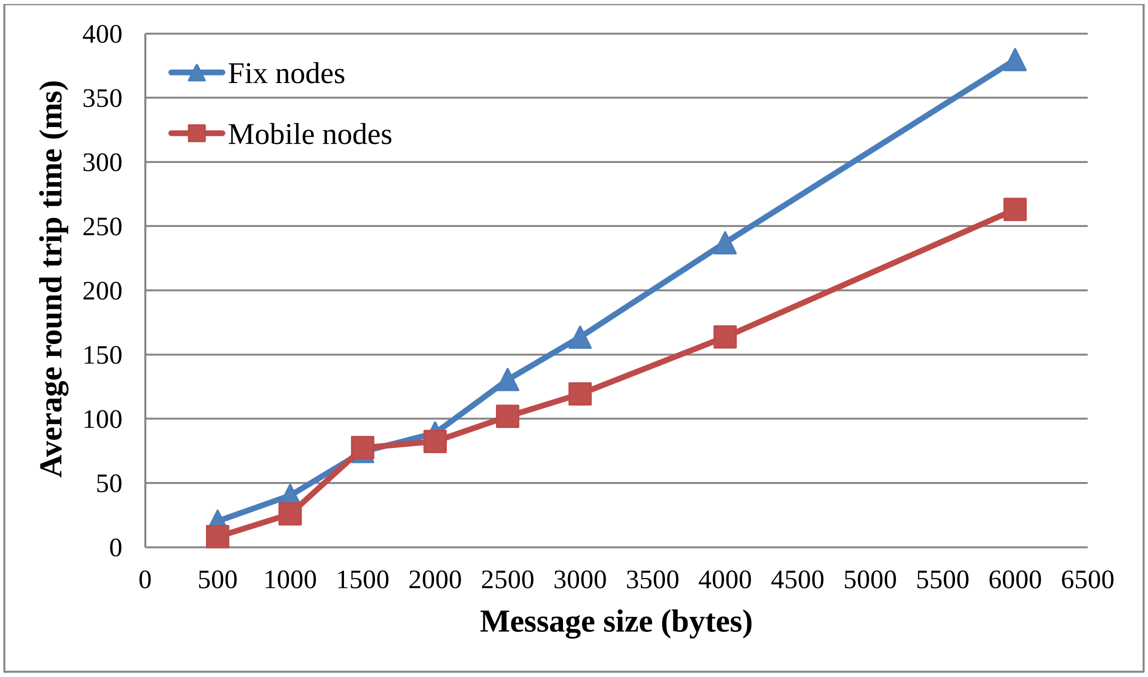


Figure 4.29: Average round trip time (ms) for fix and mobile nodes

Figure 4.29 shows the different values between the fixed and mobile nodes in average round trip time (ms) in our real life work with 9 nodes. We can realize that both of the fixed and mobile scenarios are increase when the message size increases due to its take more time for signal to send and ACK to receive. Also, the mobile scenario has less average round trip time due to the topology changes.

From the previous results of all scenarios for fix and mobile nodes of simulation and real life, we now made a comparison between them for the fix and mobile cases. Which we used the 9 nodes for the comparison. In addition, the performance metric that used to compare between the simulation and real life is delivery ratio because from the OPNET simulator the performance metrics we used is route discovery time, total route request sent, upload response time and packet delivery ratio, but from the real life work, the experimental program have specified performance metrics, which we used the delivery ratio and average round trip time. Which, the delivery ratio is the only common performance metrics between simulation and real life to compare between them.

Table 16: Comparison of simulation and real life work results for delivery ratio with 9 fix nodes.

Performance Metric	**Message size (bytes)**							
	500	**1000**	**1500**	**2000**	**2500**	**3000**	**4000**	**6000**
Delivery ratio for fix nodes for simulation	0.954	0.927	0.893	0.862	0.782	0.774	0.739	0.632
Delivery ratio for fix nodes for real life	0.852	0.733	0.329	0.298	0.187	0.096	0.045	0.012

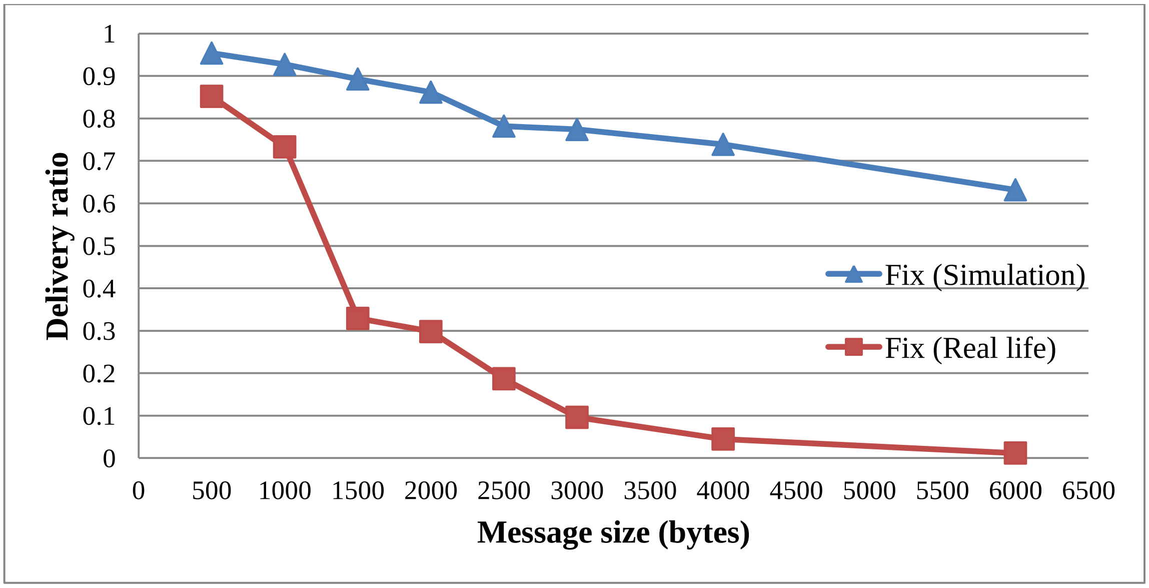


Figure 4.30: Delivery ratio for fix nodes of simulation and real life work

Figure 4.30, shows the different values between of fix nodes for our simulation and real life work by delivery ratio with 9 nodes. Here in simulation, the transmutation signal is not affected by the natural environment like real life, therefore, the amount of received packet in simulation is higher than real life and in both situation, and the delivery ratio is decreasing by increasing the message size due to the amount of packet loss is increasing.

Table 17: Comparison of simulation and real life work results for delivery ratio with 9 mobile nodes.

Performance Metric	Message size (bytes)							
	500	**1000**	**1500**	**2000**	**2500**	**3000**	**4000**	**6000**
Delivery ratio for mobile nodes for simulation	0.825	0.772	0.726	0.696	0.619	0.603	0.577	0.515
Delivery ratio for mobile nodes for real life	0.876	0.813	0.272	0.257	0.18	0.125	0.108	0.035

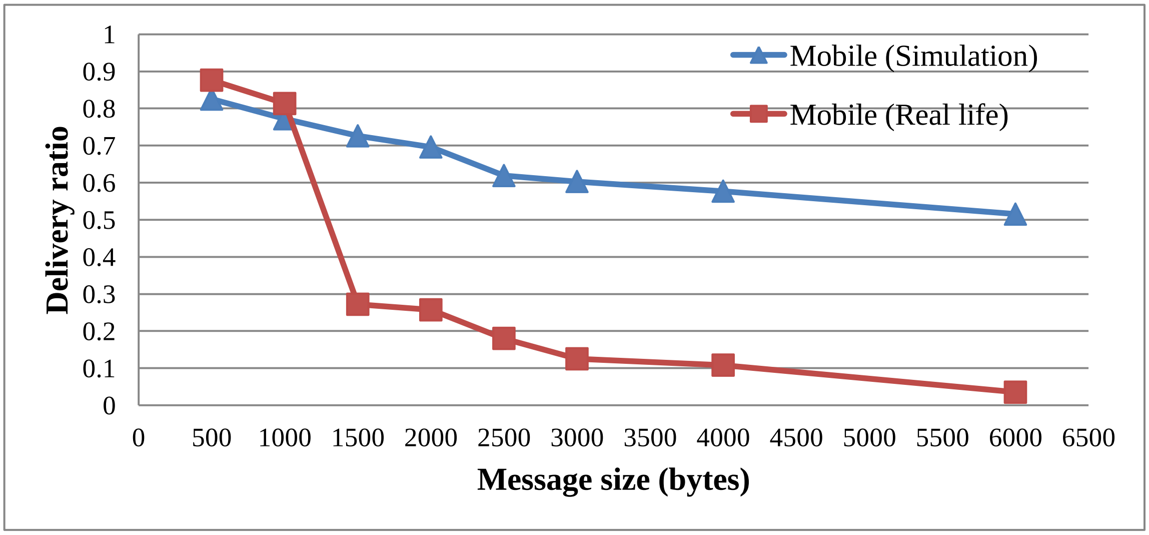


Figure 4.31: Delivery ratio for fix nodes of simulation and real life work

Figure 4.31, shows the different values between of mobile nodes for our simulation and real life work by delivery ratio with 9 nodes. Here in simulation, the natural environment is not affected on the transmutation signal like real life, therefore, the amount of received packet in simulation is higher than real life and in both situation, and the delivery ratio is decreasing by increasing the message size due to the amount of packet loss is increasing.

Chapter 5

CONCLUSION

This work includes three parts, the survey study, the simulation study and experimental work. From the first part, it is concluded that routing protocols play a very important role in the performance of ad hoc networks. Different protocols have different qualities; some of the protocols perform better than others in one metric in using them in a specific scenario and worse in the other and the selection of a suitable protocol definitely increases the performance of the network. The survey study revealed that in mobile ad hoc networks three categories of routing protocols; proactive, reactive and hybrid ones are used.

In this work, performance analysis of Ad-hoc On-demand Distance Vector (AODV) routing protocol is done in ad hoc networks using the OPNET 17.1 simulator. Several simulations are done to investigate the behavior of 802.11b with varying size of FTP traffic using different number of nodes.

Nodes in the network are used as mobile and fixed nodes. Random waypoint mobility model is used as pattern of mobility. Route discovery time, total route request sent, upload response time and delivery ratio are used as performance metrics to evaluated performance of the system.

In our simulations, we made two scenarios; the first one is with the fixed nodes and the second with the mobile nodes. Where, in both cases, the number of nodes is varying from 9, 30, 60 and 90 with message sizes 500, 1000, 1500, 2000, 2500, 3000, 4000 and 6000 bytes and in the mobility scenario, the speed of nodes is set to 1.3m/s. In the real life experiments, 9 nodes are used for mobile and fix cases. Then we compared results of the simulation and real life experiments with common performance metric which is delivery ratio.

From our simulation and real life experimental results, we observed the following. In general, when the number of nodes increases, the route discovery time and total route request sent the different increases for both fix and mobile scenarios but there is a small effect of message sizes on the route discovery time and total route request sent. For large number of nodes, route discovery time and total route request sent are more in dynamic network. Upload response time is also increases for both fix and mobile scenarios when the number of nodes increases. Also, the message sizes have effects on the upload response time, which it increase when message sizes is small and when it became large sizes, begin to decrease for the larger number of nodes. On the other hand, when the number of nodes increases, the packet delivery ratio decreases for both fix and mobile scenarios. In addition, when message sizes increases, the packet delivery ratio is decreases for both cases fix and mobile.

Furthermore, for the future work, more routing protocols or applications could be investigated with different parameters.

REFERENCES

[1] R. Misra. And C.R. Mandal. (2005). Performance comparison of AODV/DSR on-demand routing protocols for ad hoc networks in constrained situation. ICPWC International Conference, IEEE, pp. 86 – 89.

[2] D. Kiwior and L. Lam. (2007). Routing Protocol Performance over Intermittent Links. Military Communications Conference, MILCOM, IEEE, pp. 1 – 8.

[3] S. Demers and L. Kant. (2006). MANETs: Performance Analysis and Management. Military Communications Conference, MILCOM, pp.1 – 7.

[4] P. Jacquet., Muhlethaler P., Clausen T., Laouiti A., Qayyum A., & Viennot L. (2001). Optimized link state routing protocol for ad hoc networks. In Proceedings of the 5th IEEE Multi Topic Conference (INMIC 2001).

[5] J. Haerri, Fethi Filali, Christian Bonnet. (2006). Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns. Institute Eurecomz Department of Mobile Communications. Publisher: Citeseer, Pages: 14- -17.

[6] C. Adjih, M. Pascale, M. Paul, B. Emmanuel and P. Thierry. (2008). QoS Support, Security and OSPF Interconnection in a MANET using OLSR. Ad Hoc Networks Journal.

[7] A. Vallejo, G. Corral, J. Abella and A. Zaballos. (2006). Ad hoc Routing Performance Study Using OPNET Modeler. University Ramon Llull, Barcelona ,Spain.

[8] A. Zahary, A. Ayesh. (2007). Analytical study to detect threshold number of efficient routes in multipath AODV extensions. proceedings of International Conference of Computer Engineering & Systems, ICCES, pp. 95 – 100.

[9] C. Perkins, B.-R. E. and D. S. Ad hoc On-demand Distance Vector routing. Request For Comments (Proposed Standard) 3561, Internet Engineering Task.

[10] C. Perkins, E.M. Royer, S.R. Das. (2001) Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. IEEE Personal Communications Magazine, no. 8, pp. 16-28.

[11] Prof. Dr. R. Nat. habil. C. Görg. (2005). Performance Analysis of Ad hoc On-demand Distance Vector routing (AODV) using OPNET Simulator. University of Bremen, 11th April. PP. 15.

[12] A. S.Tanenbaum. (1998). Computer Networks. Prentice Hall India (PHI), November 1998.

[13] P. Brenner. A Technical Tutorial on IEEE 802.11 Protocols. http://www.sss mag.com/pdf/802_11tut.pdf.

[14] Prof. Dr. R. Nat. habil. C. Görg. (2005). Performance Analysis of Ad hoc On-demand Distance Vector routing (AODV) using OPNET Simulator. University of Bremen, 11th April 2005. PP. 13.

[15] M. Kropff, Krop T., Hollick M., Mogre P. S. and Steinmetz R. (2006). A Survey on Real World and Emulation Testbeds for Mobile Ad hoc Networks. Available at ftp://ftp.kom.tu-darmstadt.de/pub/papers/testbedsurvey.pdf.

[16] A. Kostin. (2008). Concept of Systems and System Simulation, in Advanced System Simulation. Computer Engineering Department, Eastern Mediterranean University.

[17] J. Banks, J. S. Carson and B. L. Nelson. (1996). Discrete-Event System Simulation,. 2nd ed., Prentice-Hall International.

[18] A. Kostin,. (2008). Discrete Event Simulation, in Advanced System Simulation. Computer Engineering Department, Eastern Mediterranean University.

[19] IEEE, “Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specification,” IEEE 802.11 Draft Version 4.0, May 1996.

[20] S. Chrysoula. (2012). Performance Comparison of Manet Routing Protocols based on real-life scenarios. Master in Information System (MIS). University of Macedonia. PP. 54 – 56.

[21] S. Barakovic., A. Kasapovic. And J. Barakovic. (2010). Comparison of MANET Routing Protocols in Diffrenet Traffic and Mobility Models. Telfor journal, Vol. 2, No. 1.

[22] L. Guo,. Y. Peng , X. Wang , D. Jiang. And Y. Yu. (2011). Performance Evaluation for On-demand Routing Protocols Based on OPNET Modules in Wireless Mesh Networks. Computers and Electrical Engineering 37. PP.106–114.

[23] J. Haerri., F. Filali and C. Bonnet. Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns. Department of mobile communication.

[24] J. Khan,. Dr.S. I. Hydar,. Dr.S. M. Fakar,. and D. Mustafa. (March 2011). Modeling and Simulation of Dynamic Intermediate Nodes and Performance Analysis in MANETS Reactive Routing protocols. School of Information & communication Technology, Asia e University, No. 4, Jalan Sultan Sulaiman, 50000 Kuala Lumpur, Malaysia. Graduate School of science & Engineering, PAF-KIET, PAF Base Korangi Creek karachi75190 pakistan, International Journal of Grid and Distributed Computing, Vol. 4, No. 1.

[25] V. Ayatollahi Tafti,. And A. Gandomi. (2010). Performance of QoS Parameters in MANET Application Traffics in Large Scale Scenarios. World Academy of Science, Engineering and Technology 72.

[26] N. Arora,. And S. Arora. Performance Comparisons of Routing Protocols and TCP in MANETS. Institute of Technology and Management, ECE Dept, Maharishi Dayanand University Gurgaon (Haryana), India.

[27] V. Talooki,. And K. Ziarati. Performance Comparison of Routing Protocols for Mobile Ad-Hoc Networks. Asia-Pacific conf. on Comm., APCC‟06.

[28] J. Webb. (2005). Analysis of Packet Flows in Simulated Ad-hoc networks Using Standard Network Tools. Master's thesis, University of California Santa Cruz, USA.

[29] M. Mohammad Siddique, and Andreas Konsgen. (2007). WLAN Lab Opnet Tutorial", University Bermen Press.

[30] Http://www.opnet.com. (2007). How to Design Mobile Ad hoc Networks and Protocols. January 23, 2007.

[31] OPNET, [Online]. Available: http://www.opnet.com/.

[32] T. Camp,. J. Boleng and V. Davies. (2002). A Survey of Mobility Models for Ad Hoc Network Research. Dept. of Math. and Computer Science. Colorado School of Mines, Golden, CO.

[33] B. Vanrajkumar Dineshkumar. (2012). Improvement Of Aodv Routing Protocol Based On Wireless Networks. Department Of Computer Engineering. Government Engineering College, Modasa.

[34] M. Bouhorma., H. Bentaouit and A. Boudhir. (2009). Performance Comparison of Ad-hoc Routing Protocols AODV and DSR. Departement Genie Informatique, ERIT. Faculte des Sciences et techniques de Tanger. Tangiers, Morocco.

APPENDICES

Appendix A: Formulas used in calculation of confidence interval

- Confidence Interval (CI) = $\bar{x} \pm t^* \frac{s}{\sqrt{n}}$

  $\bar{x} =$ mean

  S = standard deviation

  N = sample size

  t^* = critical value

- Standard deviation formula:

  S = $\sqrt{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \ \dots\dots\dots\dots\dots\dots\dots\dots\dots \ +(x_n - \bar{x})^2}$

- Mean calculation:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ \dots\dots\dots\dots\dots \ + x_n}{n}$$

- t^* is calculated using the formula TINV (1-level, n-1)

  N = Sample size

  Level = Confidence interval or confidence level

  n-1 = The degree of freedom

Appendix B: Average values and Confidence Intervals of the Investigated Performance Metrics

In this appendix, average values and confidence intervals of the investigated performance metrics of the experiments are provided. The performance metrics that were used in the experiments are route discovery time, total route request for AODV routing protocol and upload response time and packet delivery ratio for FTP application. The message sizes that been measured to find the confidence intervals are 500, 2000 and 6000 for both fix and mobile nodes.

Average values and 95% confidence intervals of the performance metrics for AODV with message size 500 bytes for fixed nodes.

Metric	**Number of Nodes**			
	9	**30**	**60**	**90**
Route discovery time	1.13 ± 0.12	3.51 ± 0.34	4.34 ± 0.17	4.23 ± 0.33
Total route request	18.52 ± 1.64	143.19 ± 3.33	197.26 ± 6.99	238.04 ± 5.93
Upload response time	11.89 ± 1.15	74.88 ± 10.16	79.95 ± 8.40	82.95 ± 4.69
Packet delivery ratio	0.95 ± 0.02	0.43 ± 0.004	0.34 ± 0.01	0.27 ± 0.008

Average values and 95% confidence intervals of the performance metrics for AODV with message size 2000 bytes for fixed nodes.

Metric	Number of Nodes			
	9	30	60	90
Route discovery time	1.38 ± 0.21	3.49 ± 0.24	3.91 ± 0.23	4.32 ± 0.37
Total route request	16.25 ± 1.12	139.02 ± 7.38	198.35 ± 3.62	237.66 ± 5.75
Upload response time	19.80 ± 3.21	87.01 ± 10.19	80.85 ± 9.42	85.51 ± 12.50
Packet delivery ratio	0.85 ± 0.01	0.16 ± 0.003	0.19 ± 0.01	0.14 ± 0.01

Average values and 95% confidence intervals of the performance metrics for AODV with message size 6000 bytes for fixed nodes.

Metric	Number of Nodes			
	9	30	60	90
Route discovery time	1.67 ± 0.12	3.64 ± 0.31	4.19 ± 0.32	4.24 ± 0.25
Total route request	16.67 ± 1.73	141.83 ± 3.35	195.19 ± 1.46	239.19 ± 7.54
Upload response time	27.04 ± 3.20	97.13 ± 13.87	72.67 ± 19.12	42.98 ± 25.03
Packet delivery ratio	0.62 ± 0.03	0.19 ± 0.004	0.12 ± 0.01	0.10 ± 0.006

Average values and 95% confidence intervals of the performance metrics for AODV with message size 500 bytes for mobile nodes.

Metric	**Number of Nodes**			
	9	**30**	**60**	**90**
Route discovery time	0.93 ± 0.08	3.37 ± 0.18	5.36 ± 0.92	5.66 ± 0.73
Total route request	43.97 ± 7.74	173.83 ± 4.47	301.862 ± 23.66	426.64 ± 4.98
Upload response time	29.09 ± 1.94	47.47 ± 15.10	80.06 ± 32.88	56.92 ± 23.85
Packet delivery ratio	0.82 ± 0.01	0.44 ± 0.012	0.37 ± 0.10	0.38 ± 0.05

Average values and 95% confidence intervals of the performance metrics for AODV with message size 2000 bytes for mobile nodes.

Metric	**Number of Nodes**			
	9	**30**	**60**	**90**
Route discovery time	1.05 ± 0.11	3.44 ± 0.18	5.11 ± 0.76	5.79 ± 0.32
Total route request	41.47 ± 8.29	174.17 ± 5.41	293.08 ± 10.26	430.96 ± 13.59
Upload response time	32.12 ± 7.68	64.79 ± 10.86	73.25 ± 21.09	75.28 ± 10.52
Packet delivery ratio	0.69 ± 0.01	0.39 ± 0.009	0.32 ± 0.11	0.31 ± 0.06

Average values and 95% confidence intervals of the performance metrics for AODV with message size 6000 bytes for mobile nodes.

Metric	**Number of Nodes**			
	9	**30**	**60**	**90**
Route discovery time	1.35 ± 0.05	3.49 ± 0.13	5.17 ± 0.59	5.65 ± 0.39
Total route request	48.75 ± 2.48	174.23 ± 3.36	293.29 ± 24.25	430.19 ± 12.16
Upload response time	46.53 ± 5.49	85.09 ± 11.65	80.08 ± 8.48	93.85 ± 20.20
Packet delivery ratio	0.51 ± 0.01	0.29 ± 0.01	0.24 ± 0.09	0.24 ± 0.07

Table 1: Summary of related work

Ref No	Simulator	Routing Protocol	No. of nodes	Application used	Performance metric	Simulation area (m x m)	Node speed (m/s)
[20]	NS2	AODV DSDV OLSR	10 to 100	Files, test messages	Packet delivery ratio Throughput Average delay	500 x 500	2
[21]	NS2	AODV DSR DSDV	10, 20 and 30 50	CBR traffic	Packet Delivery Ratio Average End to End delay Normalized Routing Load	500 x 500	0 - 20
[22]	OPNET	AODV DSR	20, 40	FTP	Routing discovery time Avg. number of hops Network delay Network throughput	4000 x 4000	2 - 6
[23]	NS2	AODV OLSR	40, 50, 60, 70 And 80	CBR traffic	Packet delivery ratio Routing overhead End to end delay Number of hops	1000 x 1000	uniform