

**Performance Evaluation of Routing Protocols in
Wireless Mobile Ad Hoc Networks (MANETS) using
OPNET Simulator**

Mohammadamin Roshanasan

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

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2012
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. Doğu Arifler

2. Assoc. Prof. Dr. Muhammed Salamah

3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

Mobile ad hoc networks (MANETs) have already opened a new point of view in the field of wireless networks which includes hundreds and thousands of nodes. The wireless nodes are communicating without the need of any kind of neither infrastructure like the base stations or routers, nor centralized administration. Wireless nodes are free of moving anytime, anywhere. Therefore, mobile ad hoc networks need to have dynamic routing protocols. Mobile Ad hoc network routing protocols are divided into several different categories such as Proactive, Reactive and Hybrid Routing Protocols. Also there are a lot of performance metrics to compare the routing protocols. Each of them has its own attributes and well for specific area, such as: throughput, jitter, packet delivery ratio, average number of hops, route discovery time and end-to-end delay, which are some important ones.

In this thesis three well known routing protocols; Optimized Link State Routing (OLSR), Ad-hoc On-demand Distance Vector (AODV) and Temporary Ordered Routing Algorithm (TORA) were evaluated using the OPNET simulator under the medium load traffic size in FTP protocol. The first one (OLSR) is a proactive protocol depending on routing tables which are maintained at each node. The second one (AODV) is a reactive protocol, that finds a route to a destination on-demand. And the third ones' TORA which works in both categories as reactive and proactive. The random waypoint mobility model is used as pattern of mobility. As performance metrics average throughput,

average network load and average end-to-end delay are examined in different number of nodes, file sizes and node speeds.

The result from the simulations of this study reveals that different protocols have different qualities; some of the protocols perform better than others in one metric when using them in a specific scenario and worse in other metrics. After analyzing performances of some well-known reactive and proactive routing protocols, in case of average throughput, average end-to-end delay and average network load, the superiority of proactive protocols, over reactive ones is observed in different network scenarios. From the simulation results it is observed that the average end-to-end delay increases slightly when the number of nodes increases in OLSR. Also average throughput shown in OLSR was the highest comparing to AODV and TORA. Among the reactive protocols, AODV performs better than TORA when file sizes, speed of nodes and number of nodes are changed. On the other hand, TORA gives a highest end-to-end delay and lowest throughput compared to AODV and OLSR.

Keywords: Mobile wireless ad hoc networks, simulation, routing protocols, performance evaluation, OPNET simulator.

ÖZ

Gezgin özel amaca yönelik ağlar (MANETs), kablosuz ağlar alanında yeni bir oluşum olup yüzlerce veya binlerce düğümün herhangi bir altyapı veya kontrol merkezi olmaksızın haberleşebilme imkanını sağlamaktadır. Kablosuz düğümlerin (dizüstü bilgisayarlar, kişisel digital yardımcılar ve gezgin telefonlar) özel amaca yönelik senaryolarda hareketleri serbesttir. Buna bağlı olarak, bu tip ağlarda dinamik olarak değişebilen yönlendirme protokollerine gereksinim vardır. Gezgin özel amaca yönelik ağlarda kullanılan yönlendirme protokolleri önceden etkin (proactive), tepkin (reactive) ve karma (hybrid) olarak sınıflandırılabilirler. Yönlendirme protokollerinin performanslarını ölçmek ve karşılaştırmak için kullanılan birçok performans ölçü birimleri vardır. Her birinin kendine özgü özellikleri ve iyi olduğu kullanım alanları vardır. Bazı bilinen ölçü birimleri, çıkış is oranı (throughput), seğirme (jitter), paket dağıtım oranı (packet delivery ratio), ortalama sekme sayısı (average number of hops), yön bulma zamanı (route discovery time), ve bir yönden bir yöne gecikmedir (end-to-end delay)

Bu tezin bir amacı özel amaca yönelik ağlarda kullanılan ve var olan protokolleri incelemek ve anlamaktır. Diğer bir amacı da OPNET simulatörü kullanarak iyi bilinen OLSR, AODV ve TORA protokollerinin performansını özel amaca yönelik ağlarda orta hızdaki dosya transfer (FTP) protokolünün performansını incelemektedir. OLSR protokolü önceden etkin protokoller sınıfında olup yönlendirme tabloları her sekme üzerinde yapılandırılmaktadır. AODV tepkin protokoller sınıfında olup alıcıya olan rota

talep üzerine bulunmaktadır. TORA protokolü her iki kategoriye göre çalışabilmektedir. Bu tezde tepkin protokolü olarak kullanılmıştır. Bu çalışmada rastgele ara nokta hareketlilik modeli hareketliliği sağlamak için kullanılmıştır. Performans ölçme birimi olarak, ortalama çıkan iş oranı (average throughput), ortalama ağ yükü (average network load) ve ortalama bir uçtan bir uca gecikme (average end-to end delay) farklı boyutlardaki veri, farklı sekme hızları ve farklı sekme sayıları kullanılarak incelenmiştir.

Simulasyon sonuçları seçilen protokollerin farklılıklarını göstermiştir. Protokoller aynı senaryolarda kullanılan ölçü birimlerinde farklı sonuçlar üretmiştir. Genel olarak seçilen ölçü birimlerinde önceden etkin protokoller tepkin protokollerden daha iyi sonuç vermiştir. Simulasyon sonuçlarına göre OLSR protokolü kullanırken sekme sayısını artırdığımız zaman ortalama bir uçtan bir uca gecikme az miktarda yükselmiştir. Buna ek olarak OLSR protokolünde ortalama çıkan iş oranı AODV ve TORA protokollerinden daha fazla çıkmıştır. Dosya boyutu, sekme hızı ve sekme sayısı artırıldığı zaman, etkin protokollerden olan AODV'nin performansı TORA dan daha iyi çıkmıştır. Aynı zamanda TORA, AODV ve OLSR ile karşılaştırıldığında en yüksek bir uçtan bir uca gecikme ve en düşük çıkan ortalama çıkan iş oranı değerleri vermiştir.

Anahtar Kelimeler: Gezgin kablosuz özel amaca yönelik ağlar, simulasyon, yönlendirme protokolleri, performans ölçme / değerlendirme, OPNET simulasyon programı.

TABLE OF CONTENTS

ÖZ	v
LIST OF TABLES	v
LIST OF FIGURES	vii
LIST OF ABBREVIATION	xi
1 INTRODUCTION	1
2 DESCRIPTION OF THE SELECTED ROUTING PROTOCOLS	6
2.1 Optimized Link State Routing (OLSR)	6
2.1.1 Components of OLSR.....	8
2.2 Temporally Ordered Routing Algorithm (TORA).....	9
2.3 Ad hoc On-demand Distance Vector Routing (AODV).....	12
2.3.1 Path Discovery and Path Setup	13
2.3.2 Routing Table Management.....	18
2.4 Comparison of Selected Routing Protocols	22
2.5 Review of the State of the Art.....	23
3 OPNET SIMULATION ENVIRONMENT	25
3.1 OPNET Architecture.....	26
3.2 Architecture of MANET Models in OPNET	27
3.3 Configuring Routing Protocols in OPNET	30
3.4 Taking Results of Simulation	32
4 MODELING OF MANETs IN OPNET, SIMULATION SETUP AND RESULTS ...	33
4.1 Performance Metrics	33

4.2 Modelling of MANETs in OPNET and Simulation Setup	34
4.3 Simulation With Different Ad hoc Network Scenarios and Results.....	45
4.3.1 Investigation of Different Number of Nodes	45
4.3.2 Investigation of Different File Sizes	53
4.3.3 Investigation of Different Node Speeds.....	67
4.4 Simulation Results and Discussions	76
5 CONCLUSION	79
REFERENCES	81
APPENDICES	88
Appendix A: AODV Source Code.....	89
Appendix B: Step by Step Configuration of Simulation	99

LIST OF TABLES

Table 1. Differences between three MANET routing protocols	22
Table 2: Comparison with other works	24
Table 3. General attributes for scenario 1	46
Table 4. Mobility attributes for scenario 1	46
Table 5. Application configuration attributes for scenario 1	46
Table 6. Profile configuration attributes for scenario 1	46
Table 7. Simulation results of average end-to-end delay in msec with file size 512 bytes and maximum node speed 5 m/s	47
Table 8. Simulation results of average end-to-end delay in msec with file size 512 bytes and maximum node speed 5 m/s with TORA protocol.....	48
Table 9. Simulation results of average network load in Kbits/sec with file size 512 bytes and maximum node speed 5 m/s	49
Table 10. Simulation results of average throughput in Kbits/s with file size 512 bytes and maximum node speed 5 m/s	50
Table 11. General attribute for scenario 2.....	53
Table 12. Simulation results of average end-to-end delay in msec with 100 nodes and maximum node speed 5 m/s.....	53
Table 13. Simulation results of average end-to-end delay in msec with 40 nodes and maximum node speed 5 m/s.....	54
Table 14. Simulation results of average network load in Kbits/s with 40 nodes and maximum node speed 5 m/s.....	55

Table 15. Simulation results of average network load in Kbits/s with 100 nodes and maximum node speed 5 m/s.....	56
Table 16. Simulation results of average throughput in Kbits/s with 40 nodes and maximum node speed 5 m/s.....	57
Table 17. Simulation results of average throughput in Kbits/s with 100 nodes and maximum node speed 5 m/s.....	58
Table 18. Simulation results of average end-to-end delay in msec with 40 and 100 nodes and maximum 5 m/s node speed.....	61
Table 19. Simulation results of average network load in Kbits/s with 40 and 100 nodes and maximum 5 m/s node speed.....	63
Table 20. Simulation results of average throughput in Kbits/s with 40 and 100 nodes with maximum 5 m/s node speed.....	65
Table 21. AODV performance results for 100 nodes with different speeds and file sizes.....	67
Table 22. OLSR performance results for 100 nodes with different speeds and file sizes	70
Table 23. TORA performance results for 100 nodes with different speeds and file sizes.....	73

LIST OF FIGURES

Figure 1. Infrastructure based wireless network	1
Figure 2. Ad hoc network structure	2
Figure 3. Overview of ad hoc routing protocols	3
Figure 4. Multipoint relays of the OLSR network system.....	6
Figure 5. Route discovery for QRY message	11
Figure 6. Route discoveries in TORA – update message	12
Figure 7. Reverse paths.....	15
Figure 8. Forward paths	16
Figure 9. Simulation process for OPNET	26
Figure 10. MANET model architecture	27
Figure 11. MANET object palette	28
Figure 12. Routing protocol configuration in OPNET	31
Figure 13. Choosing statistics	32
Figure 14. Review of startup wizard.....	35
Figure 15. Application configuration attribute	36
Figure 16. DES Execution Manager	38
Figure 17. Profile configuration attribute	39
Figure 18. Mobility configuration attributes.....	41
Figure 19. Wireless LAN Workstation attribute.....	43
Figure 20. Deploy application setup	44

Figure 21. Average end-to-end delay versus number of nodes with file size 512 bytes and maximum node speed 5 m/s	47
Figure 22. Average end-to-end delay versus number of nodes with file size 512 bytes and maximum node speed 5 m/s with TORA protocol.....	48
Figure 23. Average network load versus number of nodes with file size 512 bytes and maximum node speed 5 m/s.....	49
Figure 24. Average throughput versus number of nodes with file size 512 bytes and maximum node speed 5 m/s.....	50
Figure 25. Average end-to-end delay versus different file size with 100 nodes and maximum node speed 5 m/s.....	53
Figure 26. Average end-to-end delay versus different file size with 40 nodes and maximum node speed 5 m/s.....	54
Figure 27. Average network load versus different file size with 40 nodes and maximum node speed 5 m/s.....	55
Figure 28. Average network load versus different file size with 100 nodes and maximum node speed 5 m/s.....	56
Figure 29. Average throughput versus different file size with 40 nodes and maximum node speed 5 m/s.....	57
Figure 30. Average throughput versus different file size with 100 nodes and maximum node speed 5 m/s.....	58
Figure 31. Average end-to-end delay versus file size for AODV with maximum 5 m/s node speed.....	61
Figure 32. Average end-to-end delay versus file size for OLSR with maximum 5 m/s node speed.....	62

Figure 33. Average network load versus file size for AODV with maximum 5 m/s node speed	63
Figure 34. Average network load versus file size for OLSR with maximum 5 m/s node speed	64
Figure 35. Average network load versus file size for TORA with maximum 5 m/s node speed	64
Figure 36. Average throughput versus file size for AODV with maximum 5 m/s node speed	65
Figure 37. Average throughput versus file size for OLSR with maximum 5 m/s node speed	66
Figure 38. Average throughput versus file size for TORA with maximum 5 m/s node speed	66
Figure 39. Average end-to-end delay versus file size with 100 nodes for AODV protocol with different node speeds	67
Figure 40. Average network load versus file size with 100 nodes for AODV protocol with different node speeds	68
Figure 41. Average throughput versus file size with 100 nodes for AODV protocol with different node speeds	68
Figure 42. Average end-to-end delay versus file size with 100 nodes for OLSR protocol with different node speeds	70
Figure 43. Average network load versus file size with 100 nodes for OLSR protocol with different node speeds	71
Figure 44. Average throughput versus file size with 100 nodes for OLSR protocol with different node speeds	71

Figure 45. Average end-to-end delay versus file size with 100 nodes for TORA protocol with different node speeds73

Figure 46. Average network load versus file size with 100 nodes for TORA protocol with different node speeds74

Figure 47. Average throughput versus file size with 100 nodes for TORA protocol with different node speeds74

LIST OF ABBREVIATION

AODV	Ad hoc On demand Distance Vector Routing
DAG	Direct Acyclic Graph
DES	Discrete Event Simulation
MANETs	Mobile Ad hoc Networks
MID	Multiple Interface Declaration
MPR	Multipoint Relays
NS 2/3	Network Simulation 2/3
OLSR	Optimized Link State Routing
OPNET	Optimized Network Engineering Tool
RERR	Route Error Message
RREP	Route Replay Packet
RREQ	Route Request Packet
TC	Topology Control
TORA	Temporally Ordered Routing Algorithm
TTL	Time To Life
WMN	Wireless Mesh Network
WSN	Wireless Sensors Network

Chapter 1

INTRODUCTION

During this decade, wireless networks have become very famous in the area of communication. Considering this, wireless networks are also being used in all places such as military application, industrial application and even in personal networks (laptop, mobile phone, MP3 player, personal digital assistance and personal computer) as illustrated in Figure 1. These nodes can be located in cars, ships, airplanes or with people having small electronic devices [1].

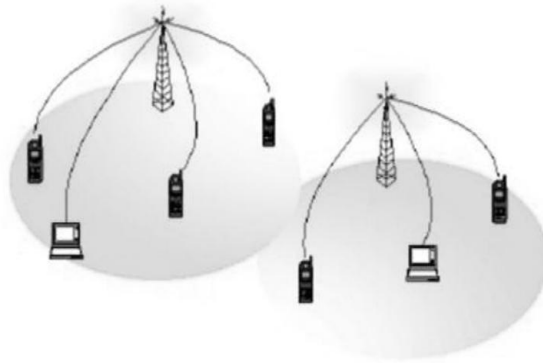


Figure 1. Infrastructure based wireless network [2]

Over the recent years, the difference between wireless and wired networks has been in the communication channel since there is physical medium in wire communication but on the other side physical medium does not exist.

Wireless networks in this decade became popular in different programs and applications as mentioned because of following factors: reliability of application, cost of program, the state of being easy for installation, bandwidth, total amount of needed power, performance and the safety of network [3].

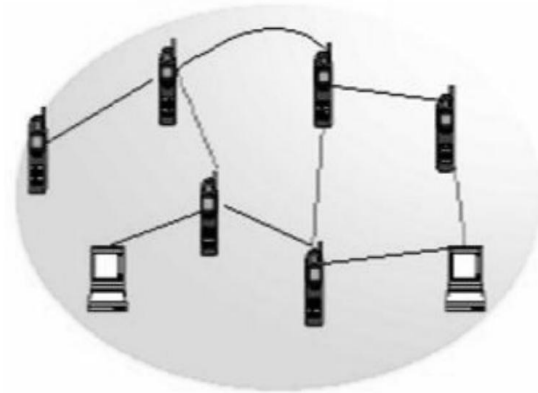


Figure 2. Ad hoc network structure [2]

MANETs (Mobile Ad hoc Networks) [4] [5] which can be observed as wireless networks, as shown in Figure 2, work without the need of any kind of neither infrastructure nor centralized administration. To cover a large area and also the topology change dynamically and uncertainly, MANET does not have fixed topologies. In the traditional routing protocols used for internet, wireless networks can not be delivered to directly end-to-end; as a matter of fact some basic communications are not valid in all situations for some dynamical changing in networks and may not be correct for mobile nodes.

Ad hoc networks act on a single-hop or multi-hop basis where wireless nodes are able to operate as routers in the intermediate stage for transfers of other members of the network.

Proactive, reactive, hierarchical, geographical, power aware, multicast, geographical multicasting, security and others are ad hoc networks classified. However, the main categories are the first three ones as shown in Figure 3. These categories are based on applications which ad hoc network used. Also, there is another category for ad hoc networks base in the area that it is running, i.e. the Mobile Ad hoc Networks or as stance form called MANETs, Wireless Mesh Networks (WMNs), Network of Wireless Sensors.

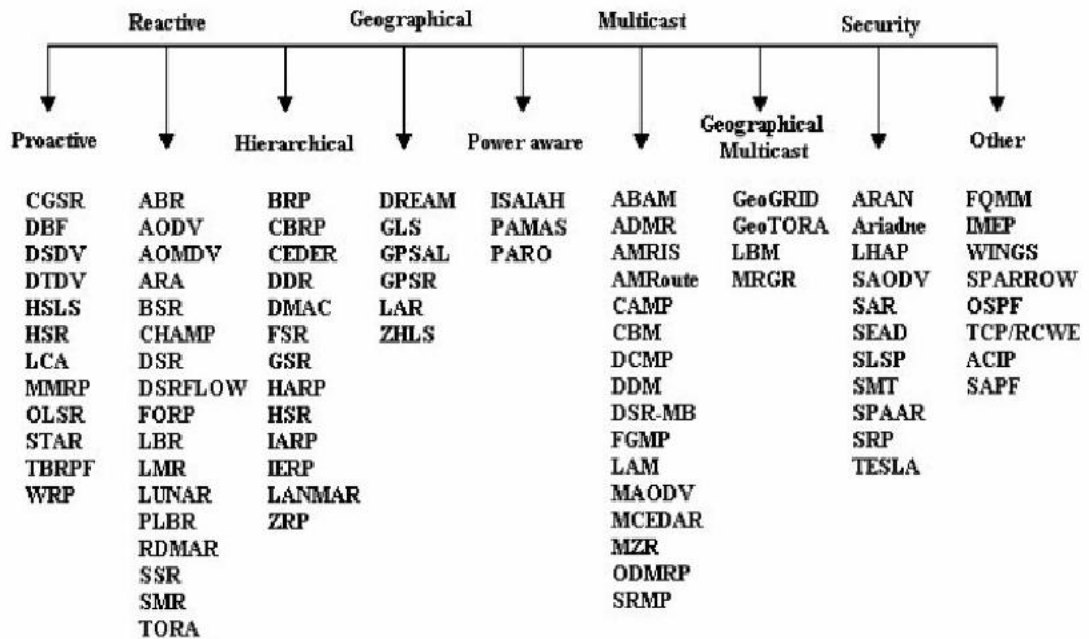


Figure 3. Overview of ad hoc routing protocols [6]

The main categories are called by other names as proactive (on-demand), reactive (table driven) and hierarchical (hybrid). In table driven approach; each router is able to contain one or more routing table together. Routing tables are absent when it needs on-demand routing protocols. In the on demand, route request starts to establish a route when it needs the route.

Table driven routing protocols are much faster and more efficient than other routing protocols like on-demand. It is difficult to maintain a complete routing table in a dynamic network i.e. MANET. However, on-demand protocols are effective by considering the bandwidth, power etc. [7].

The hybrid routing protocol is working in both divisions as proactive and reactive. As described, proactive and reactive protocols are designed to decrease the route discovery overheads and rise the scalability by letting nodes with close proximity work together to form some sort of a backbone. This is highly achieved by proactively maintaining routes to nearby nodes and finding routes to far away nodes which are using a route discovery approach. The most hybrid protocols proposed to date are zone-based, which means the network is separated or observed as a number of zones by each node. Other groups' nodes enter into some of the trees or clusters.

In this current thesis, the focus is on OLSR, AODV and TORA in MANETs which they are in different classifications of MANET protocols and available in the simulation program (OPNET) to realize the importance of routing protocols using the OPNET simulator.

One of the goals of this MS thesis is to examine existing models, algorithms and schemes that are used in MANETs. Another goal is to use the OPNET simulator to evaluate performance of ad hoc networks with well-known protocols OLSR, AODV and TORA to show their performances in ad hoc networks.

The thesis is organized as follows: Chapter 1 provides a general introduction; Chapter 2 introduces the routing protocols in MANETs which were selected for investigation. Chapter 3 describes the simulation program OPNET. In Chapter 4, modelling 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

DESCRIPTION OF THE SELECTED ROUTING PROTOCOLS

2.1 Optimized Link State Routing (OLSR)

OLSR is a proactive (table-driven) routing protocol i.e. frequently exchanges topology information with other nodes of the network [8]. This protocol is optimization of traditional link state protocol developed for mobile Ad hoc network and is also used in WiMAX Mesh. Minimizing the required number of control packets transmission makes control packets size short which are the OLSR accountabilities. The main goal of OLSR is to organize the control traffic overhead in the network with the help of Multipoint Relays (MPRs) [9]. The MPR idea is the key concept behind the OLSR protocol. It is basically a node's one-hop neighbors in the network as shown in Figure 4.

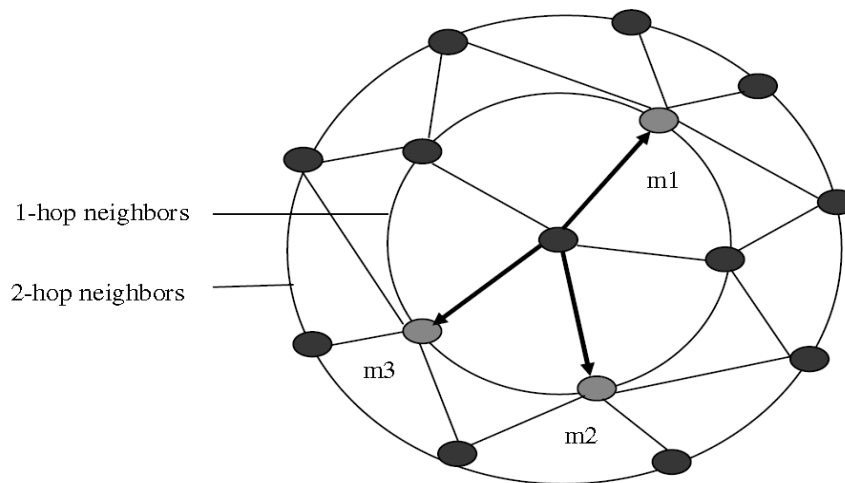


Figure 4. Multipoint relays of the OLSR network system

The MPR technique is used for route calculation between the source and the destination in the network. Furthermore, the MPRs support a mechanism for flooding the control traffic by minimizing the number of packet transmissions. However, they are to be involved in another task when the information of link state is announced in the network. The task includes announcements for the link-state information for their MPR selectors and then provides the shortest paths to all destinations in MANET. The MPRs are allocated from the one-hop adjacent nodes with symmetric or bi-directional connection, so it is possible to stay away from the hardships of experience during the packet transmission over a uni-directional link by deciding the path through the multipoint relays.

A HELLO message, Topology Control (TC) message and Multiple Interface Declaration (MID) message are three different types of control messages which OLSR uses. Due to the benefit of these messages that periodically runs, it can minimize the maximum time interval and also keep the routes safe incessantly to all destinations in MANETs. This feature makes the OLSR protocol more helpful for dense and large networks. Regarding OLSR protocol, more optimization can be obtained as compared to the pure link state algorithm in the larger and denser network [10]. OLSR is designed to work in such a way where a complete distribution algorithm can be achieved and free of central entities.

OLSR is categorized into core functionality and a set of auxiliary functionalities [11] while the core functionality specifies a protocol which can make a routing in a stand-alone MANET whereas each auxiliary behavior provides other functionalities, i.e. a scenario where a node establishes connectivity between the MANET and another routing

domain. The aim of dividing the OLSR into these two parts is to make a simple and easy understanding of the protocol and also to add complexities only where additional functionalities are needed. The core functionality explains OLSR interfaces and the mobile nodes present in the MANET. It includes the following components:

- Neighbor detection
- Packet format and forwarding
- MPR selection and MPR signaling
- Topology control message diffusion
- Route calculation
- Link sensing

2.1.1 Components of OLSR

Packet format and forwarding utility has been specified for the transport of all control messages and the optimized flooding mechanism in 32 bit format.

Link sensing of OLSR sends Hello messages regularly for sensing the connectivity of the link. For each interface, a separate Hello message is generated. This link senses results in a local link set which show the links between the local and the remote interfaces.

Neighbor detection is the main address of the nodes. The neighbor entries are closely connected to link entries. When a link entry is made, then the neighbor table is checked for any similar neighbor entries. If no hits are returned, then a new neighbor entry is created. The status of the neighbor entries must be updated accordingly if there are changes made to the link-set.

In the MPR selection and MPR signaling, a node selects a subset of its neighbors resemble when all the selected neighbors broadcast a message. At that time, the message should be received by all the nodes two hops away.

With the help of topology control message diffusion for calculation of the route Topology, control message diffusion supplies each node in the network with enough link state information.

With the help of route calculation the link state information through periodic exchange of messages, the interface configuration of nodes and route of each node is computed.

2.2 Temporally Ordered Routing Algorithm (TORA)

Temporally Ordered Routing Algorithm is a reactive routing algorithm based on the link reversal [12]. It is used in MANETs to improve the scalability by utilizing in multi hop networks. TORA makes scaling routes amid the destination source and the source which is created in the destination node by using the Directed Acyclic Graph (DAG). It should be noted that the shortest path theory is not being used in TORA. It measures another theory which uses four messages. The order of messages are listed as below:

1. Query message
2. Update message
3. Clear message
4. Optimization message

This layout is carried out by each node for sending various parameters through the destination node and source node. It should be pointed out that the nodes id (i) and (t) are the parameters for time to break the link, (r) Reflection indication bit, (oid) is the originator id and frequency sequence (d).

TORA makes the link from high to low. In the first step, the nodes which are the highest are set to null. I.e. (null, null, null, I) and destination is set to this pattern (0, 0, 0, destination). Whenever there is a change in the topology, the heights are modifying. It sends a query message including its route-required flag which is the way for a node needing a route to a destination. It should be noted that query packet contains a destination field that shows the intended destination, so query packet has a node id of the destination which is needed. Due to it, when a query packet arrives to the node with information about the destination it responses update to the reverse path then the update message sets the height value of the neighboring nodes to the node sending the update [12]. This process is illustrated in Figure 5.

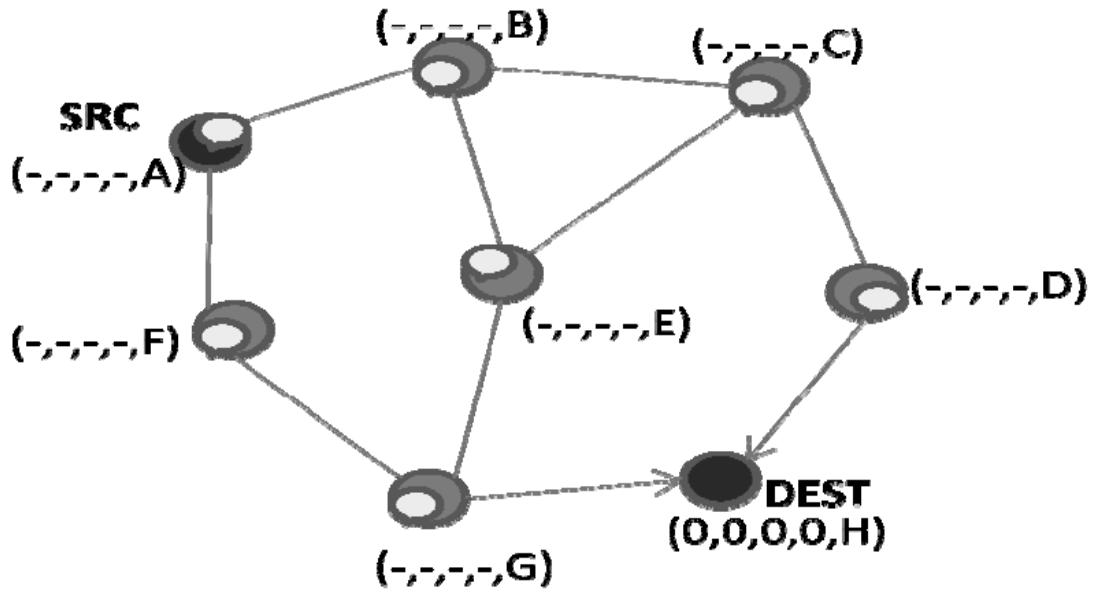


Figure 5. Route discovery for QRY message [12]

In this Figure, node *H* is the destination node and node *A* is selected as the source. Consider node *A* as only one-hop neighbor to the destination broadcasts a query message across the network, replies to a query then it sends back an update after the query arrives a node with information about the destination node. In this example, node *G* and *D* are shown as one hop far from the destination so they will propagate updates. The processes are presented in figure 6 below.

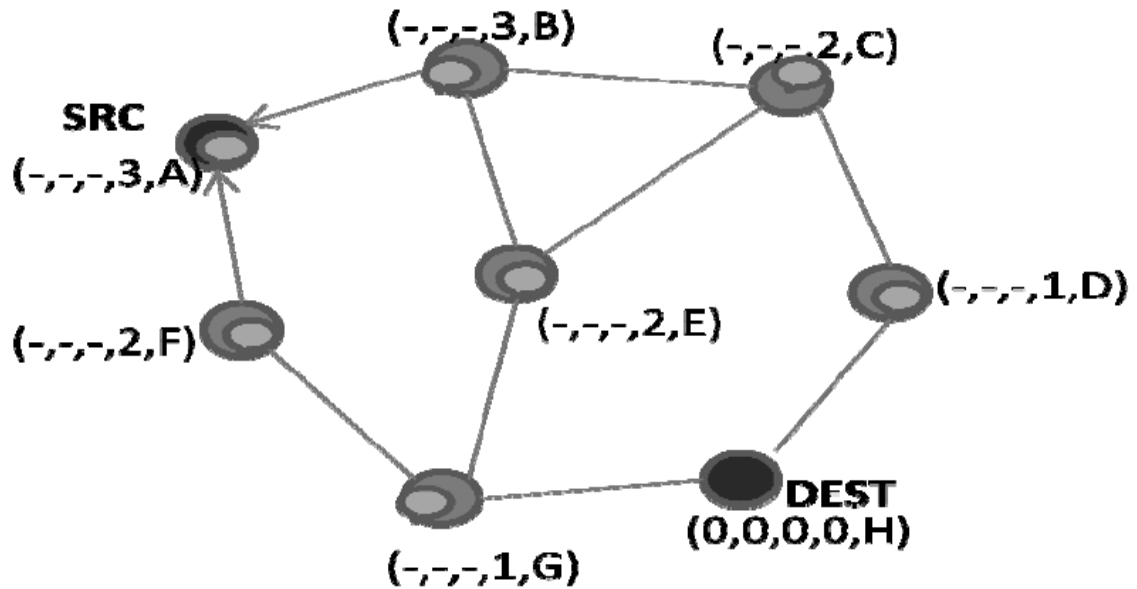


Figure 6. Route discoveries in TORA – update message [12]

There are some imperfections in this gradual procedure. In principal one, it generously depends on the number of activated nodes which were activated at initial setup [13]. The crack is that the reaction to traffic demands is not independent. So, it is dependent on the number of nodes in the network or rate of change of the amount of traffic. TORA is not good for the network with high traffic volume and also the traffic grows with a steep positive gradient. TORA guarantees to ensure reliability in the delivery of control messages and notifications about link status.

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

The Ad hoc On-demand Distance Vector Routing (AODV) [14] [15] discovers the new algorithm in operation of Ad hoc networks. In this protocol every node works as a separate router and when it needs a route, it starts to establish or obtain a route for itself. AODV does not require universal periodic routing advertisements because it is loop free route even when the link fails. Due to this fact, it requires just on the whole bandwidth

which is reachable to the mobile nodes. Note that it is substantially less than those protocols which are required for such advertisements.

AODV does not work with active paths neither maintains any routing information nor joins in any periodic routing table exchanges. The nodes in AODV do not have to discover and maintain the route to others nodes up to the time they want to make communication.

In most recent routing information between nodes, the concept of destination sequence number is used. Each node which maintains in route mathematically adds sequence number counter that is used to replace on cached routes.

There are six parts in AODV to create, delete and maintain routes defined as follows:

2.3.1 Path Discovery and Path Setup

In the path discovery when the node wants to start to communicate with other nodes, which is not valid in routing table, the path discovery will be started to work. Each node has two counters: node sequence quantity and a broadcast identification. The source node has to launch path discovery and it broadcasts the RREQ which is the abbreviation of route enquire packet to its neighbors. The mentioned RREQ has these fields:

- Broadcast ID
- Source sequence number
- Destination series number
- Source address

- Objective address
- Bounce count

Broadcast ID and source address singularly recognizes a RREQ. Broadcast identification is grown when the sources send a fresh RREQ. Every neighbor re-emits the RREQ to its own bystander or either gratifies the RREQ with releasing a route reply back (RREP) to the antecedent. When a node receives several editions of the identical route send out packet from different bystanders it refuses or drops the duplicate RREQ and does not send it out. It assumes that a compromising node arrogates a RREQ from it. Neighbors that have already arrogated a RREQ with the same send out ID and source address from them.

In the Reversing path setup RREQ has two kinds of arrangement quantity: The latest goal zone arrangement number familiar to the supplier and the supplier sequence quantity.

The destination ascertains total description of how fresh away route is before it can be accepted by the source to the destination and the source sequence number must be used to maintain new information about the reverse route to the source.

As presented in Figure 7, node *S* which is in the middle of the Figure decides to make route to the destination node *D*. Consider that node *S* does not have a root available in its routing table, so immediately it starts to broadcast RREQ message to its neighbors nodes for finding the destination node *D*. As can be seen nodes 1 and 4 are neighbors of node *S*

so they will receive the RREQ message. In process, nodes start to make an override link to the document from those gains -RREQ- from it. Because node 4 does not have information about the link which is connected to destination node, only rebroadcast is the RREQ to their neighbors node 5 and node 2. When the RREQ message goes through a source to different destinations, as illustrated in Figure 8 the reverse path from all nodes goes back to the source which will be setup automatically. It should be noted that this opposite route would be needed just when the node gains a RREP indorse to the node which has created the RREQ. In the creating node, before broadcasting the RREQ, all the growing IP address and the RREQ ID are buffered. From this procedure, the sender will not reprocess and re-forward the packet from the node which receives the packet again from its neighbors.

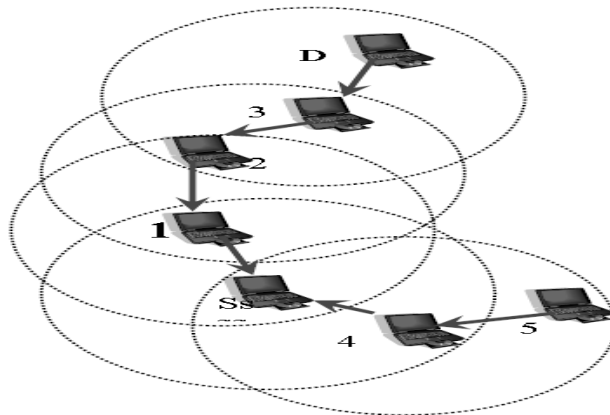


Figure 7. Reverse paths

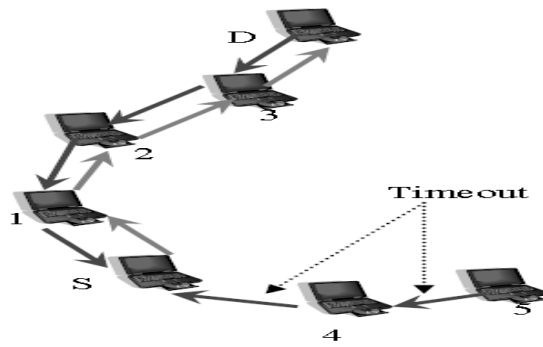


Figure 8. Forward paths

Finally in the forwarding path setup, a RREQ will receive a node which holds a current route to the destination or the destination itself. The receiving node first checks that the RREQ was received over a bi- directional link. If an intermediate node has a route entry for the desired destination, it decides whether the route is current by comparing the destination sequence number in its own route entry to the destination sequence number in the RREQ or If the RREQ's sequence number for the destination is bigger than that recorded by the intermediate nod. The intermediate node must not use its recorded route to respond to the RREQ. In place of it, the intermediate node rebroadcasts the RREQ. The intermediate node can reply only when it has a route with a sequence number that is greater than or equal to that included in the RREQ. If it does have a current route to the destination and if the RREQ has not been processed previously, the node then unicasts a route reply packet (RREP) back to its neighbor from which it received the RREQ. A RREP has these fields:

- Source address
- Objection address
- Destination series number
- Hop count
- Lifetime

Meanwhile, a broadcast packet will be arrived to the nodes which can support a route to the destination which source desires it and also the reverse path will be accepted to the source of the RREQ. Each node in the direction of the path sets up a forward pointer to the node from which the RREP arrived exactly when the RREP sends back to the source and it updates its timeout information for route entries to the source and destination and also records the latest destination sequence number for the requested destination.

The forward path process as RREP message travels through the nodes one, two and three from the destination node D to the source node S is illustrated in Figure 5. Nodes number four and five are not along the path determined by the RREP, and will delete the reverse pointers from these nodes after active route timeout. As matter of fact, a node acquiring the RREP spreads in the premier RREP for a mentioned supplier node in use of that supplier. If it gains more RREPs, it updates its routing info and propagates the RREP, only if the RREP includes either a larger goal array quantity than the previous RREP or the identical goal arrangement quantity with a less hop count. Then, as soon as the first RREP is received, the supplier node S can begin sending out data and also can later update its routing information if this gains a better route in comparison of the former ones.

2.3.2 Routing Table Management

Route request expiration timer is a timer which collaborates with reverse path entries routing. The termination time depends on the size of the ad hoc network and the route caching timeout or the time after which the route is considered to be invalid. The aim is to clear reverse path routing from the source to the destination from those nodes that are not useful on the path.

The address of active inner neighbors in the routing table's entry, through which packets for the given destination are received, is also saved. If it originates or relays at least one packet for that destination within the most recent active timeout period, a neighbor is assuming that it is active for that destination and notices that when a link along a path to the goal point cuts off this data is conserved so that all active document (source) nodes would be found. If it is in use by any active neighbors, a route entry is considered active. Route table entry will maintain about each destination for every mobile node which they are interested. Every route table entry has the following information:

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

The time out is rebooting to present time and active route timeout for each time when a route data is used to transfer information from source node to the destination node. The comparison process for destination grade quantity of the fresh route to the destination starts with the current route if a new route is found. Here, the new route is chosen only if it has a smaller metric to the destination and also if its sequence quantities are identical. Otherwise, the route with greater sequence number is selected as a new route.

In the link breakage, the node which wants to communicate must invalidate the existing route in the routing table entry. That node has to lean the infected nodes to destination and determine which neighbors are able to affect with this link breakage. In a final manner, the node can send the route error message (RERR) to the specified neighbors and if there are many neighbors, the route error message can be broadcasted or unicasted if there is only one.

Path maintenance in AODV is done in the following: movement of the nodes in the same zone does not affect the route of that way to the goal zone. If the supplier node mobile in active zone could again initiate the route discovery procedure for finding a completely fresh route to the goal point, then a special RREP message would be sent to the involved source nodes when the destination or some intermediate nodes moves. There is a special message to ensure about symmetric link in addition to detect link failures which is called periodic hello message. By using link layer acknowledgments such failures can be detected with far less latency. A link failure is also shown if it tries to forward a packet to the next hop fail. When the next hop fails and becomes unreachable to the node in upstream of it, multicast RREP with new sequence number

and hop count of infinity to upstream neighbors are started in order that those nodes subsequently relay that message to their active neighbors and etc. This process continues until the entire active source nodes are informed about it, then source node could restart the discovery process if it still requires a route to the destination and it receives notification of a broken link. For checking the required destination node in future, the obtain node can check the recently route which has been used. It should be noted that if the obtain node or some other nodes during the former route decides it would like to reconstruct new route to the goal zone, then the source node or any other nodes along the former route emit an RREQ message with a goal point series quantity of one more hug than the former familiar series quantity and for ensuring that it sets a fresh way which any of the nodes respond if they still regard the former route as reachable.

Also, there is local connectivity management in AODV. However, AODV is a proactive route and this uses greeting message periodically to its neighbors to ensure about connectivity of links. The Hello message is broadcasted to all members with time to life (TTL) equal one and this message is never forwarded more. Each node updates lifetime of the owner information in routing table of itself whenever it receives Hello message. Furthermore, the data in the route table is known as lost when the host receives no information from the neighboring. Then, the nodes inform the other nodes by broadcasting the RRER message for link breakage.

With the purpose of the local linkage management with hello messages, each greeting message is emitted by lists of nodes from which nodes were received. Due to this process it is able to ensure that only nodes with bidirectional connectivity are considered

to be neighbors. Each node checks to make sure that it uses only routes to neighbors that have heard the node's hello message.

AODV support local repair. In local repair the host can fix link breakage locally anytime if the destination is not farther than the amount of hops which is specified. For repairing the brakeage, the host will rise the sequence number of destination and broadcast RREQ communication to the controller node and the TTL for the IP header should be measured and saved up to locally mending. For the RREP messages, the host waits for its RREQ message for considering the amount of time. If the RREP message is not received by the owner, then the routing table condition for the entry becomes out of reach. The hop count metric would compare if host received the RREP message. The RERR with the N field set up is broadcasted if the hop metric from the message is bigger than the former one. The N field in the RERR notices that the owner has locally mended the link so the entry in the table should not be omitted and the received RREP message would be considered as the original RREP communication. The source code of AODV is available in appendix A.

2.4 Comparison of Selected Routing Protocols

The differences between three MANET routing protocols are show in Table 1.

Table 1. Differences between three MANET routing protocols

Parameters	AODV	OLSR	TORA
Routing mechanism	On demand	Table driven	Table driven or on demand
Multiple routing mechanism	NO	NO	YES
Loop free routing	YES	YES	YES
Multicasting possibilities	YES	NO	NO
Beacons	Yes, hello messages	YES	NO
Structure of the route mechanism	Flat	Flat	Flat
Routing method	Broadcast or Flooding	Flooding	Broadcast
Update of routing information	As required	Periodically	As required
Network information maintenance	Route table	Route table	Route table
Depth of information	Up to neighbor nodes	The whole topology	The height of the neighbor nodes
Control message	Only hello message used	Hello, TC and MID message	LMR message
Advantages	Much more efficient to dynamic topology	Trim down the number of broadcasts	Multiple loop free and reliable routing
Disadvantages	Scalability and large delay	The MPR sets could be overlapped	Temporary routing loops results in larger delay in the network

2.5 Review of the State of the Art

Many researchers are continuously working on MANET environments area in order to find out efficient routing protocols suitable for real time network scenarios. Different routing protocols follow different strategies to avoid loop within the network. If the destination node is not available in the network or any link fails, the routing may face count to infinity loop problems. To ensure the loop free routing, protocols use destination sequence number and DAG algorithm (it calculates path always in unidirectional) and feasible distance etc.

Where TORA uses a link reversal algorithm and AODV uses a sequence number for each destination. AODV and OLSR has shown greater packet delay and network load compare to TORA. Experimental results also show that TORA has lower throughput compared to AODV and OLSR. In heavy traffic environment, AODV works better than OLSR and TORA in high congestion network scenarios. ([2], [17], [18], [19]).

In papers [20], [21] OPNET model 14.5 is used to investigate the performance of routing protocols OLSR, AODV, DSR and TORA with varying network sizes, node mobility and traffic load. Experimental results reveal that TORA shows the better performance under high traffic loads in medium and large sized networks. DSR is well suited for small size networks with lower node mobility. It also performs better at high node mobility in large networks. AODV performs well in medium sized networks under high traffic load. OLSR performs comparatively better in many cases than others. However, its performance suffers and degrades when mobility and traffic load are increased. TORA delivers much lower throughput than AODV and OLSR. In AODV, the decision

is taken based on distance reported in the reply associated with the destination sequence numbers. LDR also uses the sequence numbers but it is controlled by the destination to which it belongs. Ordering of nodes is done based on the label to each destination and it always ensures loop free in any scenarios using label which is combined with feasible distance and destination sequence numbers [16].

With variable pause times and for random waypoint model in QualNet simulator, simulation results show that with respect to end-to-end delay, packet delivery ratio and TTL based hop count AODV has shown better performance than DSR and ZRP ([22], [23], [24]).

With respect to packet delivery ratio, DSR and AODV show better performance than ZRP. David Oliver Jorg has analyzed the performance of AODV, DSR, LAR and ZRP with the various sizes of mobile ad-hoc networks [25]. In case of small sized networks, all protocols have shown better performance, but only AODV supports more packet delivery in large network where ZRP and DSR completely fail.

A new approach of routing protocol, which is FZRP, was introduced which combines with zone routing protocol and hierarchical proactive-Fisheye Routing protocol [26]. It normally works on two levels of zone; basic zone and extended zone. This approach offers more advantages in a larger zone with a small increase of maintenance overhead. With respect to different metrics average maintenance overhead, average route finding cost and hit ratio, FZRP shows better efficiency than traditional ZRP for different zone sizes such as 2 and 4 etc. The Table below shows some of the same works.

Table 2 summaries some recent works which had been done using OPNET simulation. Detailed simulation and parameters could be observed from this table. In some of the simulation, results had been drawn respect to time as X axis but here in this thesis the different number of nodes, different file size and effect of different speed is shown in the results.

In this study, the number of nodes, file (data) size and nodes speed were changed by getting the idea from references ([20], [21], [29]); which are used to investigate the performance of routing protocols OLSR, AODV, DSR and TORA with changed number of nodes, speed of nodes and data size.

Table 2: Comparison with other works

Ref No	Routing Protocol(s)	Simulation Time	No. of Nodes	Application	Node Speed (m/s)	File Size (bytes)	Mobility model	Performance metric	environment (m x m)
[24]	AODV DSDV	600	5, 3	-	5	-	-	Throughput	-
[27]	AODV OLSR DSR	300	50, 120	FTP	-	5000000	Random waypoint	Throughput Delay Drop packet	1000 x 1000
[28]	AODV OLSR DSR TORA	600 1800	16	-	Fix, 2, 20	1, 64	Random waypoint	Throughput Delay	1000 x 1000
[29]	AODV DSR	3600 500	20, 40 4,25	FTP	2 and 6	1024	-	Routing discovery time Avg. number of hops Network delay Network throughput	4000 x 4000
[6]	AODV OLSR DSR	600	10,30	VOIP	Laptops and sensors	1024	up, up-right, up-left, down, down-right, down-left, left and right	Average routing traffic Average load Throughput	laptops and sensors
[30]	AODV	-	fewer and large number nodes	-	28, 14	1024	-	Routing load Throughput	-
[31]	AODV DSR OLSR	240	20, 40, 80	FTP	-	512	Random waypoint	Delay Network load Throughput	1000 x 1000

Chapter 3

OPNET SIMULATION ENVIRONMENT

The behavior of mobile ad hoc network for researchers is too expensive, hard, and time consuming in real environment. Hence, for imitation to appraise and analyze MANETs with varied routing protocols, research community usually relies on computer, but the results of simulation are a little different from real environment. However, doing simulation study is still supported well in understanding [32] the behavior of such system at different stage. Different simulators are used to design MANETs, i.e., NS-2/3 (Network Simulator-2/3) [33], OPNET (Optimized Network Engineering Tool) [32], and GloMoSim [34].

In this study OPNET was selected for the simulation. Since this program is one of the most measurable and efficient simulation tools due to its powerful characteristic such as comprehensive graphical user interface and animation, also it contains hundreds of protocol and vender devices model with giant flexibility for examination and analysis. Furthermore, it provides object oriented modelling and open source code model which provides easier understanding of the system. Due to these simulation tools, users are able to maximize availability of communication networks also optimize performance [35].

This chapter describes the architecture of OPNET simulator in 4 parts; OPNET Architecture, MANET Model Architecture in OPNET, Configuring routing protocols in OPNET, and Taking results of Route.

3.1 OPNET Architecture

OPNET supports big modeling, evaluate communication networks and distribute systems. It includes a lot of instruments that each of them focuses on special views of modeling role. These tools are divided into three fields:

- Specification
- Data collection
- Simulation analysis

The orders of these phases are important. It looks like a cycle which returns back to specification analysis. Also specification analysis is divided into two compartments as beginner specification and regeneration. Where the second phase is part of duplication cycle is illustrated in Figure 9.

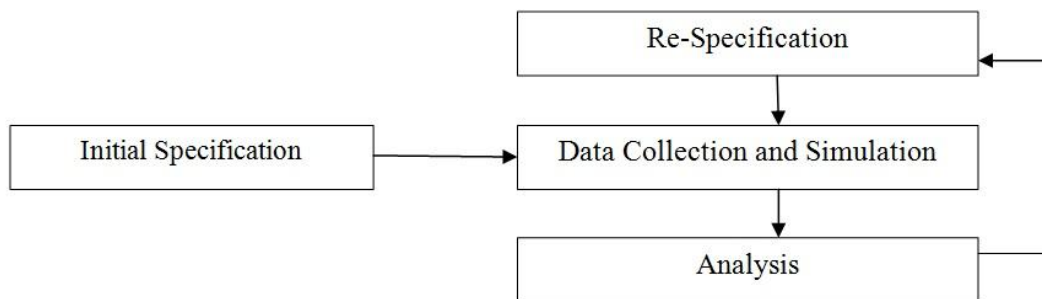


Figure 9. Simulation process for OPNET

3.2 Architecture of MANET Models in OPNET

Routing protocols OLSR, DSR, AODV and TORA are reachable at IP layer through MANET model structure. OSPFv3 for the MANET model is under development. Protocols of TORA, DSR, GRP, AODV and OLSR are ready for use in OPNET version 17.1. Node model component of a MANET node is illustrated in Figure 10.

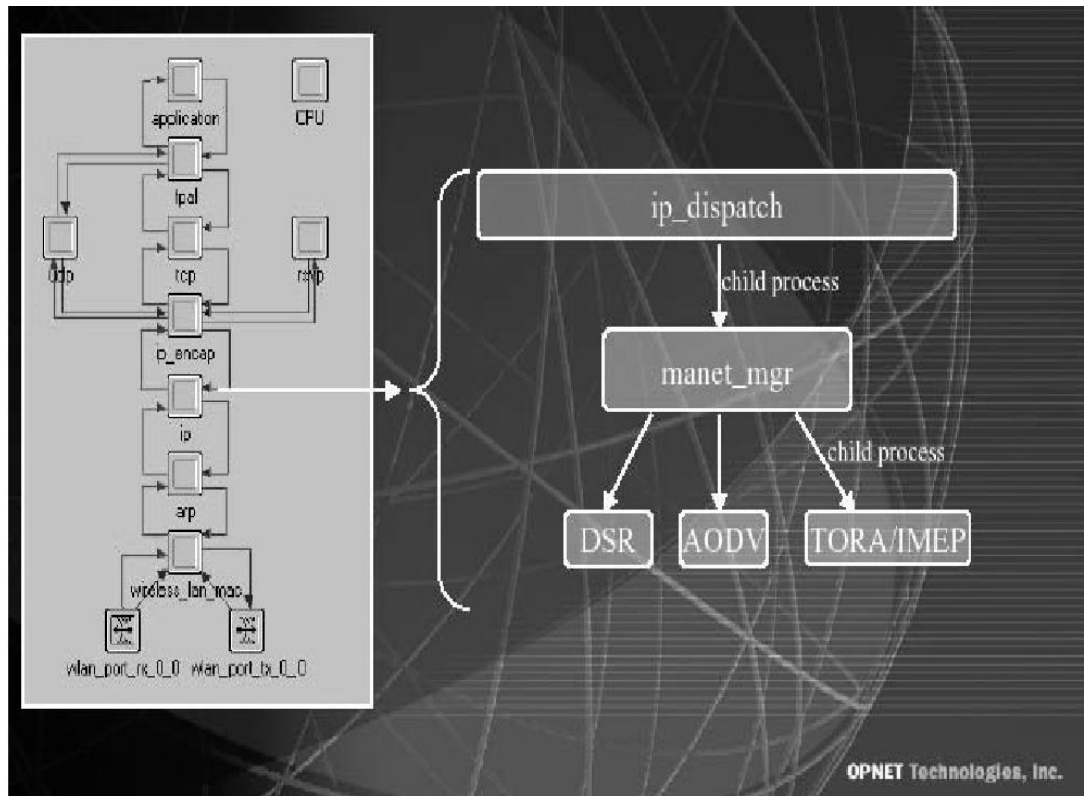


Figure 10. MANET model architecture [36]

The figure defines the node model architecture of a MANET's node. Creation of a child process manet_mgr function which controls the whole ad hoc routing protocols in the OPNET and enrich a common interface to multiple Ad hoc routing protocols. It is made from the function ip_dispatch of the ip_encap process which is root process for IP in

MANETs routing protocol. One of more child processes for required MANETs protocol as setup in parametric system is the Manet_Mgr, since the MANETs of this node would be a Wireless LAN work zone operating in mobile Ad hoc mode.

We have different models of nodes in MANETs. All MANET adroit nodes are included in the contents of in the MANETs object palette as illustrated in the Figure 11 to simulate different routing protocols while nodes of the mentioned object palette are used in the mobile Ad hoc network models. Prevalently using nodes in MANETs network models are defined in the following;

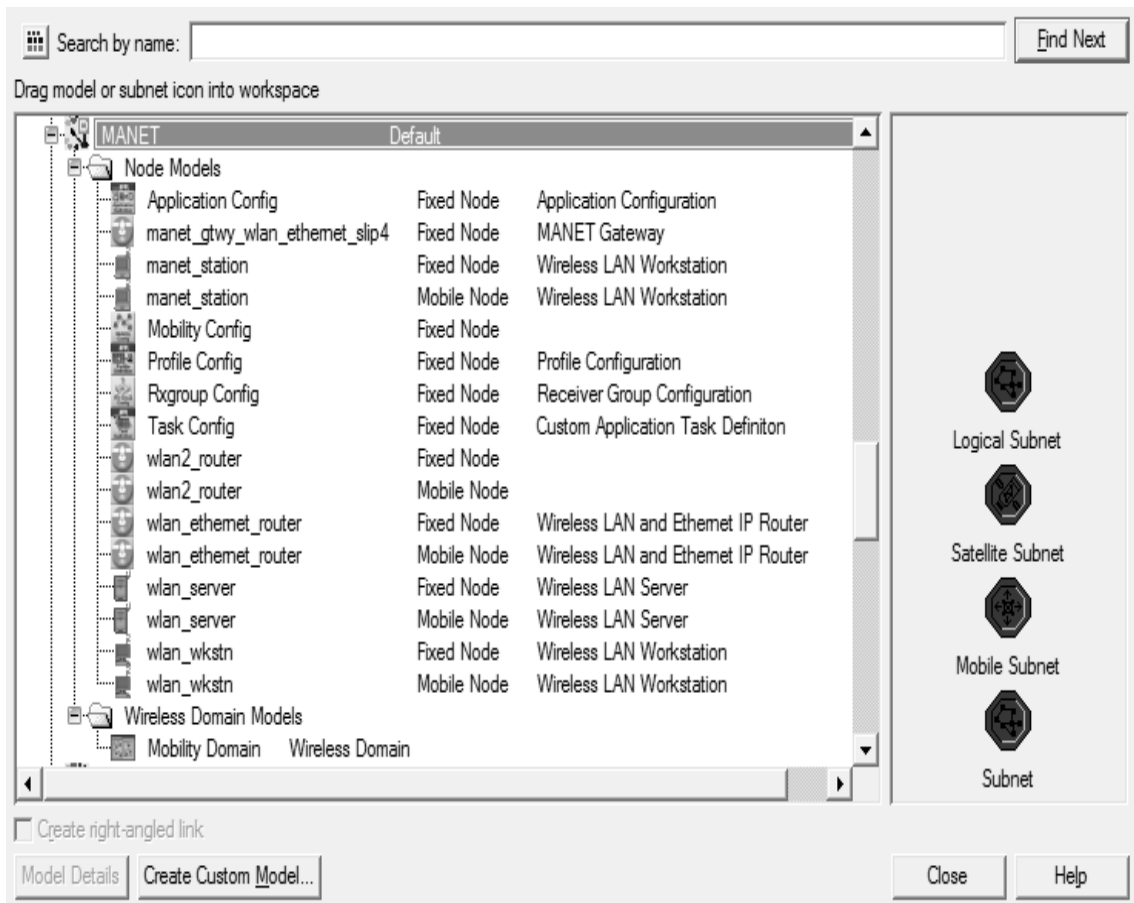


Figure 11. MANET object palette

- Wireless LAN servers and workstations

In a MANET network model these node models could be used for professionalized application traffic like E-mail, FTP and HTTP on TCP on IP over wireless LAN. These nodes would be set to start the cycle for each MANET routing of protocol also configured for specific way.

- MANETs Stations

This station can be used over IP on wireless LAN the node models of MANET to generate raw packets. They can be configured as a destination traffic or source and can be functioned to run each MANET routing protocol.

- Wireless LAN routers and MANET gateway

These nodes can perform as an access point role in ad hoc network. These nodes of object palette could also connect the mobile nodes of network to the IP based networks when MANET gateway is enabled.

- Profile configuration

Profile configuration describes application activity models or shape of user or group of users over a period of the time while it is possible to have some varied profiles running on a considered LAN or work zone which these profiles can present varied user teams.

- Application configuration

A profile is assembling different application definitions. There are designated for some parameters like duration, start time and repeatability for each application definition.

Also, to have two completely same applications with different application parameters, different names to identify two identical applications with varied usage parameters as two distinct application definitions are acceptable to use.

- Rx group configuration

Rx group configuration is used to estimate a group of possible receiver's node that could do the communication role. This tool could greatly accelerate a simulation by getting rid of receivers which do not match.

- Configuration task

It is used for special applications which are configuration.

- Mobility configuration

Mobility configuration is used to define movement of nodes based on the settled parameters which individual nodes reference to model mobility profile.

3.3 Configuring Routing Protocols in OPNET

By using the right button of the mouse over the nodes fixed in the project modifier, a fresh frame of window will be popped up for editing ascribed values of unlike parameters. The way of configuring routing protocols parameters in OPNET 17.1 is shown in Figure 12.

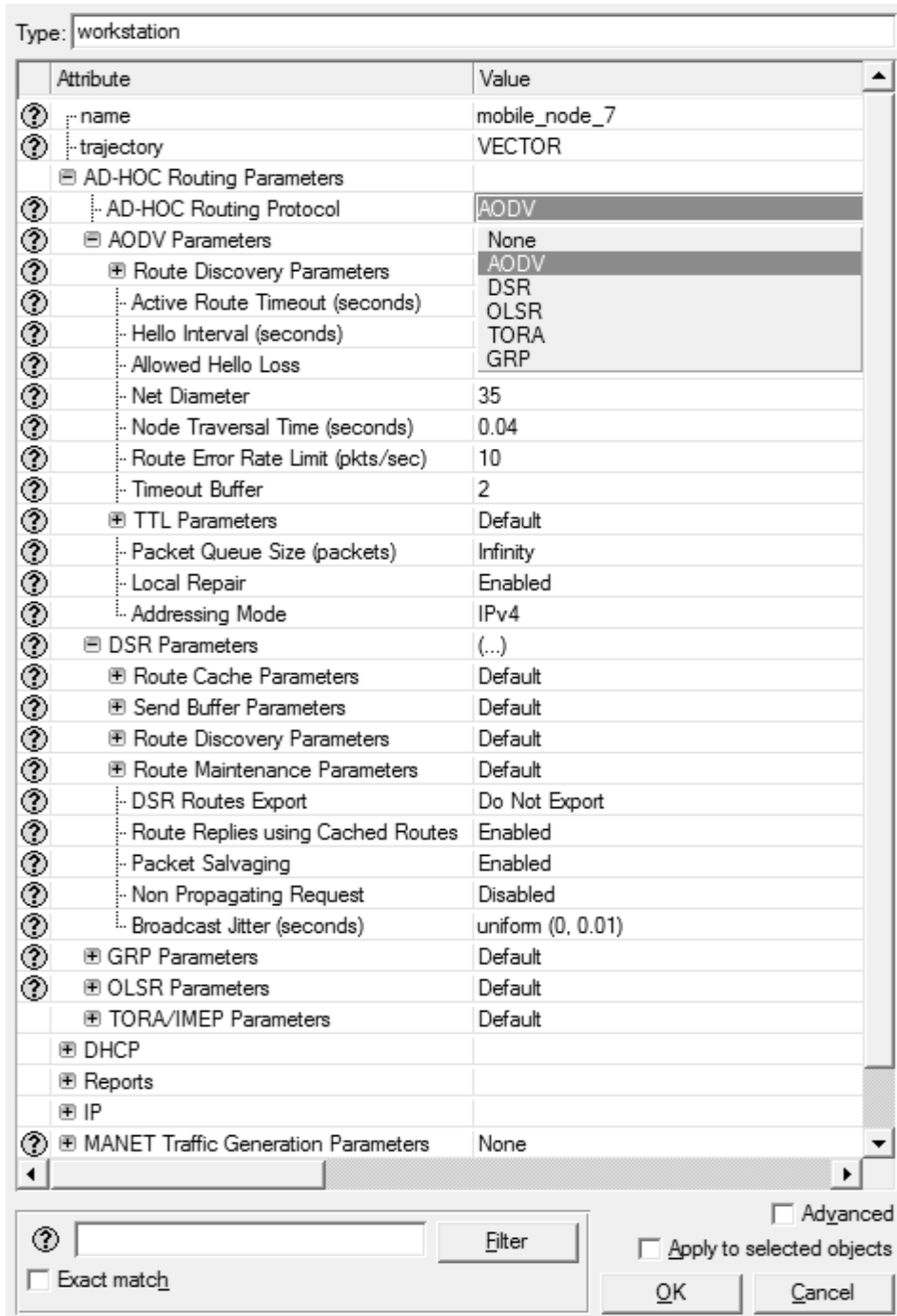


Figure 12. Routing protocol configuration in OPNET

As illustrated in the figure, it is possible to select 5 routing protocols in OPNET 17.1; AODV, DRS, TORA, OLSR, GRP and change individual parameters.

3.4 Taking Results of Simulation

To choose individual DES (Discrete Event Simulation) statistics right click on the project editor. There are different statistics available to be simulated as can be seen in Figure 13.

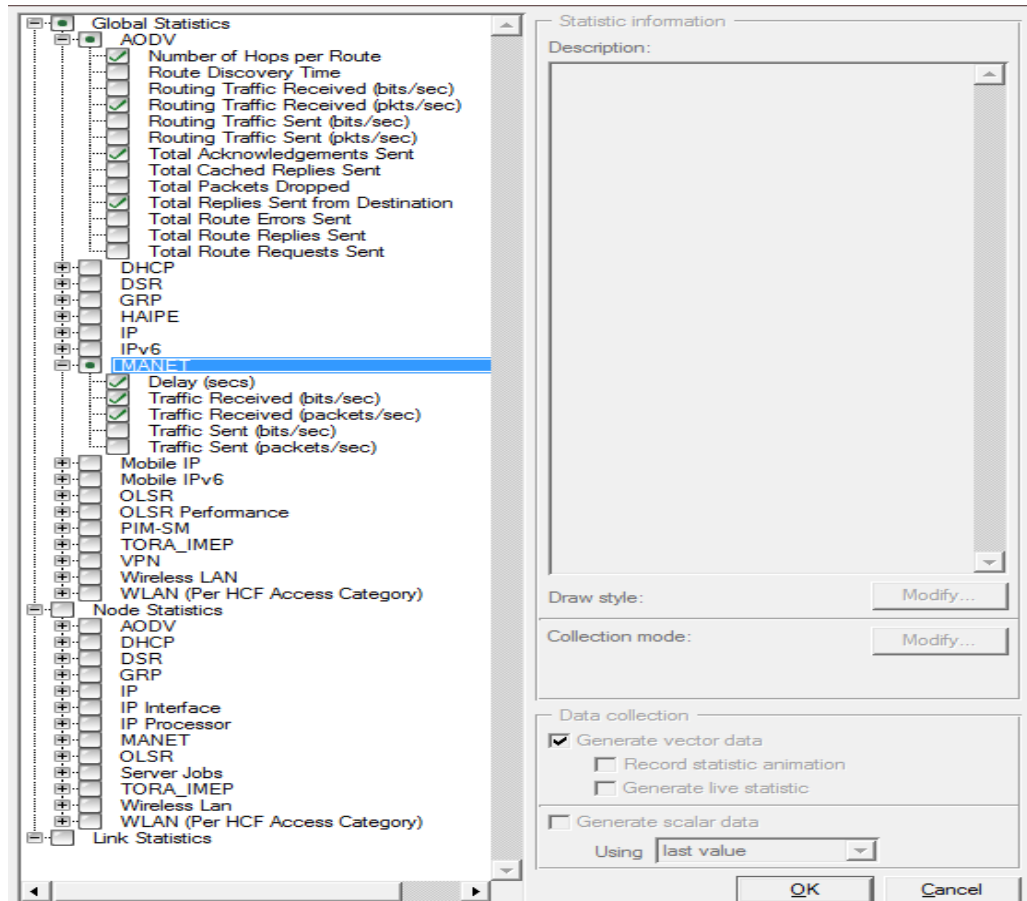


Figure 13. Choosing statistics

Chapter 4

MODELING OF MANETs IN OPNET, SIMULATION SETUP AND RESULTS

In this chapter the selected performance metrics, simulation setup, and modeling of network protocols with default parameters using a MANET model in OPNET17.1 are defined. Furthermore, the network scenarios are explained and simulation results are compared.

4.1 Performance Metrics

The performance of routing protocols was analyzed using performance metrics, average network throughput, average end- to-end delay and average network load.

Average throughput: It is the total amount of packets rate bear in case of data loss which is received by a destination node. High throughput is always expected for any routing protocol.

Throughput = number of bits contained in accepted packet / simulation time.

Average end-to-end delay: It is the average delay of routing discovery, waiting of packets in the interface queues and transmission of the MAC layer data packets from source to destination. It is also called data latency. It is measured by differences of time taken between the generation of a data packet and the last bit of arrival at the destination.

Average network load: It represents the total load (in bits/sec) submitted to WLAN MAC layer by all higher layers in all WLAN nodes of the network. All of the data traffic is received (in bits/sec) by all the 802.11e-capable WLAN MACs in the network from higher layers for each access category. Higher layer data packets are assigned to the access categories based on their user priority (Type of Service (ToS)) values [37]. The network load occurs when there is more traffic coming on the network, and it is difficult for the network to handle all this traffic. The efficient network can easily cope with large traffic coming in. [37]

High network load affects the MANET routing packets and slow down the delivery of packets for reaching to the channel [38], and it results in increasing the collisions of these control packets. Thus, routing packets may be slow to stabilize.

4.2 Modelling of MANETs in OPNET and Simulation Setup

In order to simulate a MANET network, there is a need to design a virtual network environment in OPNET. In this study, OPNET version 17.1 is used which supports AODV, DSR, GRP, OLSR and TORA routing protocols in total. All devices with IP address version 4 were auto configured. In order to complete the project, a total of 60 sets of simulations were designed. To collect statistical data, all of the scenarios were run for 300 seconds. In order to design a MANET with a routing protocol, the steps below must be followed:

1. File => Project name: AODV

Scenario name: A name for each set of simulation must be given (For example: 20 nodes with file size 512 bytes and maximum speed 5 m/s in AODV)

Create empty scenario

Network Scale: Campus

Specify size: X span: 1000, Y span: 1000 and units: meters

Model family: MANET

Figure 14 shows review of these settings.

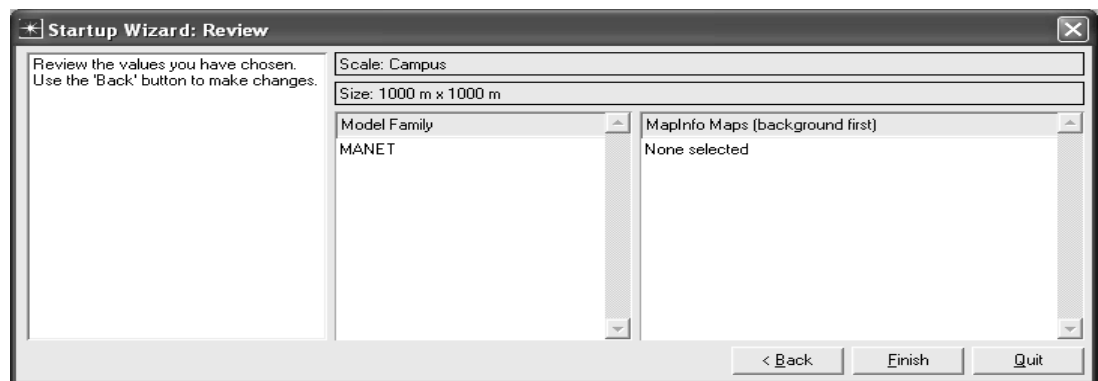


Figure 14. Review of startup wizard

2. Application configuration: application configuration form object palette is chosen and inserted on the campus network as shown in Figure 15.

Edit Attribute => Name: App Conf

Application definition => Number of Rows: 1 (Number of application during simulation -only FTP is used)

Application name: FTP.APP

Description: FTP with medium load configuration

Inter-request time (seconds): exponential (720)

File size: 512 bytes

These settings are valid for all sources in the system.

FTP application: FTP is a file transfer protocol used by FTP applications to perform huge data transfer from server to user agents. Main objects of FTP include [39] 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.

Inter-request time: 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.

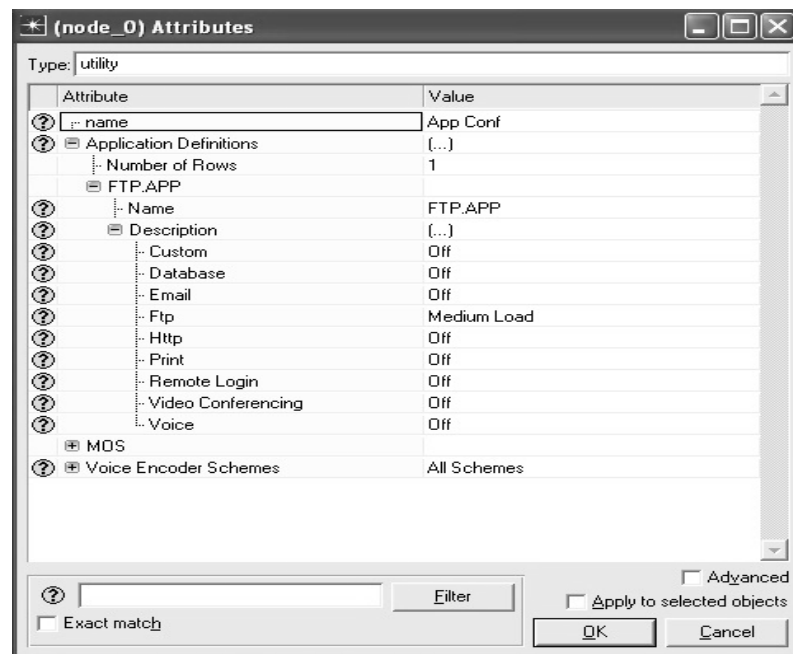


Figure 15. Application configuration attribute

If we set inter-request time (secs) attribute to exponential (720) and file size (bytes) attribute to constant (512), in our FTP application we are transferring 512 bytes every 720 seconds. Since our simulation time is 300 secs each source may only transfer one, 512 bytes file.

In [39] it is shown that in 1 second of elapsed (actual) time, OPNET Modeler has simulated 19 minutes and 25 seconds of network time. The entire simulation should take less than one minute to complete—the elapsed time varies according to the speed of the computer.

For example, in one of our case, in order to simulate the AODV protocol with 20 nodes, 512 bytes and with other fixed parameters elapsed (actual) time measured as 2 secs as shown in Figure 16.

Configuration	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 Suffix
ADDV 100 NODES 512 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	1m 53s		15,867,203	50,161	139,878		3	-DES-1
ADDV 100 NODES 1024 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	1m 49s		15,717,986	51,541	144,844		3	-DES-1
ADDV 100 NODES 2048 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	1m 46s		15,201,523	49,977	142,843		3	-DES-1
ADDV 100 NODES 4096 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	1m 44s		15,037,281	52,138	144,654		2	-DES-1
ADDV 100 NODES 512 PACKET SIZE 30 SPEED	Completed	localhost	5m 00s	5m 00s	1m 46s		15,098,090	50,416	143,025		2	-DES-1
ADDV 100 NODES 512 PACKET SIZE 50 SPEED	Completed	localhost	5m 00s	5m 00s	1m 45s		14,887,057	50,315	141,718		2	-DES-1
ADDV 100 NODES 1024 PACKET SIZE 30 SPEED	Completed	localhost	5m 00s	5m 00s	1m 51s		15,929,978	51,690	143,090		2	-DES-1
ADDV 100 NODES 1024 PACKET SIZE 50 SPEED	Completed	localhost	5m 00s	5m 00s	1m 53s		15,851,389	50,891	140,687		2	-DES-1
ADDV 100 NODES 2048 PACKET SIZE 30 SPEED	Completed	localhost	5m 00s	5m 00s	1m 52s		16,175,841	51,482	143,986		2	-DES-1
ADDV 100 NODES 2048 PACKET SIZE 50 SPEED	Completed	localhost	5m 00s	5m 00s	1m 51s		15,742,363	50,191	141,643		2	-DES-1
ADDV 100 NODES 4096 PACKET SIZE 30 SPEED	Completed	localhost	5m 00s	5m 00s	1m 50s		16,013,345	51,592	145,700		2	-DES-1
ADDV 100 NODES 4096 PACKET SIZE 50 SPEED	Completed	localhost	5m 00s	5m 00s	1m 50s		15,964,584	50,016	144,803		2	-DES-1
ADDV 80 NODES 512 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	1m 02s		9,089,208	42,430	147,007		2	-DES-1
ADDV 60 NODES 512 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	30s		4,349,654	35,564	147,057		2	-DES-1
ADDV 40 NODES 512 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	11s		1,663,156	27,171	154,267		2	-DES-1
ADDV 20 NODES 512 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	2s		248,244	20,304	148,471		2	-DES-1
ADDV 40 NODES 1024 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	11s		1,700,793	27,071	154,617		2	-DES-1
ADDV 40 NODES 2048 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	11s		1,723,346	27,179	153,609		2	-DES-1
ADDV 40 NODES 4096 PACKET SIZE 5 SPEED	Completed	localhost	5m 00s	5m 00s	11s		1,721,480	27,290	154,309		2	-DES-1

Figure 16. DES Execution Manager

From Figure 16 results, it is observed that elapsed time is increasing when the number of nodes is increasing but it is slightly decreasing when file size is increasing for fixed number of nodes.

3. Profile configuration: profile configuration form object palette is chosen and inserted on the campus network as shown in Figure 16.

Edit attributes: Name: Pro Def

Profile definition: Number of rows: 1

Profile name: Pro FTP

Application: Number of rows: 1 (only FTP)

Profile name: FTP APP

Start time offset (seconds): Constant (0)

Start time (seconds): Uniform (100,300) - start to collect statistics after 100sec up to end of simulation.

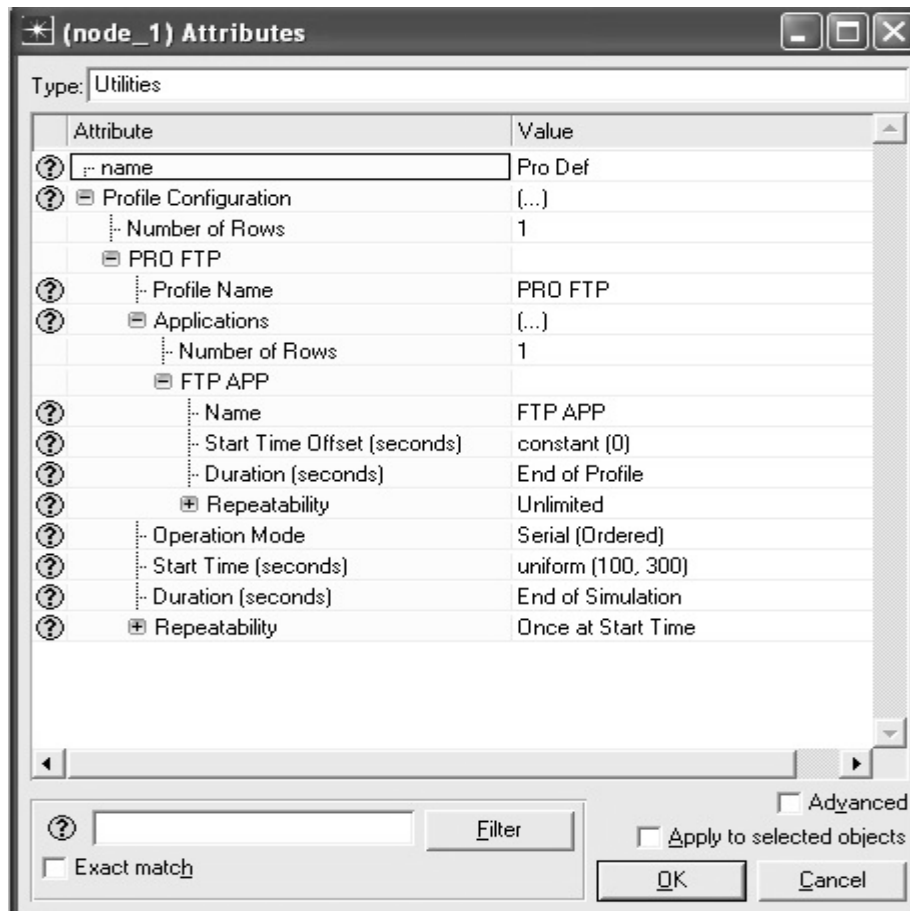


Figure 17. Profile configuration attribute

A profile 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".

Various loading characteristics for the different applications on this profile can be specified. Each application is described in detail within the application configuration object. The profiles created on this object will be referenced by the individual workstations to generate traffic.

4. Mobility configuration: The mobility profile defined in the mobility configuration can be specified to model the mobility over the nodes. In this particular design, random waypoint mobility model has been specified [29]. Generally, mobile nodes engaged in a network move randomly and take random destinations. Moreover, random mobility model is more appropriate for simulation studies. Therefore, mobility configuration form object palette is chosen and inserted on the campus network as shown in Figure 17.

Edit attributes => Name: Mob

Random mobility profiles => Number of rows: 1

Random Waypoint Parameters: X and Y axis (meters): (min:0 ,max:500)

Speed (meters/seconds): uniform (0, 5)

Pause time (seconds): constant (100)

Start time (seconds): constant (0)

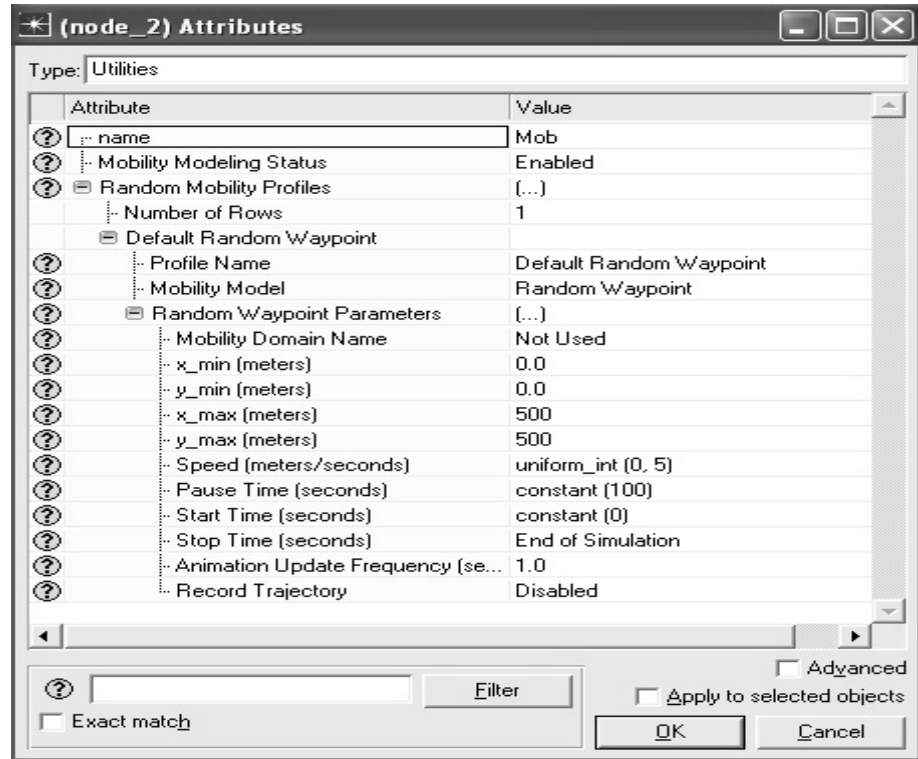


Figure 18. Mobility configuration attributes

- Wireless LAN Workstation (Mobile Node): Wlan wkstn form object palette is selected (For example 20 of them are inserted on the campus network as shown in Figure 18).

Edit attributes => Trajectory: Vector

Ad-Hoc routing protocol: AODV

Routing parameters: Default (see Fig 20 in Appendix B)

Applications: Destination preferences: none

Source preferences: none

Supported profile: FTP Profile

Traffic type: all discrete

Destination preferences: They provide mappings between symbolic destination names specified in the Application Definition or Task Definition objects and actual names specified in Deploy Application dialog box with Source and Server buttons for each node. Each symbolic destination can map to a set of real destinations, in which case a destination will be chosen based on its relative weight. The following applies only to Standard Applications and not to Custom Applications:

If Destination Preferences is set to None, then a random destination (server) will be chosen from among the existing number of nodes that supports the application of interest. Selection weight specified in the Supported Services attributes on the destination will determine the probability with which the destination will be chosen. So here none has selected as a Destination Performances to select random destination from among of destinations.

If Source Preferences is set to None, then a number of client (source) maybe selected from among the existing number of nodes -1 that supports the application of interest. In our simulations, if there are n nodes in the system we have selected remaining $n-1$ nodes as source node. For example; if there are 20 nodes in the system one of them will be selected as a server node randomly and the remaining are used as source node.

Supported profile: It specifies the names of all profiles which are enabled on this node. Each profile is defined in detail in the profile configuration object that can be found in the "utilities" palette.

A profile describes user behavior in terms of what applications are being used and the amount of traffic each application generates. Profiles can be repeated based on a "Repeatability pattern". It can also execute more than one profile on a particular device.

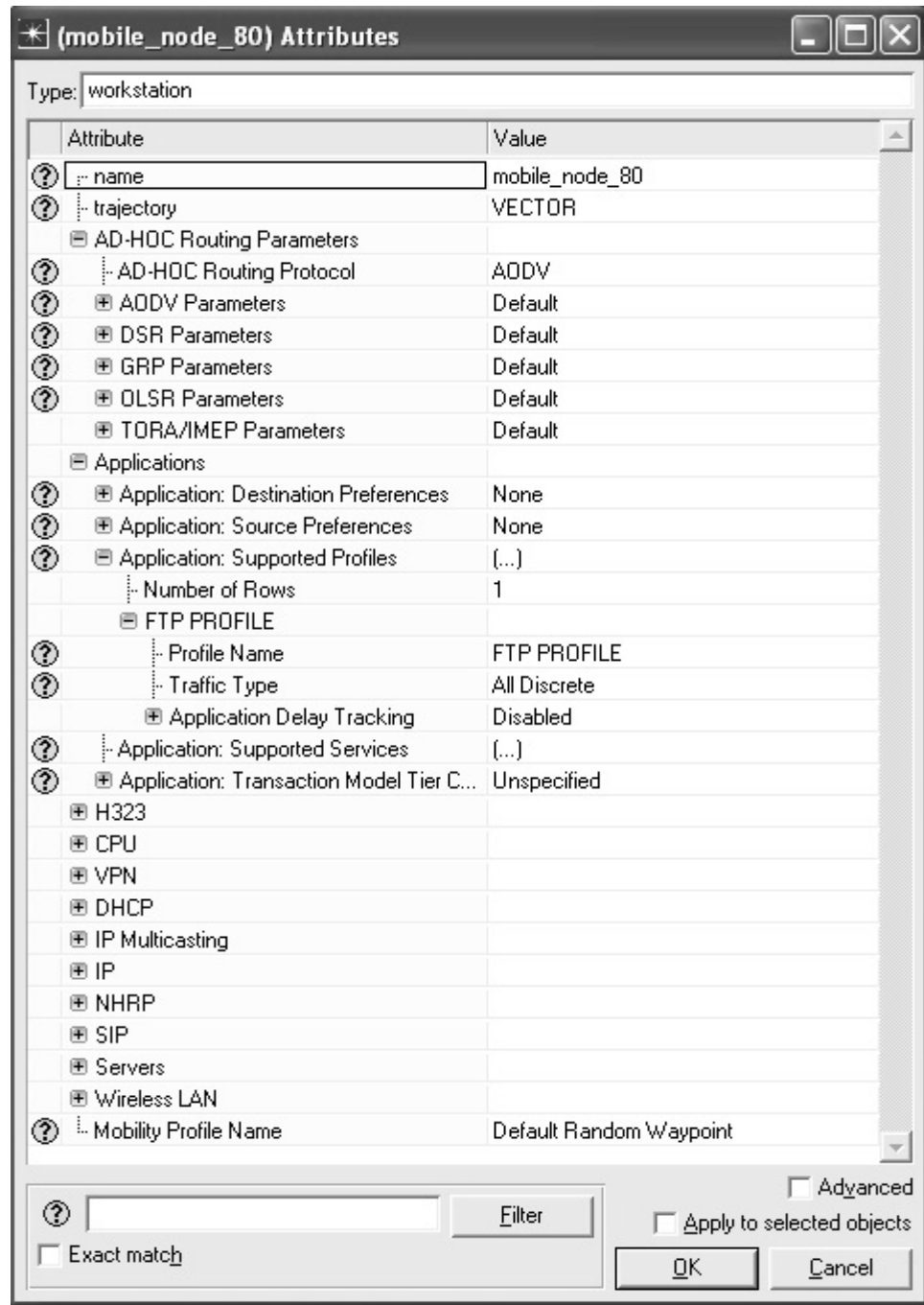


Figure 19. Wireless LAN Workstation attribute

Traffic type: It specifies the type of traffic that will be generated for this profile. If it is set to All Discrete, discrete data packets will be generated for the application contained as part of this profile.

This attribute cannot be configured directly. To change the value of this attribute, use the utility, "Protocols / Applications / Deploy Defined Applications...".

Application Deployment dialog box helps in deploying the application in the network. To configure the nodes for server select those in the network tree on the left hand side of the window and then assign them to the selected tier in the right hand side, so from number of servers one of them could be selected randomly as main server as shown in Figure 19.

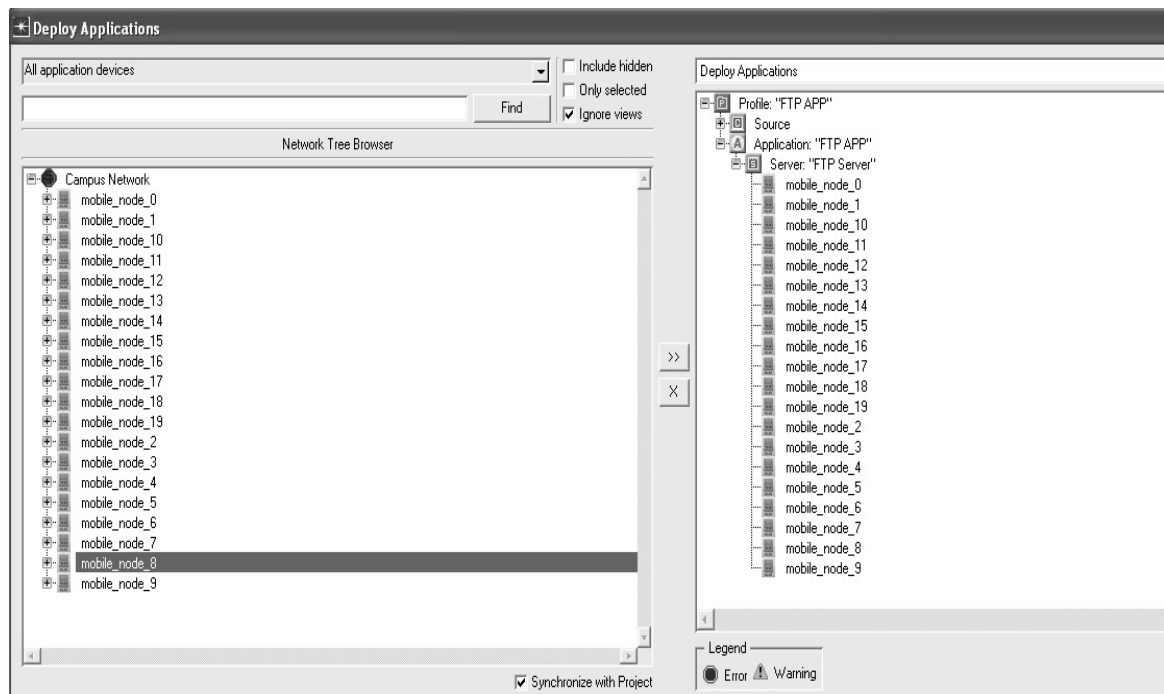


Figure 20. Deploy application setup

In a similar way, to configure nodes as source select those in the network tree on the left hand side of the window and then assign them to the selected tier in the right hand side under the source button.

4.3 Simulation With Different Ad hoc Network Scenarios and Results

The results obtained during the simulation are depicted through a number of scenarios. In our simulation study, there are three types of different scenarios based on the number of nodes, different file (data) sizes and speeds as performed with performance metrics average throughput, average end-to-end delay and average network load for AODV, OLSR, and TORA routing protocols. Each scenario is discussed separately so as to provide detailed analysis.

4.3.1 Investigation of Different Number of Nodes

In first scenario was prepared in which there were 20, 40, 60, 80 and 100 mobile nodes from the object palette window of OPNET Modeler 17.1 and pasted all of them in the workspace window and routing protocols AODV, OLSR and TORA were used individually. After the processes of inserting application configuration and profile configuration from object palette to workspace window, the settings had to be done according to the requirements. The FTP was selected as traffic with medium load; FTP file size set to 512 bytes. Mobility configuration was also inserted into workspace window. In the first scenario the maximum node speed was set to 5 m/s and then random waypoint mobility model was set to MANET as a profile. All these attributes are illustrated in the table below:

Table 3. General attributes for scenario 1

Attributes	Value
Number of nodes	20, 40, 60, 80, 100
File(data) size	512 Byte
Protocols	AODV, OLSR, TORA
Simulation run time	300 seconds
Simulation area	1000 m * 1000 m

Table 4. Mobility attributes for scenario 1

Mobility	Speed (seconds)	Uniform (0,5)
	Pause time (seconds)	Constant (100)
	Start time (seconds)	Constant (0)

Table 5. Application configuration attributes for scenario 1

Application configuration	FTP (Medium load)	Inter request time (seconds)	Exponential (720)
----------------------------------	----------------------	---------------------------------	----------------------

Table 6. Profile configuration attributes for scenario 1

Profile configuration	Start time offset	Constant (0)
	Duration	End of profile
	Start time (seconds)	Uniform (100,300)
	Duration	End of simulation

A set of simulations were done for each protocol by various number of nodes. The results were obtained in the form of graphs and all graphs were displayed as sample mean of 5 runs. Simulation results for different performance metrics are shown below:

Table 7. Simulation results of average end-to-end delay in msec with file size 512 bytes and maximum node speed 5 m/s

Protocol	Number of nodes				
	20	40	60	80	100
AODV	0.17	0.32	0.45	0.61	0.85
OLSR	0.22	0.27	0.33	0.39	0.45
TORA	3.55	31.21	275.82	22724.21	36144.81

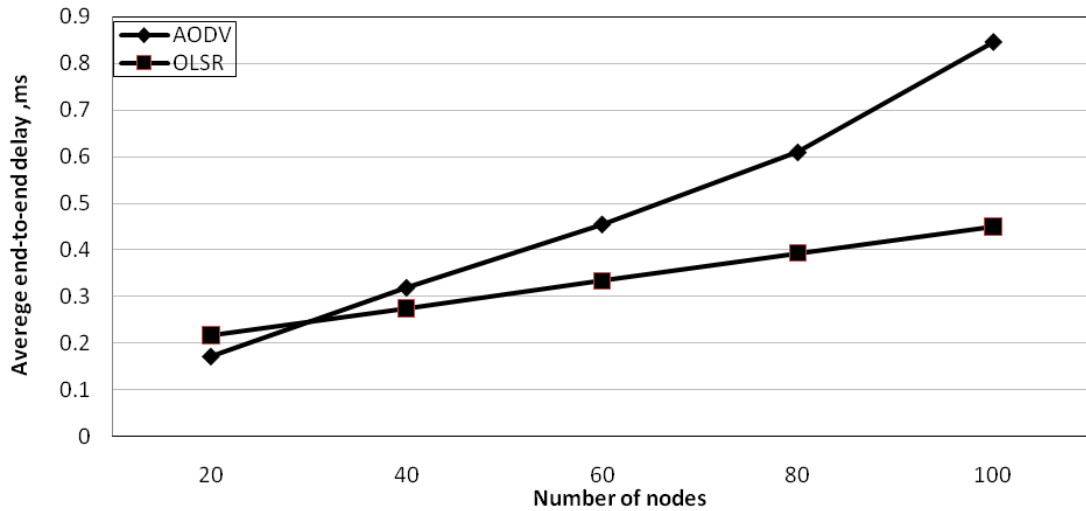


Figure 21. Average end-to-end delay versus number of nodes with file size 512 bytes and maximum node speed 5 m/s

It should be noted from Table 7 that, starting from 20 nodes TORA protocol has too much end-to-end delay so TORA result are not shown in Figure 20. In order to investigate the behavior of TORA in more detail a series of simulations were done with 5, 10 and 15 nodes. The results are shown below:

Table 8. Simulation results of average end-to-end delay in msec with file size 512 bytes and maximum node speed 5 m/s with TORA protocol

Protocol	Number of nodes				
	5	10	15	20	40
TORA	0.68	1.36	2.41	3.55	31.21

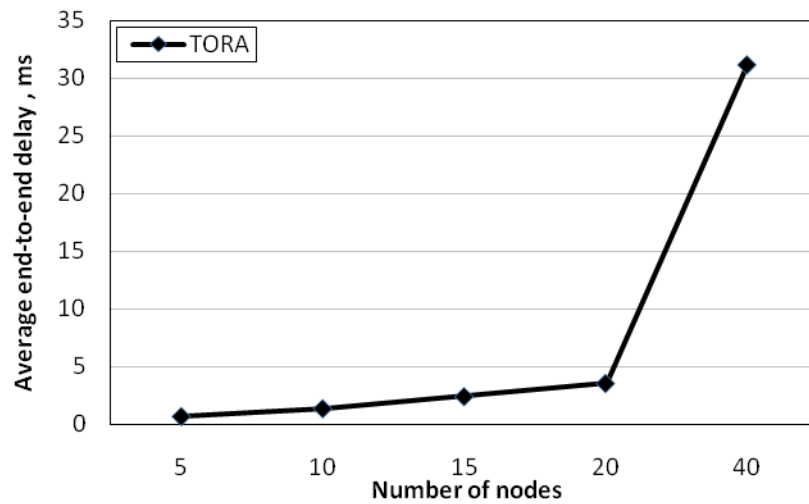


Figure 22. Average end-to-end delay versus number of nodes with file size 512 bytes and maximum node speed 5 m/s with TORA protocol

Table 9. Simulation results of average network load in Kbits/sec with file size 512 bytes and maximum node speed 5 m/s

Protocol	Number of nodes				
	20	40	60	80	100
AODV	2.10	6.64	11.66	18.86	27.31
OLSR	11.20	34.71	71.19	119.72	179.87
TORA	11.66	225.52	226.17	359.98	386.33

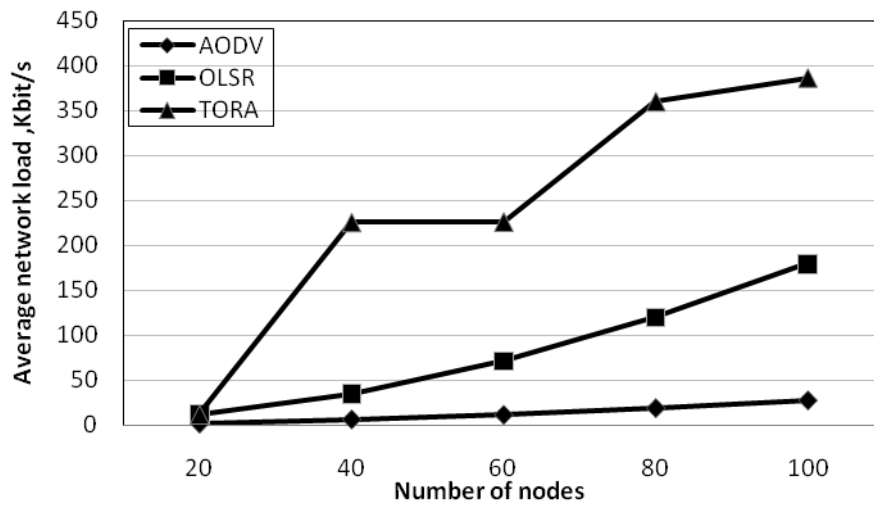


Figure 23. Average network load versus number of nodes with file size 512 bytes and maximum node speed 5 m/s

Table 10. Simulation results of average throughput in Kbits/s with file size 512 bytes and maximum node speed 5 m/s

Protocol	Number of nodes				
	20	40	60	80	100
AODV	24.02	183.25	463.39	971.74	1603.08
OLSR	196.68	1289.70	4040.24	9136.39	17023.76
TORA	21.38	486.05	561.31	698.80	770.38

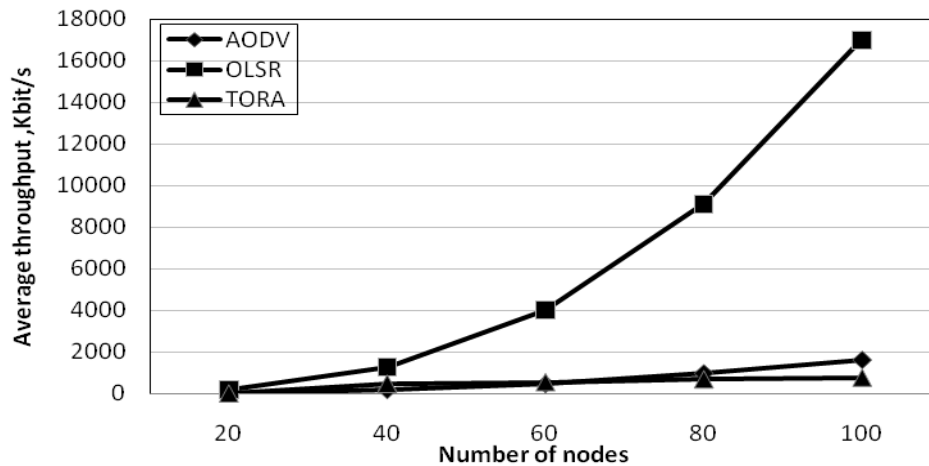


Figure 24. Average throughput versus number of nodes with file size 512 bytes and maximum node speed 5 m/s

Reactive protocols have much end-to-end delay due to broadcasting the routing request by source nodes for whole network and keep them waiting for responses. As it is shown in Figures 20, 21 and Tables 7, 8; TORA protocol has the highest average end-to-end delay. TORA does not use shortest path theory. When the number of mobile nodes increases, the data should pass from many mobile nodes until it reaches to the exact destination. Apart from that, it increases the average end-to-end delay and makes it immoderate in TORA. TORA has the highest delay as compared to OLSR and AODV which is shown in the simulation results.

AODV is always searching about new routes when it needs (on demand method), thus it doesn't save whole routes in the network and also unable to preserve the unused routes in the network. The benefit of this strategy is low controlled traffic. However, overall average end-to-end delay increases in network because the files are waiting in buffer, up to they will be sent by new routes. In addition, AODV maintains only one route per destination in its routing table.

OLSR protocol has the lowest end-to-end delay because of several reasons; using low latency of route discovery process, keeping whole neighbor tables and maintaining track of other nodes available through of them, and not showing the failure link until associated MPR transfer its topology information to other nodes across the network. Stands to these reasons OLSR works efficiently when the number of nodes increases. OLSR protocol maintains and updates routing tables regularly so; it is efficient and has low latency. As a result, OLSR has the lowest end-to-end delay among the three routing protocols.

TORA achieved the highest network load, as it is shown in Table 9 and Figure 22; when the number of nodes increases the network load become worse. TORA performance depends on the number of activated nodes where they were activating at initial setup. In TORA, every intermediate node sends route request reply to the source node so control overhead increases due to the multiple route replies to single route request packets. Moreover, because of the lack of multiple paths to use as alternative routes for the traffic, a route error message will propagate to all its neighbors when a single node in the

path fails. This initiates route rediscovery process, consequently increases the network load.

AODV protocol does not maintain any cache routes. When network topology changes in AODV, it sets up new routes according to requests. This will help AODV protocol to avoid loss of files and make average network load low (Comparing with OLSR and TORA).

Since OLSR protocol always maintains and updates its routing table (proactive method); it helps the OLSR protocol to follow its routing traffic to the destination although there is increase in network load.

In OLSR due to the advantage of MPR in enabling forwarding of the control messages to other nodes, the network load gets minimized and throughput gets maximized.

4.3.2 Investigation of Different File Sizes

In the second set of simulations numbers of nodes were fixed with 40 and 100 where file size was changed as 1024, 2048 and 4096 bytes. All other parameters remained the same as the first scenario. Table 11 presents scenario attributes.

Table 11. General attribute for scenario 2

Attributes	Value
Number of nodes	40, 100
File(data) size	512, 1024, 2048, 4096 bytes
Protocols	AODV, OLSR, TORA
Simulation run time	300 seconds
Simulation area	1000 m * 1000 m

Table 12. Simulation results of average end-to-end delay in msec with 100 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	0.85	0.81	0.79	0.56
OLSR	0.45	0.45	0.45	0.46
TORA	36144.81	36144.81	36144.81	36144.81

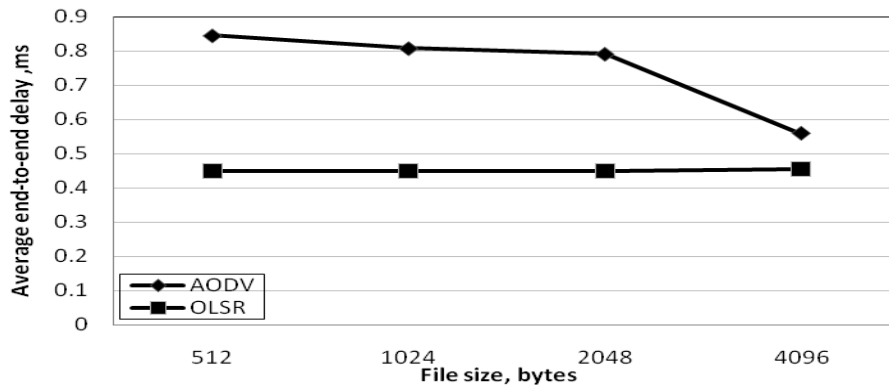


Figure 25. Average end-to-end delay versus different file size with 100 nodes and maximum node speed 5 m/s

It should be pointed out here that since TORA protocol has high end-to-end delay its results are not shown in the figure.

Table 13. Simulation results of average end-to-end delay in msec with 40 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	0.32	0.34	0.35	0.22
OLSR	0.27	0.27	0.28	0.28
TORA	31.21	26.54	24.46	31.17

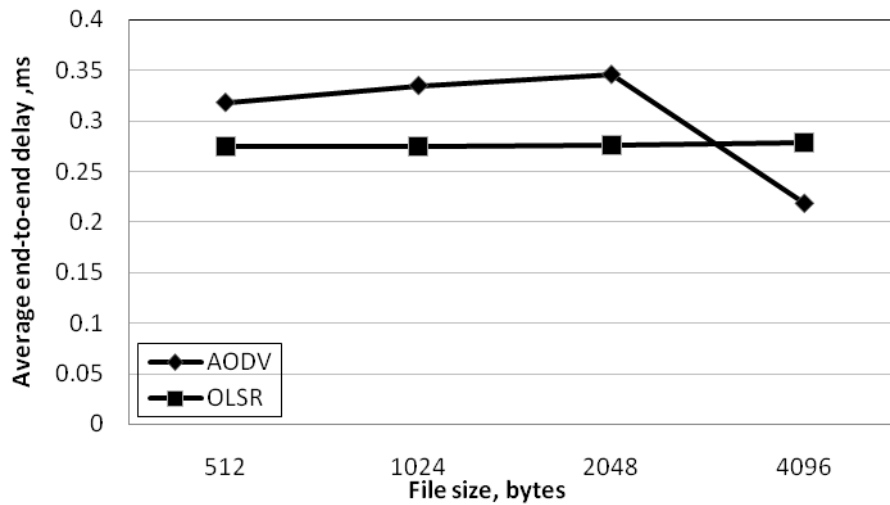


Figure 26. Average end-to-end delay versus different file size with 40 nodes and maximum node speed 5 m/s

Table 14. Simulation results of average network load in Kbits/s with 40 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	6.64	7.43	8.74	10.50
OLSR	34.71	35.36	36.77	39.44
TORA	225.52	200.89	186.95	225.38

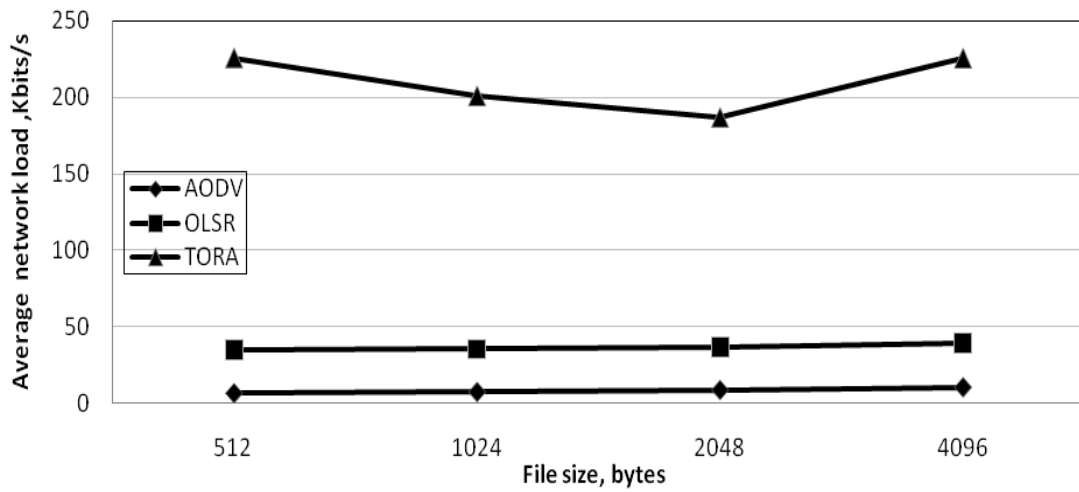


Figure 27. Average network load versus different file size with 40 nodes and maximum node speed 5 m/s

Table 15. Simulation results of average network load in Kbits/s with 100 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	27.31	29.04	30.40	34.58
OLSR	179.87	181.31	184.20	190.79
TORA	386.33	386.33	386.33	386.33

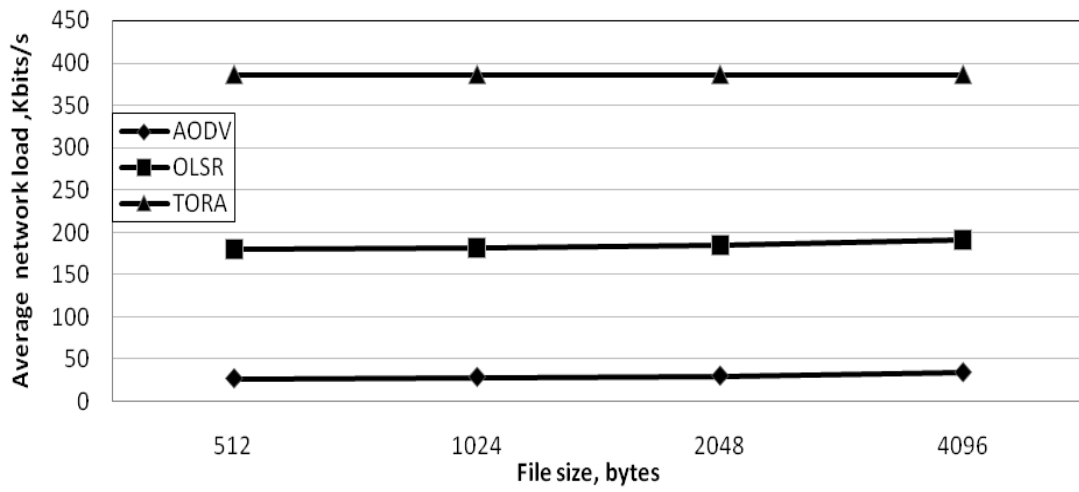


Figure 28. Average network load versus different file size with 100 nodes and maximum node speed 5 m/s

Table 16. Simulation results of average throughput in Kbits/s with 40 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	183.25	186.93	189.19	192.89
OLSR	1289.70	1291.64	1293.59	1293.74
TORA	770.38	770.38	770.38	770.38

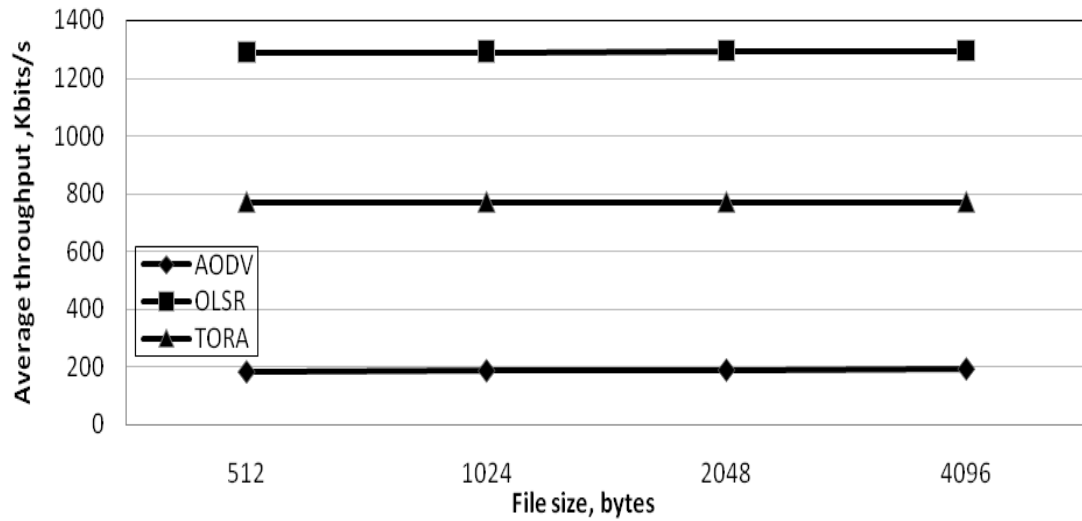


Figure 29. Average throughput versus different file size with 40 nodes and maximum node speed 5 m/s

Table 17. Simulation results of average throughput in Kbits/s with 100 nodes and maximum node speed 5 m/s

Protocol	File size, bytes			
	512	1024	2048	4096
AODV	1603.08	1609.12	1555.19	1576.05
OLSR	17023.76	17013.43	16997.27	17002.14
TORA	766.38	766.38	766.38	766.38

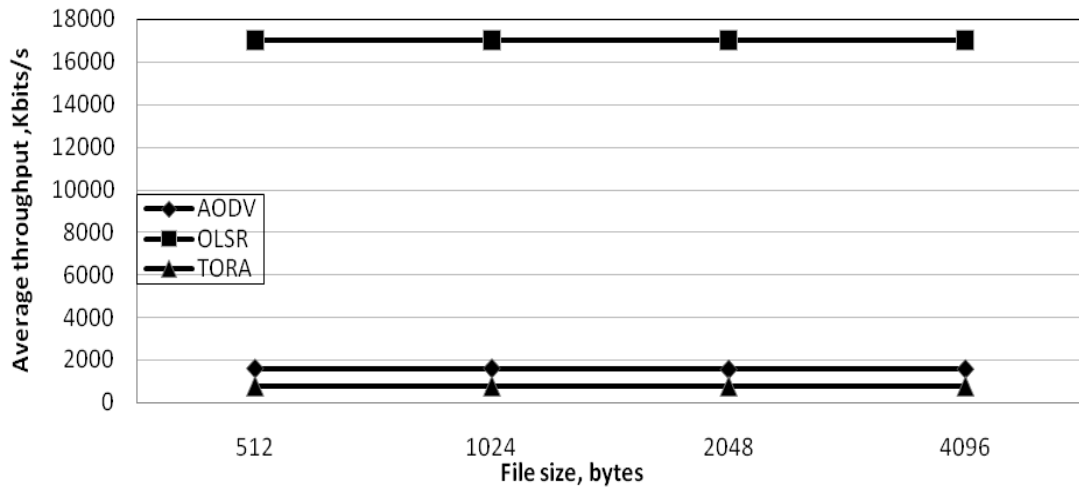


Figure 30. Average throughput versus different file size with 100 nodes and maximum node speed 5 m/s

In MANETs there may be different varying condition problems such as congestion, hidden terminal and network degradation. These problems become more effective when the numbers of traffic sources is increased. Hence makes delay become an important factor determining in the network.

Figures 24, 25 and also Tables 12 and 13 for average end-to-end delay reveal that, Normally in the AODV; there are not many packets in the buffer that should wait for the transmission on the route but the loss rate of the packet are increase with the increase of

file size because they were sent on the old routes and it need more time to send the file with large size. Thus AODV requires periodic update of information but exhibit reasonable average end-to-end delay. In this figure due to AODV characteristic (used hop-by-hop routing mechanism and eliminates the source routing overhead in the network) when the file size increases the average-end-to-end delay will be decreased. Resulting show this affect more when the file size become more. OLSR achieves shorter delays when it is corresponded with AODV since it is a proactive routing protocol where each node maintains a routing table with possible destinations and the number of hops to each destination. When a packet arrives at a node; it is either forwarded immediately or dropped off.

Figures 26, 27 also Tables 14 and 15 presents the average network load for protocols. In case of topological changes, TORA performs updating path information and route establishment that increases average network load and decrease throughput in TORA when compared to other protocols.

Figures 28, 29 also Tables 16 and 17 for average throughput reveal that, among three proposed existing routing protocols in shown that, OLSR protocol is the most effective one. In OLSR with the help of MPR there is continues maintaining information and updating routing, as result reduction of routing overhead. This makes OLSR protocol independent in the network traffic in receiving more data packets.

AODV is admirable, when the goal is to achieve more throughputs regardless of the incremental file size. AODV was used hop-by-hop routing mechanism and eliminates

the source routing overhead in the network. Besides of that, the availability of multiple route information in AODV makes it easy to produce the higher amount of throughput in the network.

From Tables 12-17 and Figure 24-29, it is observed that changing the file size is slightly effect the metrics in OLSR and AODV protocols. Also it is shown that there is almost no effect in the TORA protocol.

For clarity of second scenario in Figures 29-36 and Tables 18-20, the results of 40 nodes and 100 nodes were compared with different performance metrics. The differences between them are illustrated below:

Table 18. Simulation results of average end-to-end delay in msec with 40 and 100 nodes and maximum 5 m/s node speed

No of nodes	Protocol	File size, bytes			
		512	1024	2048	4096
40	AODV	0.32	0.34	0.35	0.22
	OLSR	0.27	0.27	0.28	0.28
	TORA	31.21	26.54	24.46	31.17
100	AODV	0.85	0.81	0.79	0.56
	OLSR	0.45	0.45	0.45	0.46
	TORA	36144.81	36144.81	36144.81	36144.81

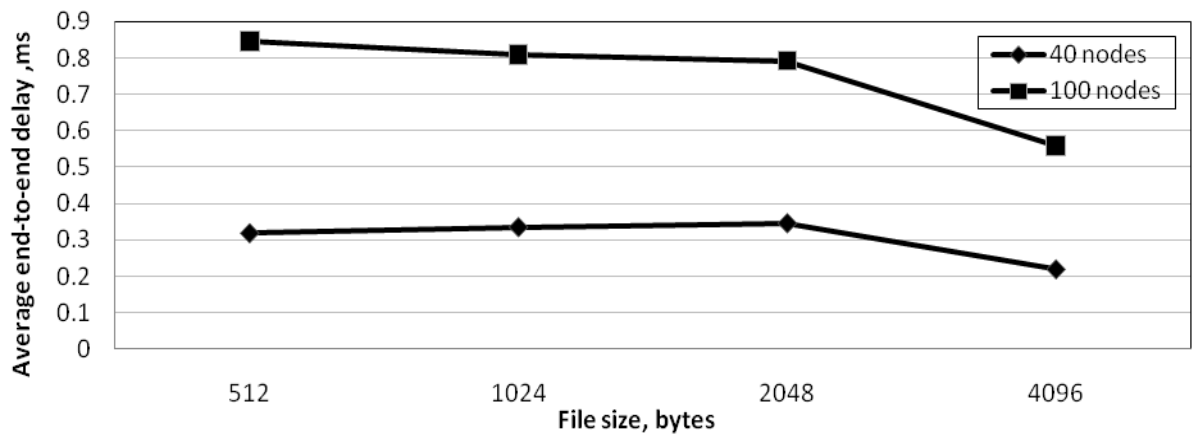


Figure 31. Average end-to-end delay versus file size for AODV with maximum 5 m/s node speed

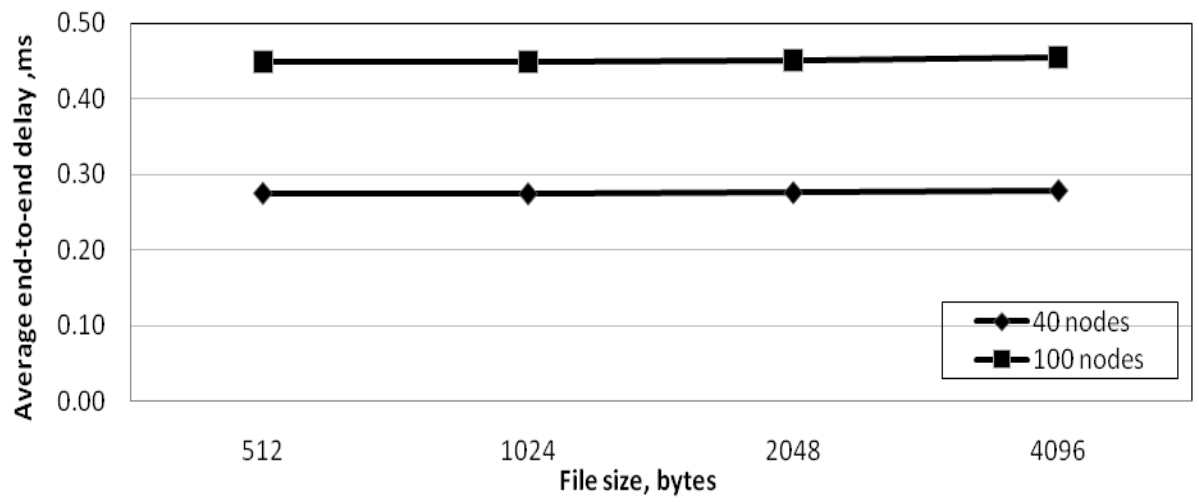


Figure 32. Average end-to-end delay versus file size for OLSR with maximum 5 m/s node speed

Table 19. Simulation results of average network load in Kbits/s with 40 and 100 nodes and maximum 5 m/s node speed

No of nodes	Protocol	File size, bytes			
		512	1024	2048	4096
40	AODV	6.64	7.43	8.74	10.50
	OLSR	34.71	35.36	36.77	39.44
	TORA	225.52	200.89	186.95	225.38
100	AODV	27.31	29.04	30.40	34.58
	OLSR	179.87	181.31	184.20	190.79
	TORA	386.33	386.33	386.33	386.33

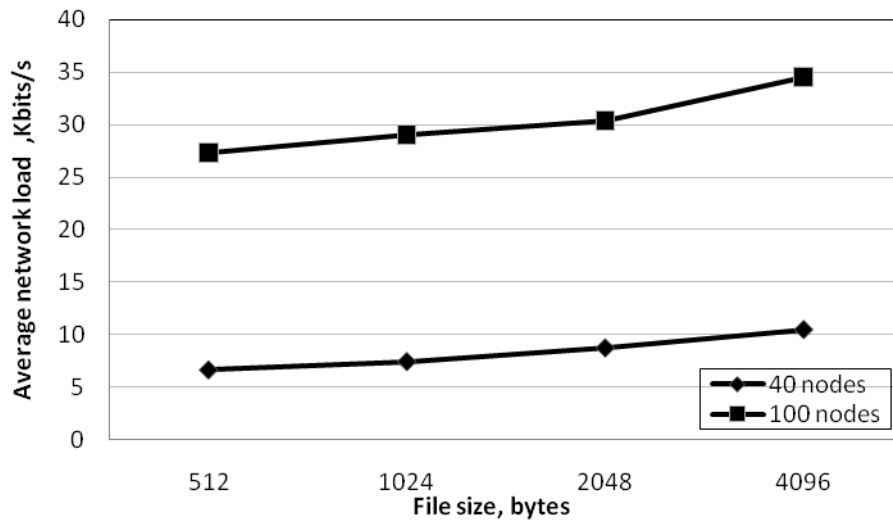


Figure 33. Average network load versus file size for AODV with maximum 5 m/s node speed

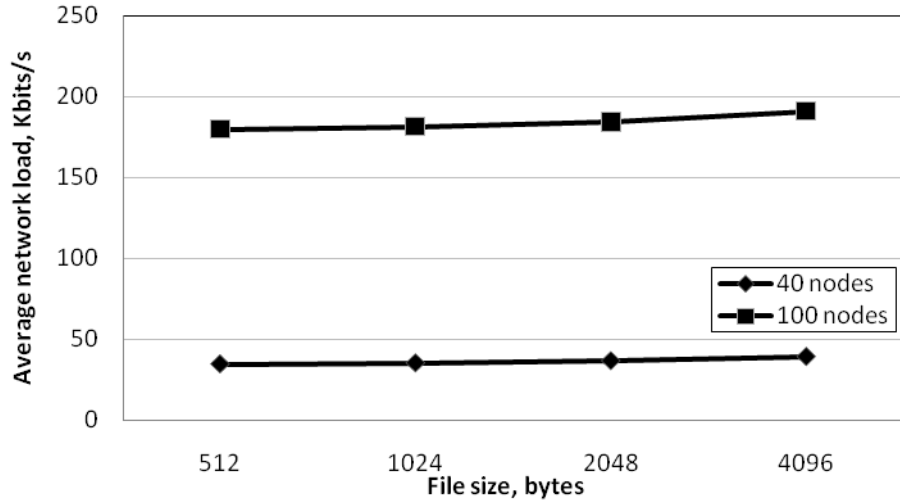


Figure 34. Average network load versus file size for OLSR with maximum 5 m/s node speed

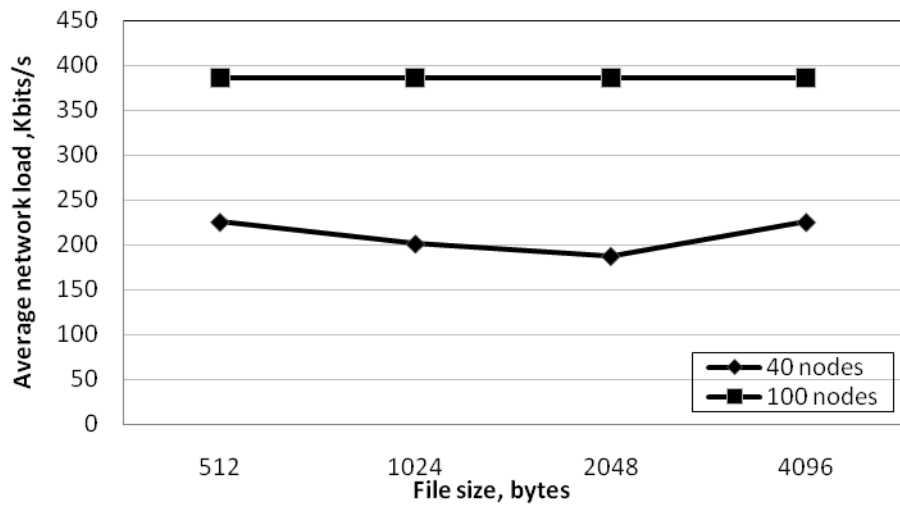


Figure 35. Average network load versus file size for TORA with maximum 5 m/s node speed

Table 20. Simulation results of average throughput in Kbits/s with 40 and 100 nodes with maximum 5 m/s node speed

No of nodes	Protocol	File size, bytes			
		512	1024	2048	4096
40	AODV	183.25	186.93	189.19	192.89
	OLSR	1289.70	1291.64	1293.59	1293.74
	TORA	486.05	429.11	396.57	481.56
100	AODV	1603.08	1609.12	1555.19	1576.05
	OLSR	17023.76	17013.43	16997.27	17002.14
	TORA	770.38	770.38	770.38	770.38

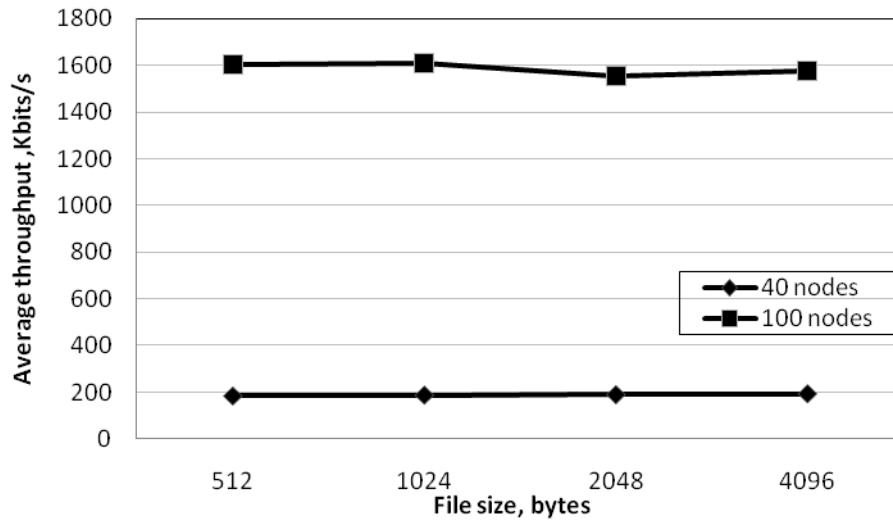


Figure 36. Average throughput versus file size for AODV with maximum 5 m/s node speed

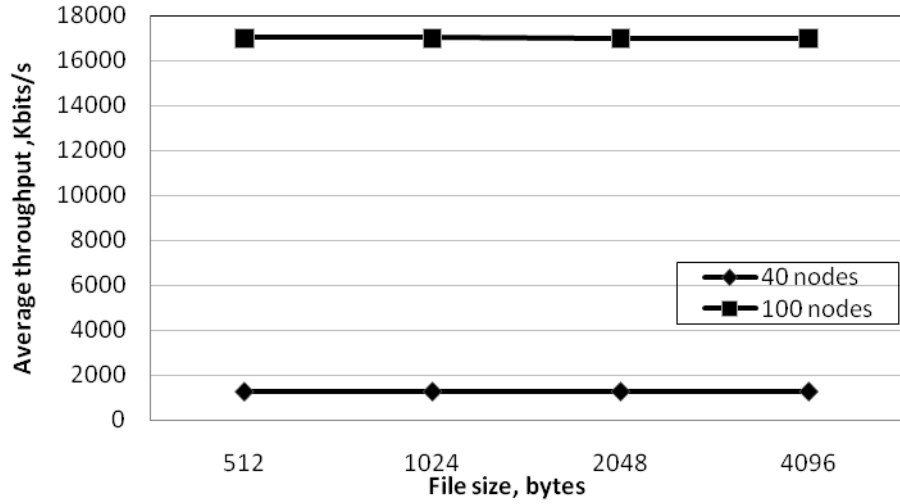


Figure 37. Average throughput versus file size for OLSR with maximum 5 m/s node speed

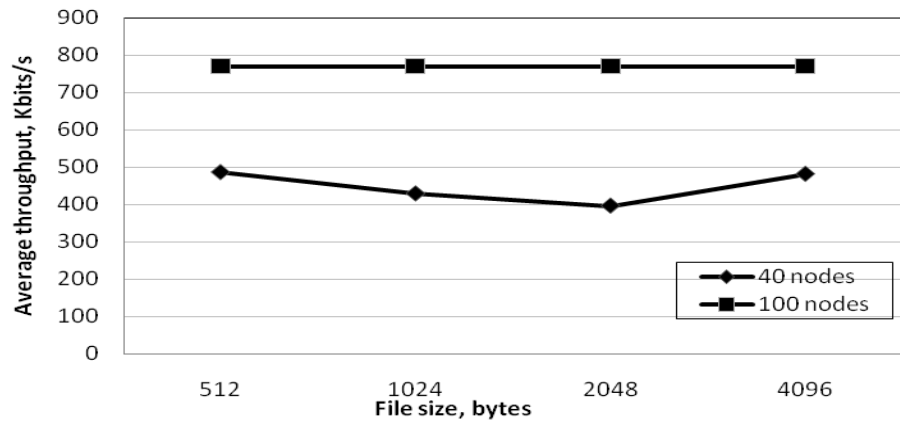


Figure 38. Average throughput versus file size for TORA with maximum 5 m/s node speed

It should be pointed here when the numbers of nodes changed from 40 to 100 nodes, all the results are increased.

4.3.3 Investigation of Different Node Speeds

In this set of simulations, the effect of different node speeds (5 m/s, 30m/s and 50 m/s) to routing protocols with fix number of nodes (100) was observed. All of the remaining parameters are the same as the previous scenario.

Table 21. AODV performance results for 100 nodes with different speeds and file sizes

Performance metrics	Speed (m/s)	File size, byte			
		512	1024	2048	4096
Average end-to-end delay, ms	5	0.85	0.81	0.79	0.56
	30	0.58	0.64	0.78	0.78
	50	0.51	0.40	0.39	0.60
Average network load, Kbits/s	5	27.31	29.04	30.40	34.58
	30	25.14	28.91	32.21	38.73
	50	24.64	26.98	29.91	37.58
Average network throughput, Kbits/s	5	1603.08	1609.12	1555.19	1576.05
	30	1631.12	1699.00	1619.32	1639.46
	50	1658.37	1795.37	1804.02	1693.97

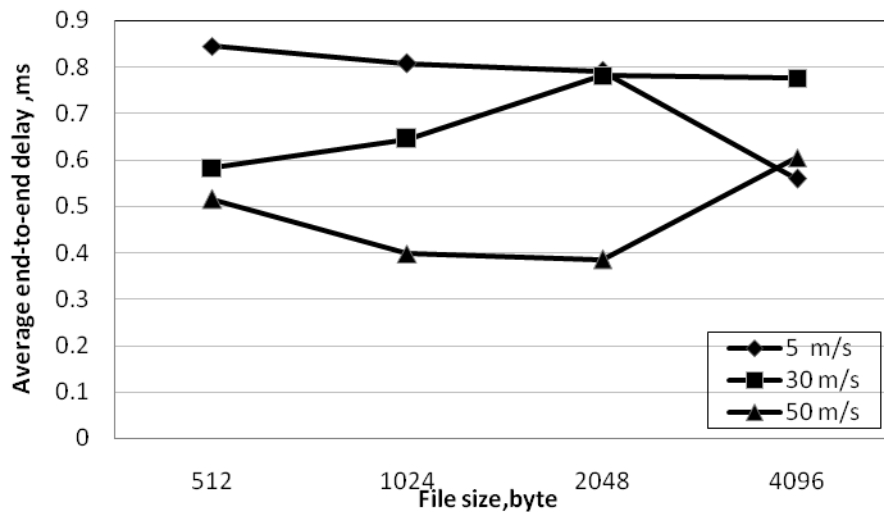


Figure 39. Average end-to-end delay versus file size with 100 nodes for AODV protocol with different node speeds

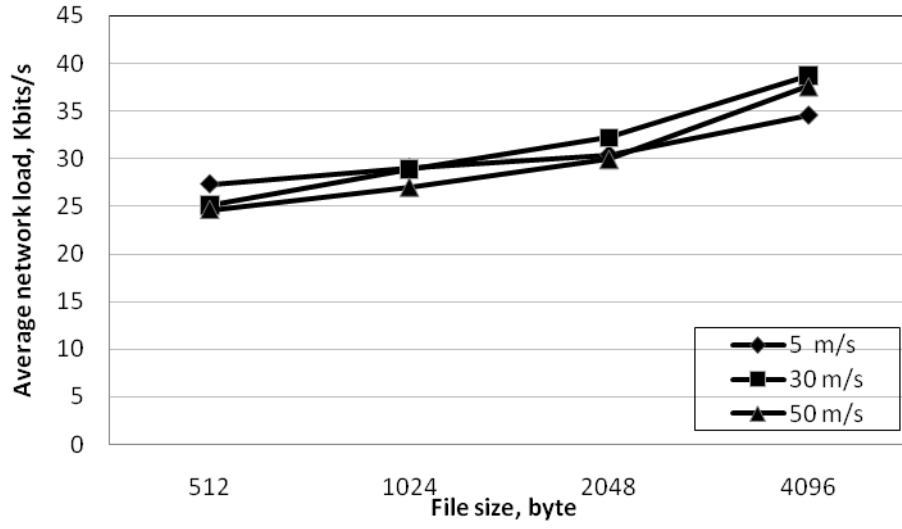


Figure 40. Average network load versus file size with 100 nodes for AODV protocol with different node speeds

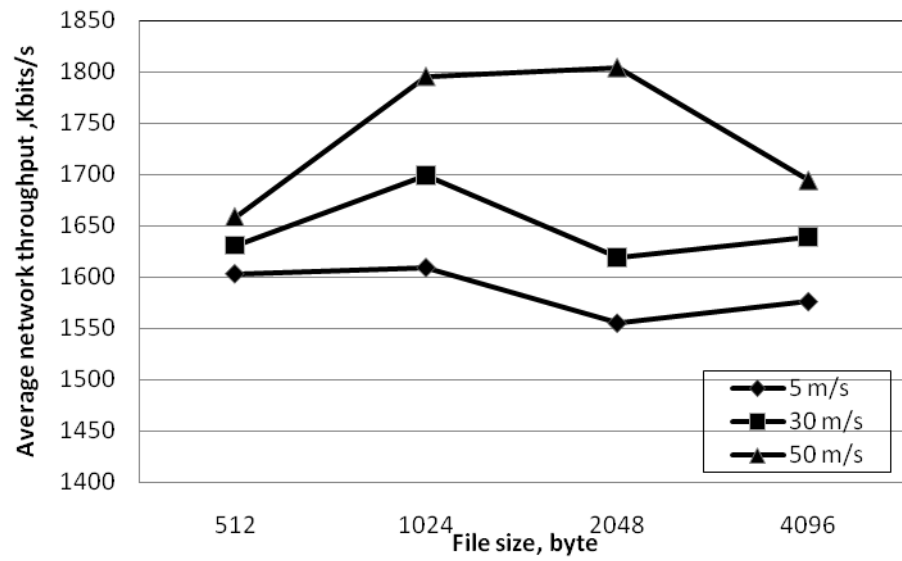


Figure 41. Average throughput versus file size with 100 nodes for AODV protocol with different node speeds

Nodes speed is played a high role in determining the performance metrics of routing protocols. It should be noted that, when the nodes speed increases, more packets are dropped due to unavailable routes.

Table 21 and Figures 38 and 39 are shown with the incidence of increased rate of mobility. The performance of AODV is found to be increased as the network topology stays constant for a low speed network with the lower mobility rate. Even when the speed increases, AODV is slightly affected. Routing tables are more frequently updated in response to topology changes in the network that is shown in fewer packet drops and less performance degradation.

AODV operates the on-demand routing strategy. It is unable to keep the unused routes in the network. Instead, AODV is always searching about new routes when it needs (on-demand method) thus it doesn't save whole routes in the network also unable to preserve the unused routes in the network. This strategy usually generates less control traffic. However, overall average end-to-end delay increases in network because files are waiting in buffer, up to they will be sent by new routes.

Table 22. OLSR performance results for 100 nodes with different speeds and file sizes

Performance metrics	Speed (m/s)	File size, byte			
		512	1024	2048	4096
Average end-to-end delay, ms	5	0.4496	0.4497	0.4508	0.4552
	30	0.4507	0.4517	0.4501	0.4524
	50	0.4533	0.4544	0.4544	0.4566
Average network load, Kbits/s	5	179.87	181.31	184.20	190.79
	30	180.98	183.08	185.46	191.72
	50	180.31	181.89	184.46	191.26
Average network throughput, Kbits/s	5	17023.76	17013.43	16997.27	17002.14
	30	17410.80	17473.68	17426.61	17423.62
	50	17392.23	17383.44	17369.93	17367.78

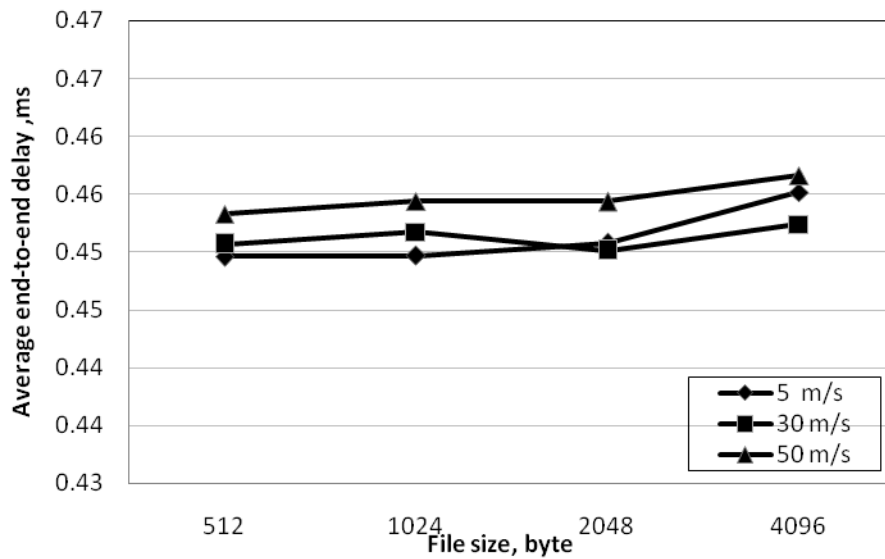


Figure 42. Average end-to-end delay versus file size with 100 nodes for OLSR protocol with different node speeds

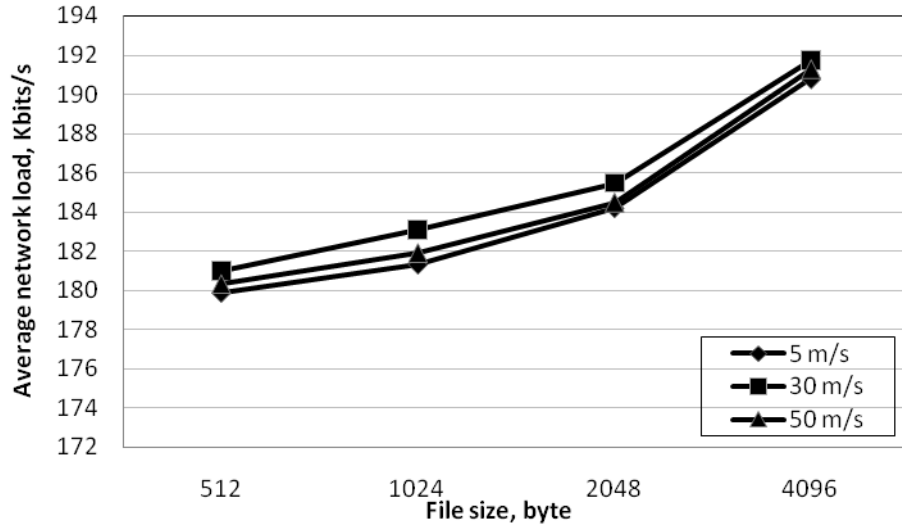


Figure 43. Average network load versus file size with 100 nodes for OLSR protocol with different node speeds

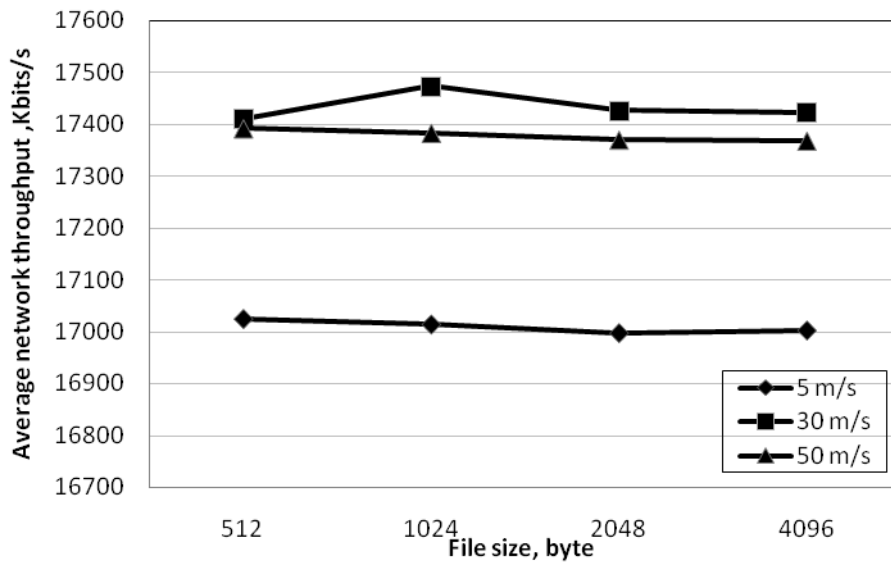


Figure 44. Average throughput versus file size with 100 nodes for OLSR protocol with different node speeds

Form Table 22 and Figures 41-43, OLSR protocol to maintain consistent paths, it updates its routing table frequently. Thus mobility of nodes shows less impact over the performance of OLSR protocol. OLSR can detect link failure sooner than AODV and TORA protocols, so fewer packets are dropped when the speed increases. By exchange of periodical routing updates between nodes even in the absence of data, OLSR shows the highest average network throughput.

By considering a pervious description OLSR protocol has the lowest end-to-end delay (due to using low latency of route discovery process, keep whole neighbor tables and maintaining track of other nodes available through of them, and doesn't show the failure link until associated MPR transfer its topology information to other nodes across the network). As a result, it exhibits the lowest end-to-end delay among the three routing protocols, the delay even being found almost insensitive to change in speed.

Table 23. TORA performance results for 100 nodes with different speeds and file sizes

Performance metrics	Speed (m/s)	File size, byte			
		512	1024	2048	4096
Average end-to-end delay, ms	5	36144.81	36144.81	36144.81	36144.81
	30	41687.44	41687.44	41687.44	41687.44
	50	39838.19	39838.19	39838.19	39838.19
Average network load, Kbits/s	5	386.33	386.33	386.33	386.33
	30	387.49	387.49	387.49	387.49
	50	380.99	380.99	380.99	380.99
Average network throughput, Kbits/s	5	770.38	770.38	770.38	770.38
	30	762.02	762.02	762.02	762.02
	50	741.94	741.94	741.94	741.94

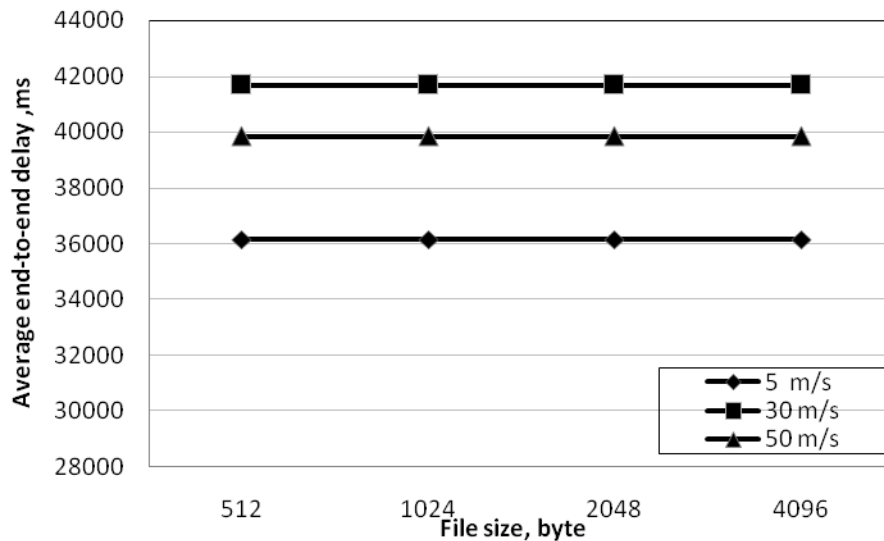


Figure 45. Average end-to-end delay versus file size with 100 nodes for TORA protocol with different node speeds

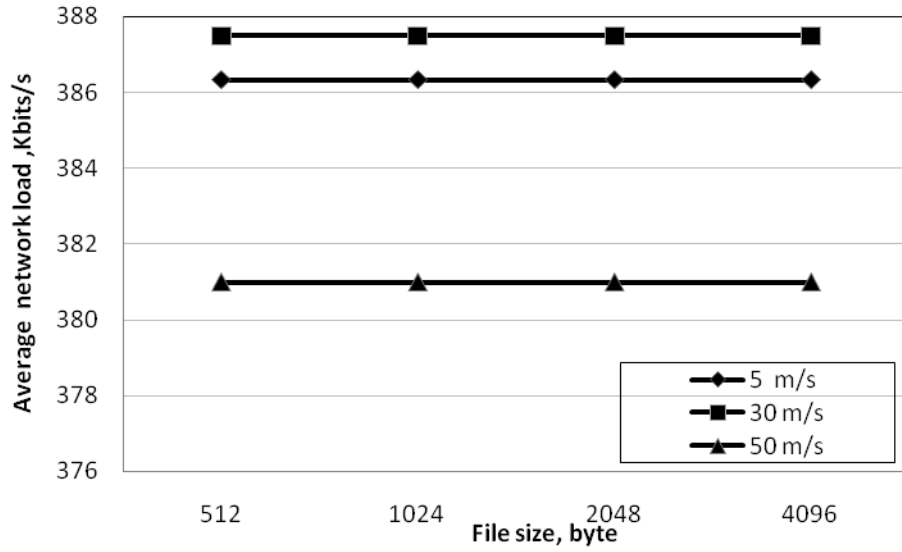


Figure 46. Average network load versus file size with 100 nodes for TORA protocol with different node speeds

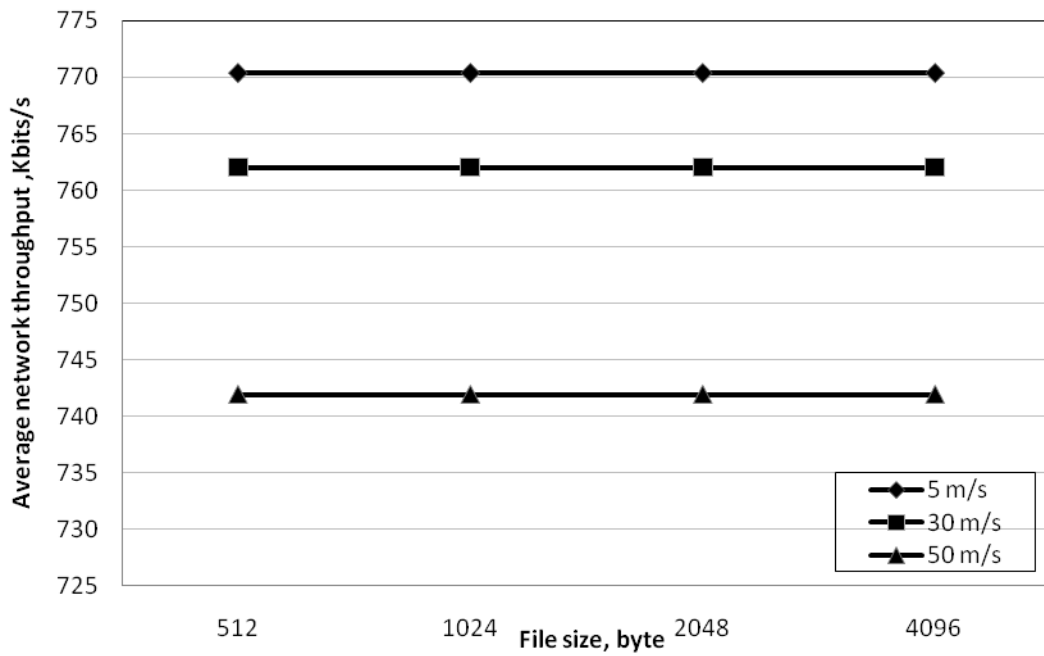


Figure 47. Average throughput versus file size with 100 nodes for TORA protocol with different node speeds

In the reactive routing protocols network layer need to drop more packets while the routing protocol is still computing the route to the destination also there is more possibility of buffer overflow. Due to these attribute poor performances are shown in the TORA protocol in the Table 23 and Figures 44-46.

Due to taking longer time to initial route discovery mechanism in TORA performance might affects in network partition owing to the high mobility. Apart from that, the loss of distance information due to the link failure in a mobility network also makes TORA with poor average end-to-end delay in the network.

Corresponding to high mobility and responding to topological changes, TORA follows an adaptive method which increases the network load and decrease throughputs for updating the path information.

4.4 Simulation Results and Discussions

Analysis for every different parameter produces different results. To find the highest throughput, lowest end-to-end delay and network load between the source and destination nodes some scenarios were done in the previous part of this thesis.

By considering first scenario tables and figures which were fixed 512 byte file size, 5 m/s maximum speed for each nodes and different number of nodes; TORA has shown greater end-to-end delay compared to AODV and OLSR. Experimental result also shows that TORA has lower throughput compared to AODV and OLSR. AODV and OLSR have lowest average end-to-end delay where as in case of TORA, the average of end-to-end delay is significantly high. When the number of mobile nodes increases then the data which is needed to deliver to the specific destination has to pass from many mobiles, so it increases end-to-end delay in TORA and make it excessive and also when the number of nodes with high traffic is increased, the cache of routes make the end-to-end delay gets worse.

In medium traffic environment by notice second scenarios tables and figures which the file size was changed to 1028, 2048 and 4096 byte OLSR shows better throughput than AODV and TORA also the lowest end-to-end delay time. Here for AODV to find an optimal fresh path due to frequent broadcasting of route re-initialization and RRQ message also because of using destination sequence number for every RRQ, they increase the efficiency of the link without needing to execute the large routing table every time.

In high mobility scenarios in the third part, OLSR also shows better throughput than AODV and TORA with different file size and speed. Since OLSR without saving all the nodes parts maintains one hop and two hop neighbors, it becomes more impressive in link update process. In addition, OLSR minimizes the traversal of control message by multipoint relays and decreasing the average end-to-end delay compared to AODV and TORA.

OLSR is well suited for small and large size network with high mobility. It also performs better at low node mobility in large network. AODV performs well in medium sized networks under high traffic load. In respect of average end-to-end delay, average network load time and average throughput, OLSR has shown better performance than AODV and TORA.

In TORA with the increasing number of nodes and speed of them, throughput is not affected; these were due to maintain cluster of nodes in the topology by dividing them into different node sets.

OLSR exhibited very low end-to-end delay in all scenarios. AODV had an improved end-to-end delay when network grows but when the speed increases it did not have obvious effect on end-to-end delay. It can be concluded that MANET could have dynamic number of nodes connectivity in mobility, in general, when the number of nodes is higher, AODV and TORA would be avoided. With increase in the number of nodes and due to mobility, throughput performance of AODV and TORA are minor

affected. It is important to realize that OLSR has better throughput performance, as it is shown in all figures, comparing to AODV and TORA.

Chapter 5

CONCLUSION

This thesis includes two parts, the survey study and the simulation study. From the first part it is concluded that routing protocols are playing 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 study from proactive category Optimized Link State Routing (OLSR), from reactive category Ad-hoc On-demand Distance Vector (AODV) and Temporary Ordered Routing Algorithm (TORA) are evaluated using OPNET simulator under the medium load traffic size in FTP protocol. TORA can work as reactive and proactive manner but here it is used as reactive protocol.

In this work, a number of simulation experiments are performed by using OPNET (version 17.1) simulator to determine and evaluate the performance of mobile ad hoc networks. Random waypoint mobility model is used as pattern of mobility. As performance metrics average throughput, average network load and average end-to-end

delay are examined in different number of nodes, file sizes and node speeds. In the first part of simulation the number of nodes is varied from 20 to 100 with file size 512 bytes and node speed 5m/s. The file size is changed from 512 bytes to 4096 bytes in the second scenario with the other fixed attributes of the first scenario; and in the last scenario the speed was used as 5 m/s, 30 m/s and 50 m/s with the file size varying from 512 bytes to 4096 bytes using 100 nodes in the network.

According to the simulation results and observations a number of conclusions are drawn as follows. In general, proactive protocols perform better in case of average throughput, average end-to-end delay and average network load. OLSR seems to be well as it exhibits lower end-to-end delay and highest throughput. The OLSR delay has very minor changes when the numbers of nodes increases. On the other hand, between two reactive protocols, AODV and TORA, AODV seems to be more successful than TORA in the performance metrics. However, TORA has lower throughput compared to AODV and OLSR.

The OPNET version 17.1 supports six MANET routing protocols only. It does not support other protocols for instance LDR and ZRP. So, different protocols from different classifications could be implemented in OPNET. In addition to this, suggesting for the future research is to develop a modified version of the selected routing protocols which could consider different aspects of routing protocols such as rate of higher route establishment with lesser route breakage and any weakness of the used protocols could be improvised.

REFERENCES

- [1] X. Zhao, “An Adaptive Approach for Optimized Opportunistic Routing Over Delay Tolerant Mobile Ad Hoc Networks” ,December 2007.
- [2] Idris Skloul Ibrahim, “Ad Hoc Wireless networks Architecture and Protocols,” PhD Research Proposal, HERIOT WATT University, UK.
- [3] K. Gorantala , “Routing Protocols in Mobile Ad-hoc Networks” ,Master’s Thesis, Dept. of Computer Science, Umea University, June 15, 2006.
- [4] M.Ilyas , “The Handbook of Ad Hoc Wireless Networks” 1st edition, Boca Raton, Florida, ISBN 0-8493-1332-5, 2003.
- [5] C. Toh, “Ad Hoc Mobile Wireless Networks: Protocols and Systems” ,1st edition, New Jersey (USA), Prentice Hall, ISBN 0-13-007817-4, 2002.
- [6] K. Karapetsas, ““Building a Simulation Toolkit for Wireless Mesh Clusters and Evaluating the Suitability of Different Families of Ad hoc Protocols for the Tactical Network Topology” March 2005, MONTEREY, CALIFORNIA
- [7] J. Uddi Md, R. Zasad, “Study and Performance Comparison of MANET Routing Protocols: TORA, LDR and ZRP”, Master Thesis Electrical Engineering Emphasis on Telecommunications, Thesis no: MEE-2010-5834, May 2010.

- [8] C. Adjih, M. Pascale, M. Paul, B. Emmanuel and P. Thierry, "QoS Support, Security and OSPF Interconnection in a MANET using OLSR", Ad Hoc Networks Journal, 2008.
- [9] A. Vallejo, G. Corral, J. Abella and A. Zaballos, "Ad hoc Routing Performance Study Using OPNET Modeler" University Ramon Llull, Barcelona ,Spain, 2006.
- [10] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot "Optimized Link State Routing Protocol(OLSR)" RFC 3626 (Experimental), IETF, October 2003.
- [11] S. Murphy, J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks" ACM Mobile Networks and Applications Journal, pp. 183-197, Nov.1996.
- [12] I. Sung Han , H. Bin Ryou, J. Kim, J. Baek Kwon, "A Novel Approach to Search a Node in MANET", Information Science and Security, ICISS, p.p. 44 – 48, 2008.
- [13] S. R. Chaudhry, A. Al-Khwildi, Y. Casey, H. Aldelou "A Performance Comparison of Multi On- Demand Routing in WIREless Ad Hoc Networks" (WNCG), School of Engineering & Design, Brunel University, West London
- [14] C. E. Perkins, E.M. Royer, S.R. Das, "Ad hoc On-demand Distance Vector routing" ,Request For Comments (Proposed Standard) 3561, Internet Engineering Task Force <http://www.ietf.org/rfc/rfc3561.txt?number=3561>, July 2003.

- [15] C. E. Perkins, E.M. Royer, S.R. Das, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", IEEE Personal Communications Magazine, no. 8, pp. 16-28, 2001.
- [16] M. Haas, J. Zygmunt, N. Pearlman, R. Marc, "A New Routing Protocol for The Reconfigurable Wireless Networks", Proc. of 6th IEEE Intl, Conf. on Universal Personal Comm., IEEE ICUPC'97, San Diego, California, USA, 1997.
- [17] S. Ali, A. Ali, "Performance Analysis of AODV, DSR and OLSR in MANET", Master's Thesis, M.10:04, COM/School of Computing, BTH, 2010.
- [18] W. G. LOL, "An Investigation of the Impact of Routing Protocols on MANETs using Simulation Modeling" Master Thesis, School of Computing and Mathematical Science, Auckland university of Technology, 2008.
- [19] M.K. J. Kumar, R.S. Rajesh, "Performance Analysis of MANET Routing Protocols in different Mobility Models" IJCSNS International Journal of Computer Science and Network 22 Security, VOL.9 No.2, February2009.
- [20] N Vetrivelan, A.V. Reddy, "Performance Analysis of Three Routing Protocols for Varying MANET Size" Proceedings of International M. Conference of Eng. & Computer Scientists, Hong Kong, Vol II IMECS 2008.

- [21] A. Shrestha, F. Tekiner, "Investigation of MANET routing protocols for mobility and scalability" International Conference on Parallel and Distributed Computing, Applications and Technologies, Higashi Hiroshima, 2009.
- [22] S. Mittal, P. Kaur, "Performance Comparison of AODV, DSR and ZRP Routing Protocols in MANET'S" Intl. Conf. on Adv. in Comp., Control, and Telecom. Technologies, Trivandrum, Kerala, India, 28-29, December, 2009.
- [23] S. Giannoulis, C. Antonopoulos, E. Topalis, S. Koubias "ZRP versus DSR and TORA: A comprehensive survey on ZRP performance" 10th IEEE Conference Emerging Technologies and Factory Automation, Greece, 2005.
- [24] S. C. Sharma, Kumar Manoj, "Effect of Throughput in MANET for Static and Dynamic Network For IEEE 802.11" , Proceedings of the World Congress on Engineering 2011 Vol II,WCE 2011, July 6 - 8, 2011, London, U.K.
- [25] D.O. Jörg, "Performance Comparison of MANET Routing Protocols In Different Network Sizes" Comp. Science Project, Institute of Comp. Science and Networks and Distributed Sys, University of Berne, Switzerland, 2003.
- [26] C. Yang, L. Tseng "Fisheye Zone Routing Protocol for Mobile Ad-Hoc Networks" Multimedia Communications Laboratory, Second IEEE Consumer Communications and Networking Conference, Taiwan, 2005.

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

[28] A. Zaballos, A. Vallejo, G. Corral, J. Abella, “Ad hoc Routing Performance Study Using OPNET Modeler” , University Ramon Llull (URL - La Salle Engineering) Barcelona (Spain).

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

[30] S. Kaur, N. Bhatia, N. Kapoor, “Simulation Analysis of AODV Routing Protocol of MANET using OPNET” , Dept. of information, S.B.B.S.I.E.T, Jalandhar, Punjab, India,2,3DAV College, Jalandhar, Punjab, India,ISSN : 2229 - 4333(Print) | ISSN : 0976 - 8491 (On line), IJCST Vol. 2, Issue 3, September 2011

[31] J. Khan, Dr.S. I. Hydar, Dr.S. M. Fakar, D. Mustafa, “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, March 2011.

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

[33] V.N. Talooki, K. Ziarati "Performance Comparison of Routing Protocols for Mobile Ad-Hoc Networks" Asia-Pacific conf. on Comm., APCC'06.

[34] S. Ahmed, M. Bilal, U. Farooq, F. Hadi, "Performance Analysis of Various Routing Strategies in Mobile Ad-hoc Network Using QualNet Simulator", Int. Conf. on Emerging Technologies, ICET, Islamabad, 2007.

[35] K. Halgamuge, P. Wang, "Classification and Clustering for Knowledge Discovery", 1st edition. Netherlands, Springer, 2005.

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

[37] S. Bin Abd Latif, M.A. Rashid, F. Alam, "Profiling Delay and Throughput Characteristics of Interactive Multimedia Traffic over WLANs using OPNET", School of Engineering Institute of Technology & Engineering, Massey University, Auckland, New Zealand.

[38] J. Hsu, S. Bhatia, K. Tang, R. Bagrodia, “Performance of Mobile Ad Hoc Networking Routing Protocols in Large Scale Scenarios”, Milcom, IEEE Military Communications Conference, 2004.

[39] OPNET, [Online]. Available: <http://www.opnet.com/>. [Accessed: June. 24, 2009].

APPENDICES

Appendix A: AODV Source Code

```
/******  
  
    aodv.h - description  
    -----  
  
Start      : Tue Jul 1 2003  
  
Refrence   : (C) 2003 by Luke Klein-Berndt  
  
Email Address : kleinb@nist.gov  
  
*****/
```

```

#ifndef AODV_H

#define AODV_H

#include <linux/netdevice.h>

#define AODVPORT      654

#define TRUE          1
#define FALSE         0

// Notice Part 10 of AODV draft

// Milliseconds is assuming

#define ACTIVE_ROUTE_TIMEOUT  3000

#define ALLOWED_HELLO_LOSS    2

#define BLACKLIST_TIMEOUT     RREQ_RETRIES * NET_TRAVERSAL_TIME

#define DELETE_PERIOD         ALLOWED_HELLO_LOSS * HELLO_INTERVAL

#define HELLO_INTERVAL        1000

#define LOCAL_ADD_TTL         2

#define MAX_REPAIR_TTL        0.3 * NET_DIAMETER

#define MY_ROUTE_TIMEOUT      ACTIVE_ROUTE_TIMEOUT

#define NET_DIAMETER          10

#define NODE_TRAVERSAL_TIME    40

#define NET_TRAVERSAL_TIME     2 * NODE_TRAVERSAL_TIME * NET_DIAMETER

#define NEXT_HOP_WAIT          NODE_TRAVERSAL_TIME + 10

#define PATH_DISCOVERY_TIME    2 * NET_TRAVERSAL_TIME

```

```
#define RERR_RATELIMIT    10

#define RING_TRAVERSAL_TIME 2 * NODE_TRAVERSAL_TIME * (TTL_VALUE + TIMEOUT_BUFFER)

#define RREQ_RETRIES      2

#define RREQ_RATELIMIT    10

#define TIMEOUT_BUFFER    2

#define TTL_START         2

#define TTL_INCREMENT     2

#define TTL_THRESHOLD     7

#define TTL_VALUE         3

// Message Types

#define RREQ_MESSAGE      1

#define RREP_MESSAGE      2

#define RERR_MESSAGE      3

#define RREP_ACK_MESSAGE  4

//Tasks

#define TASK_RREQ         1

#define TASK_RREP         2

#define TASK_RERR         3

#define TASK_RREP_ACK     4

#define TASK_RESEND_RREQ  101
```

```
#define TASK_HELLO          102

#define TASK_NEIGHBOR    103

#define TASK_CLEANUP      104

#define TASK_ROUTE_CLEANUP 105

// Structures

// Route table

struct _flood_id {

    u_int32_t src_ip;

    u_int32_t dst_ip;

    u_int32_t id;

    u_int64_t lifetime;

    struct _flood_id *next;

};

typedef struct _flood_id flood_id;

struct _aadv_route {

    u_int32_t ip;

    u_int32_t netmask;

    u_int32_t seq;

    u_int32_t old_seq;

    u_int8_t metric;
```

```

    u_int32_t next_hop;
    u_int32_t rreq_id;
    u_int64_t lifetime;
    struct net_device *dev;
    u_int8_t route_valid:1;
    u_int8_t route_seq_valid:1;
    u_int8_t self_route:1;
    struct _aadv_route *next;
    struct _aadv_route *prev;
};

typedef struct _aadv_route aadv_route;

struct _aadv_dev {
    struct net_device *dev;
    aadv_route *route_entry;
    int index;
    u_int32_t ip;
    u_int32_t netmask;
    char name[IFNAMSIZ];
    struct _aadv_dev *next;
    struct socket *sock;
};

```

```

};

typedef struct _aadv_dev aadv_dev;

struct _aadv_neigh {

    u_int32_t ip;

    u_int32_t seq;

    u_int64_t lifetime;

    unsigned char hw_addr[ETH_ALEN];

    struct net_device *dev;

    aadv_route *route_entry;

    int link;

    u_int8_t valid_link;

    struct _aadv_neigh *next;

};

typedef struct _aadv_neigh aadv_neigh;

struct _task {

    int type;

    u_int32_t id;

    u_int64_t time;

    u_int32_t dst_ip;

    u_int32_t src_ip;

```

```

struct net_device *dev;

u_int8_t ttl;

u_int16_t retries;

unsigned char src_hw_addr[ETH_ALEN];

unsigned int data_len;

void *data;

struct _task *next;

struct _task *prev;

};

typedef struct _task task;

//Route reply message type

typedef struct {

    u_int8_t type;

} rrep_ack;

typedef struct {

    u_int8_t type;

#ifdef __BIG_ENDIAN_BITFIELD

    unsigned int a:1;

    unsigned int reserved1:7;

#elif defined(__LITTLE_ENDIAN_BITFIELD)

```



```

    unsigned int reserved1:7;

    unsigned int a:1;

#else
#error "Please fix <asm/byteorder.h>"
#endif

        u_int8_t reserved2;

    u_int8_t metric;

    u_int32_t dst_ip;

    u_int32_t dst_seq;

    u_int32_t src_ip;

    u_int32_t lifetime;

}rrep;

//Endian handling based on DSR implemetation by Alex Song s369677@student.uq.edu.au

typedef struct {

    u_int8_t type;

#ifdef(__BIG_ENDIAN_BITFIELD)

    u_int8_t j:1;

    u_int8_t r:1;

    u_int8_t g:1;

    u_int8_t d:1;

```

```
    u_int8_t u:1;
    u_int8_t reserved:3;
#elif defined(__LITTLE_ENDIAN_BITFIELD)
    u_int8_t reserved:3;
    u_int8_t u:1;
    u_int8_t d:1;
    u_int8_t g:1;
    u_int8_t r:1;
    u_int8_t j:1;
#else
#error "Please fix <asm/byteorder.h>"
#endif

    u_int8_t second_reserved;
    u_int8_t metric;
    u_int32_t rreq_id;
    u_int32_t dst_ip;
    u_int32_t dst_seq;
    u_int32_t src_ip;
    u_int32_t src_seq;
}rreq;
```

```

typedef struct {
    u_int8_t type;
#ifdef __BIG_ENDIAN_BITFIELD
    unsigned int n:1;
    unsigned int reserved:15;
#elif defined(__LITTLE_ENDIAN_BITFIELD)
    unsigned int reserved:15;
    unsigned int n:1;
#else
#error "Please fix <asm/byteorder.h>"
#endif
    unsigned int dst_count:8;
} rerr;

typedef struct {
    u_int32_t ip;
    u_int32_t seq;
} aadv_dst;

struct _rerr_route {
    u_int32_t ip;
    u_int32_t seq;
    struct _rerr_route *next;
};

typedef struct _rerr_route rerr_route;
#endif

```

Appendix B: Step by Step Configuration of Simulation

In this appendix explanation and providential procedure which are used in the final thesis is included. There are three classes for the simulations; each division has three simulations, one for each AODV, OLSR and TORA as protocols of Ad hoc networks.

A: Evaluation Platform:

The evaluation platform was OPNET (Simulator 17.1). The procedures to create a new project are:

1. Go to start menu, click on the Visual Studio (to open), as shown in Figure 1.
2. Go to OPNET directory, and then run Modeler.Exe, as shown in Figure 2.
3. After reading the agreement of OPNET accept it (Figure 3) so that OPNET window appears.

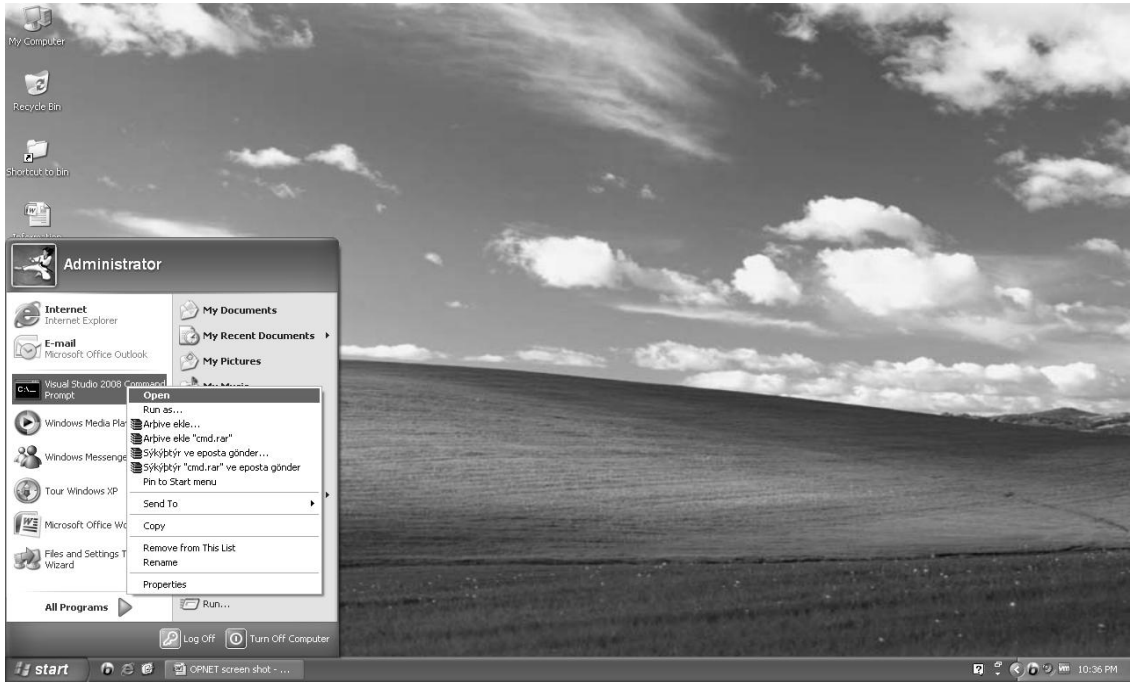
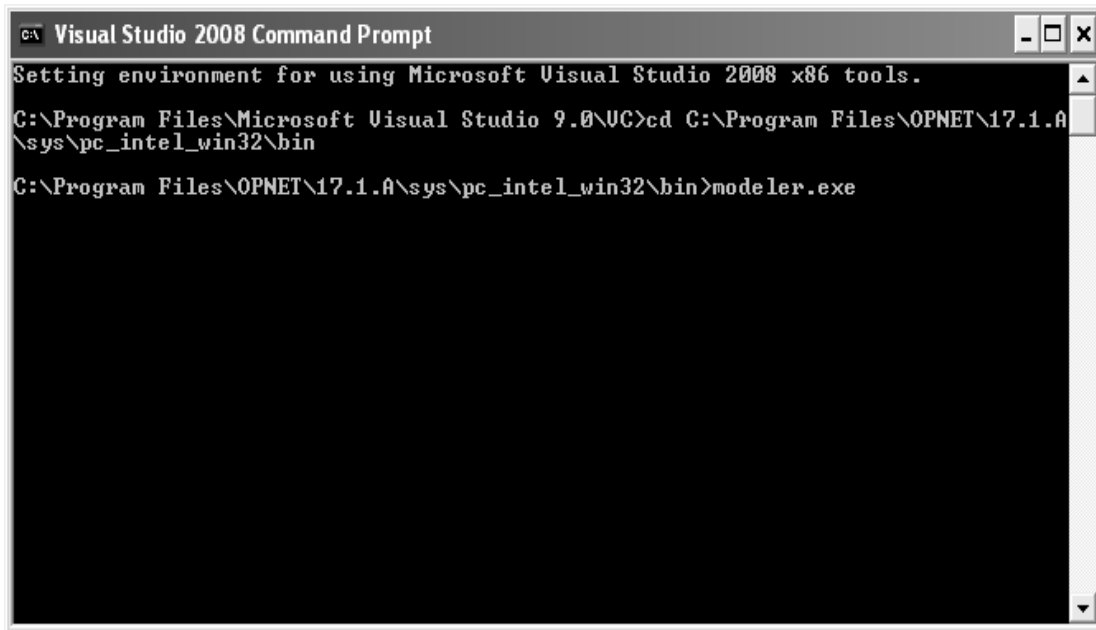


Figure 1



```
Visual Studio 2008 Command Prompt
Setting environment for using Microsoft Visual Studio 2008 x86 tools.
C:\Program Files\Microsoft Visual Studio 9.0\VC>cd C:\Program Files\OPNET\17.1.A
\sys\pc_intel_win32\bin
C:\Program Files\OPNET\17.1.A\sys\pc_intel_win32\bin>modeler.exe
```

Figure 2

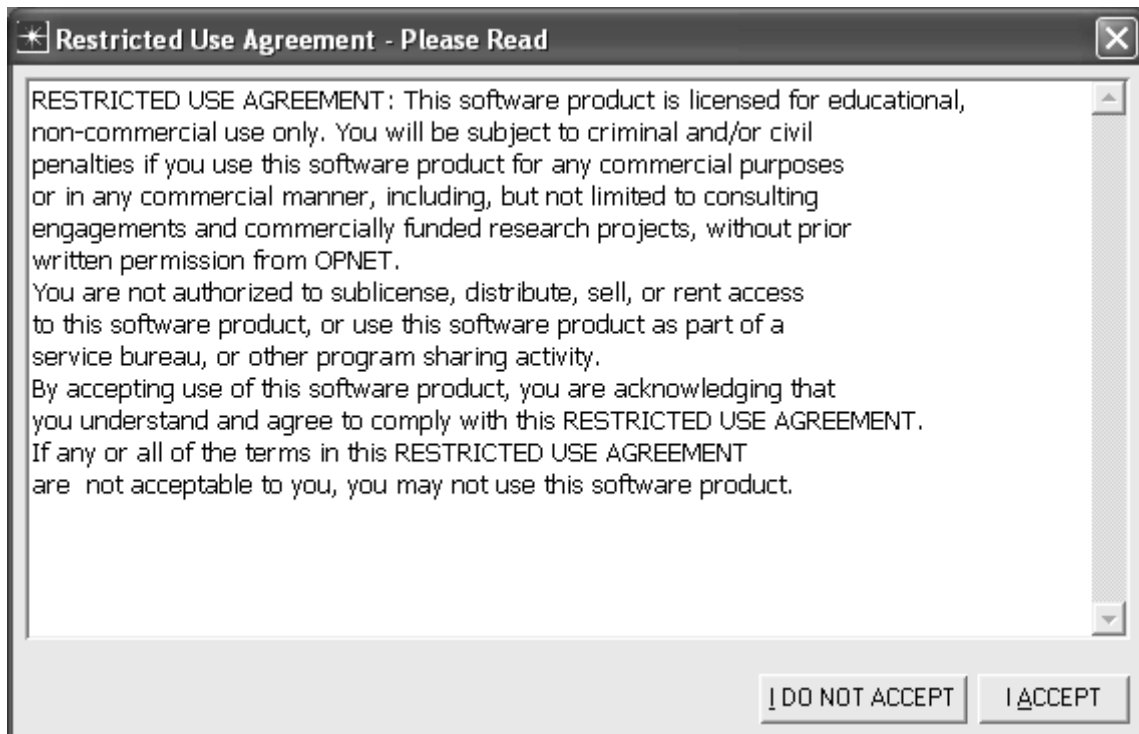


Figure 3

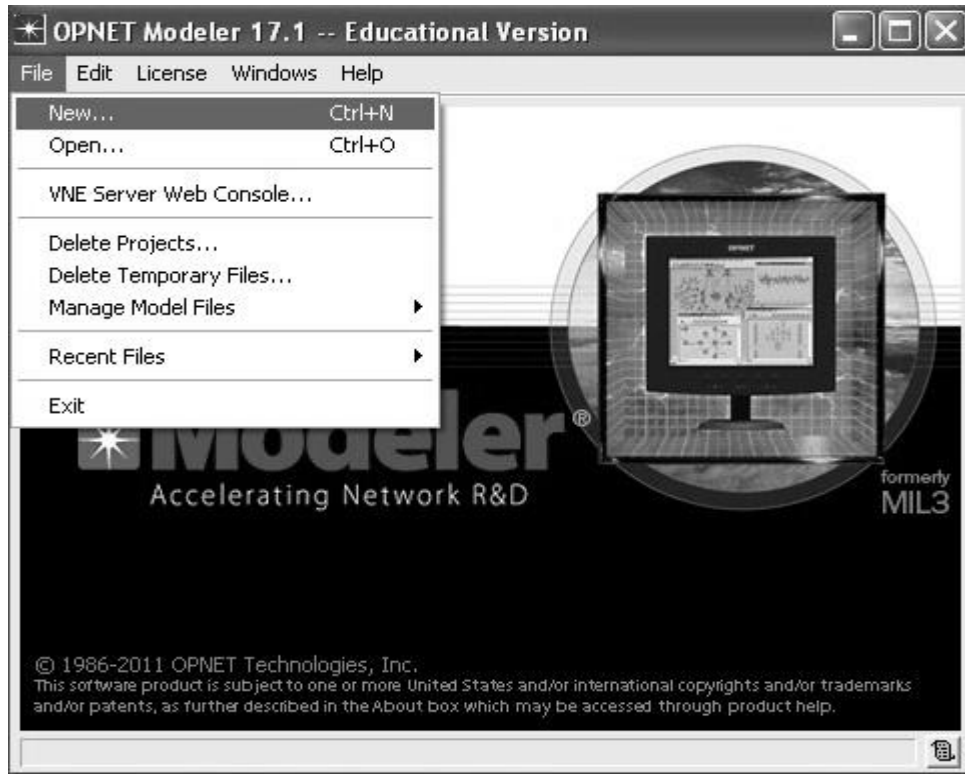


Figure 4

4. To open a new project; single click on file menu and then select new and click OK in the New window as shown in Figures 4 and 5.
5. Name the project (Figure 6).



Figure 5

6. Select create new empty scenario from initial topology windows (Figure 7).
7. Use campus as network scale as in Figure 8 and size of it as shown in Figure 9; 1000 meters to 1000 meters.
8. In the technologies selection just chose MANET as in Figure 10.

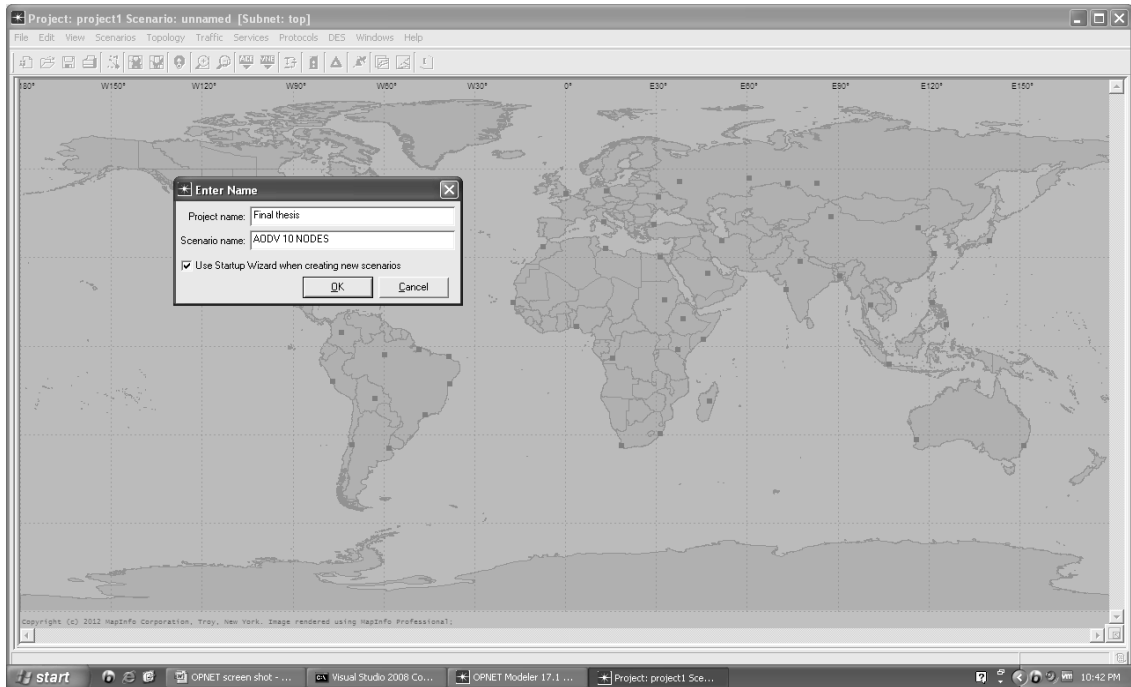


Figure 6

9. As in Figure 11, select finish to go to the next step.

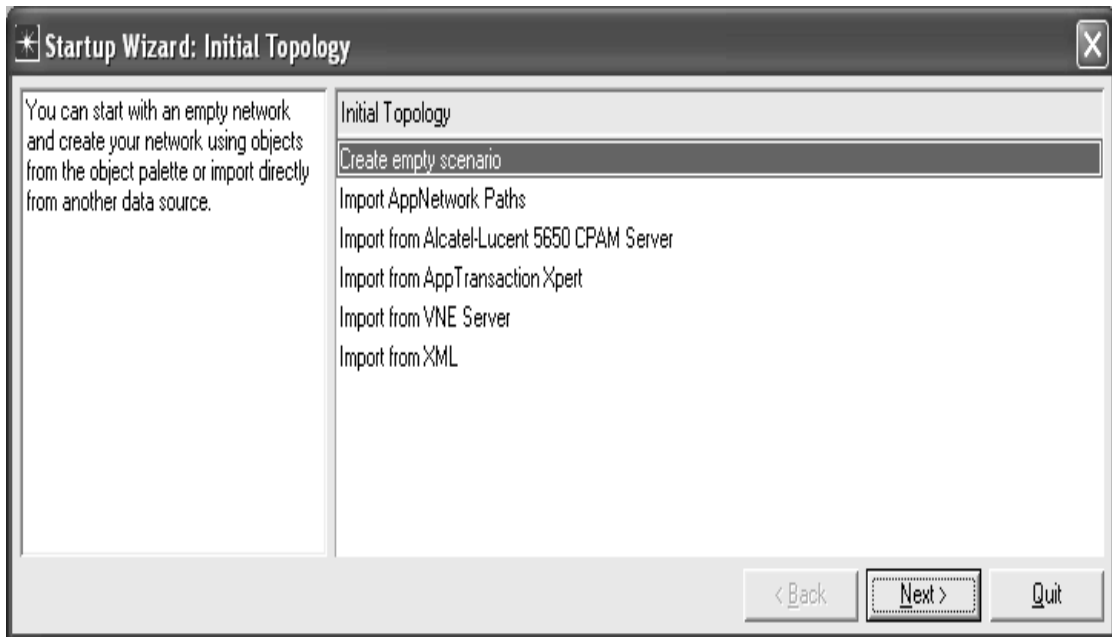


Figure 7

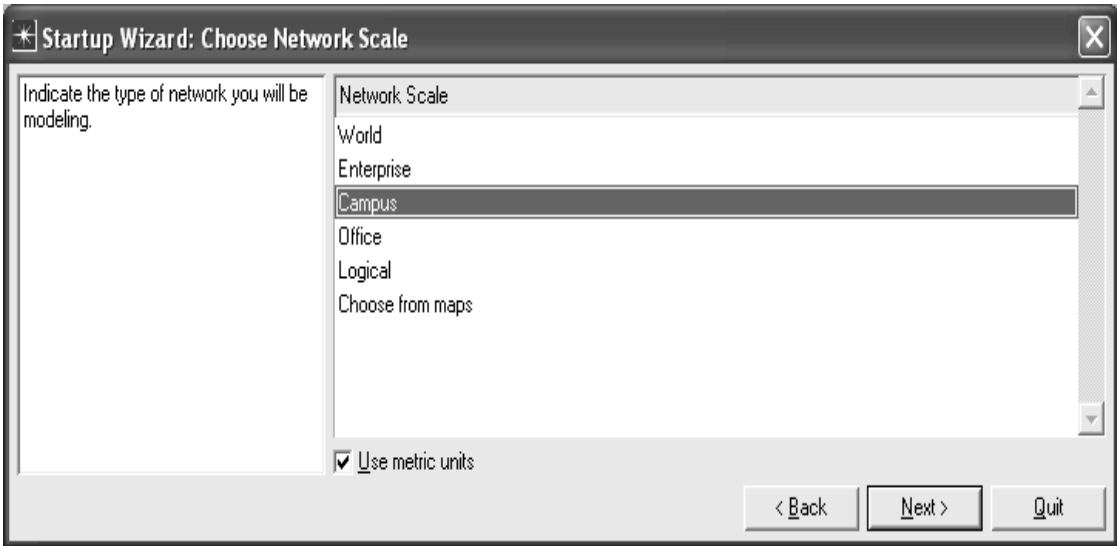


Figure 8

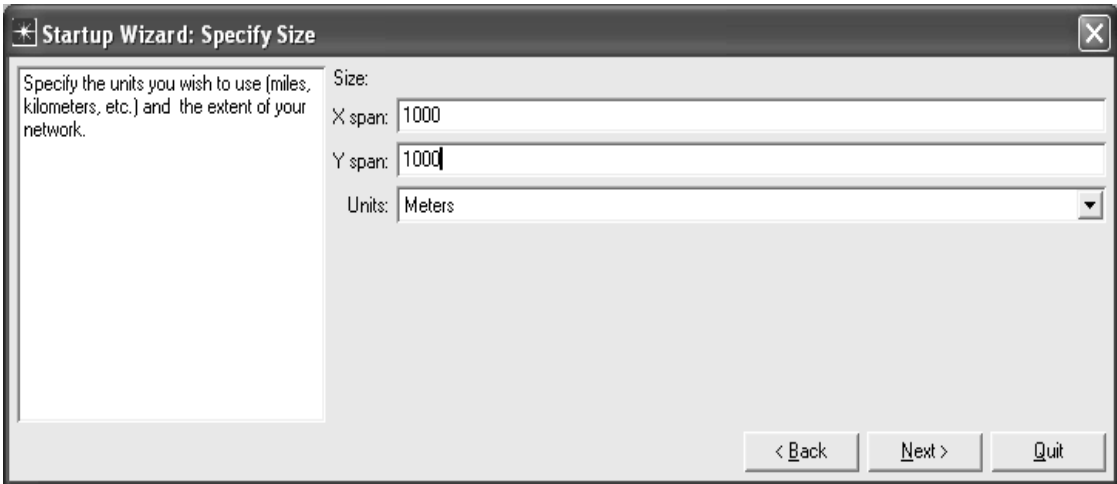


Figure 9

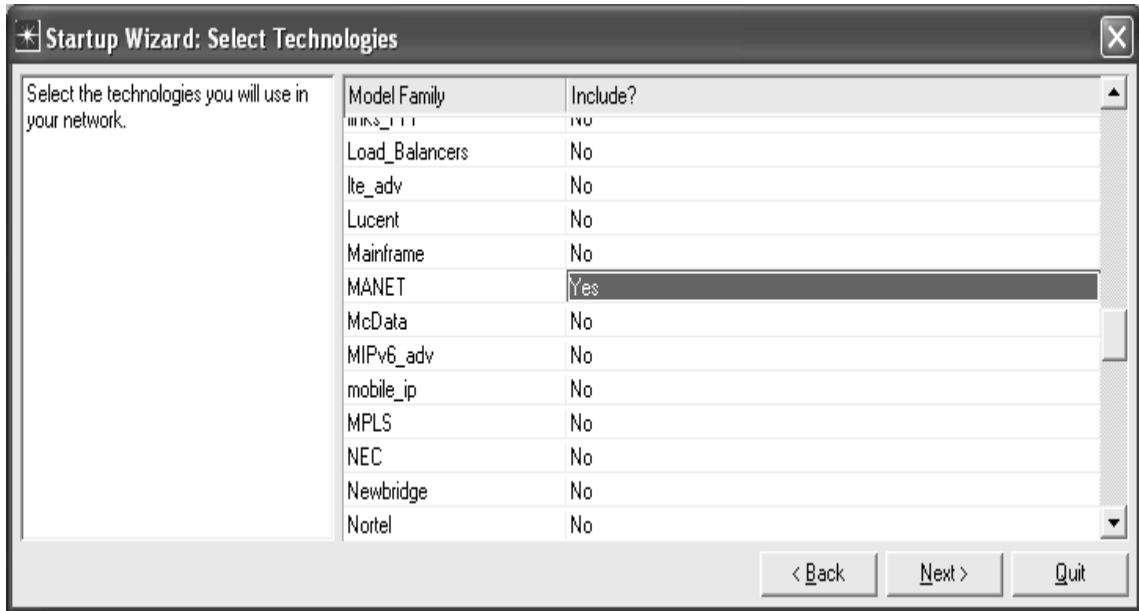


Figure 10

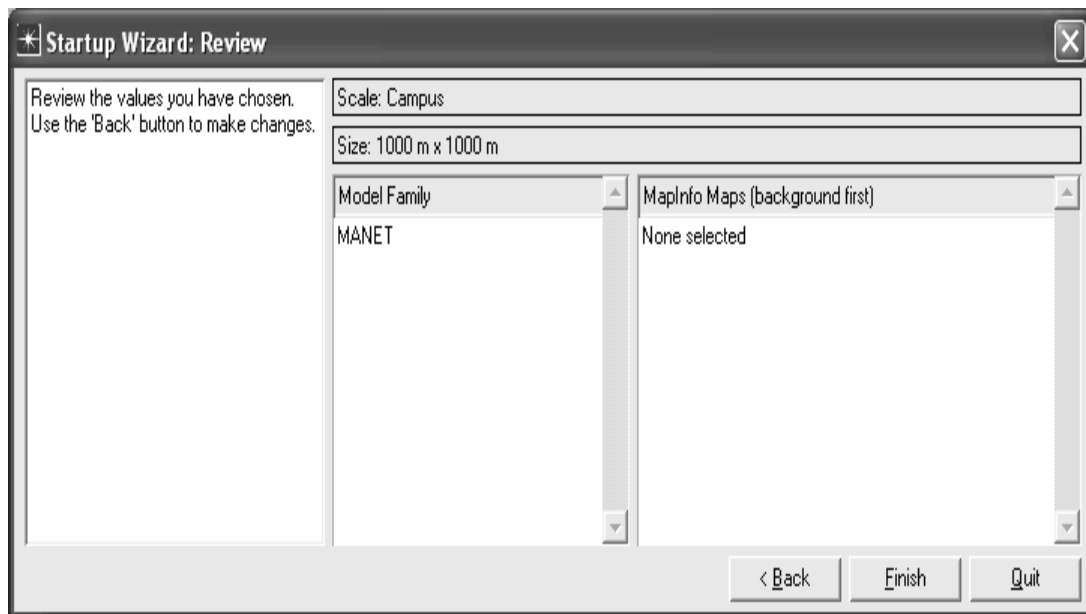


Figure 11

10. Click on Open Object Palette Tree; choose Mobility Config and drag it on the campus network and then choose mant_station (Mobile Node) and drag it on the campus network. (Figures 12 and 13)
11. Also depending on the scenario manet_station can be selected (Mobile Node) as explained in part I.
12. Right click on Application Config, ProfileConfig, Mobility Config and wlan_wksth (Mobile Node) respectively to set their name one by one.(Figure 14,15)

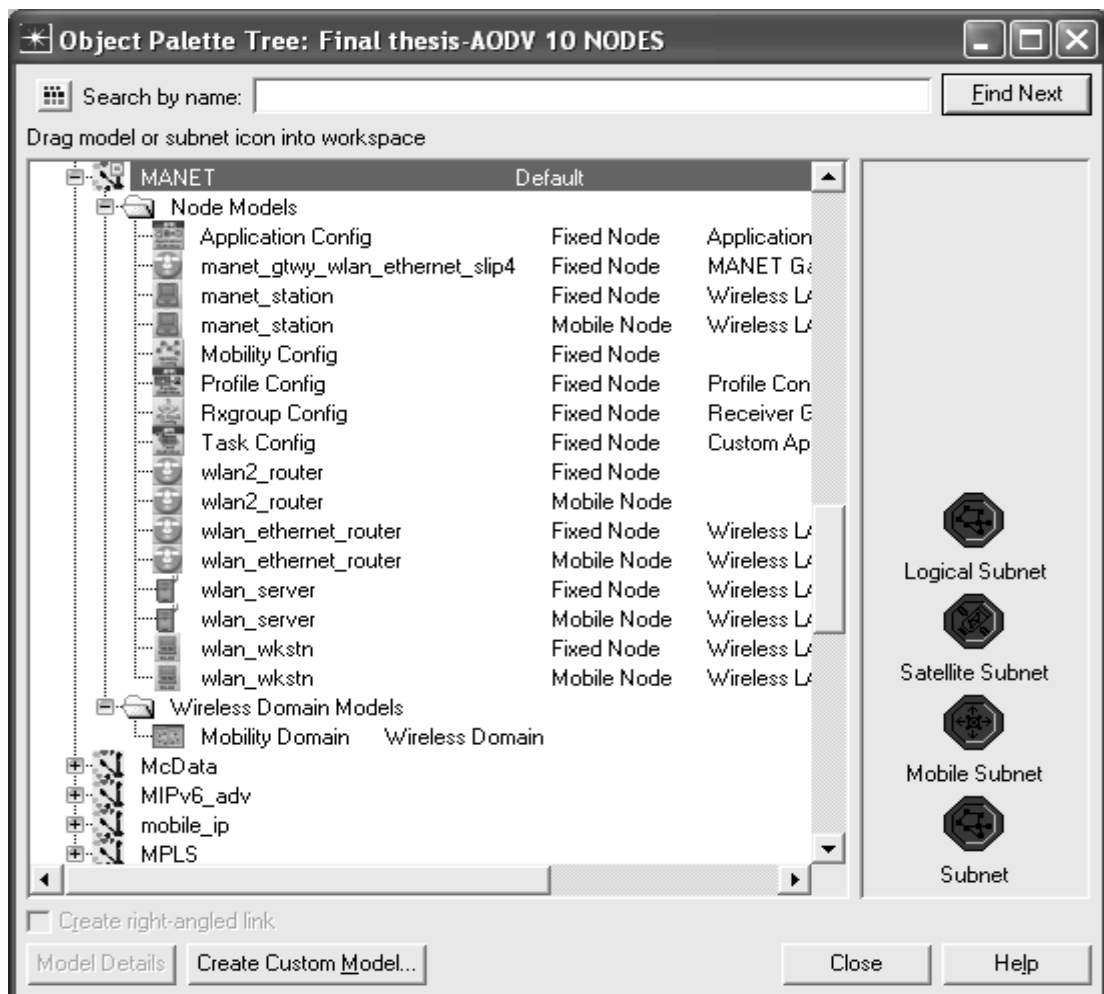


Figure 12



Figure 13

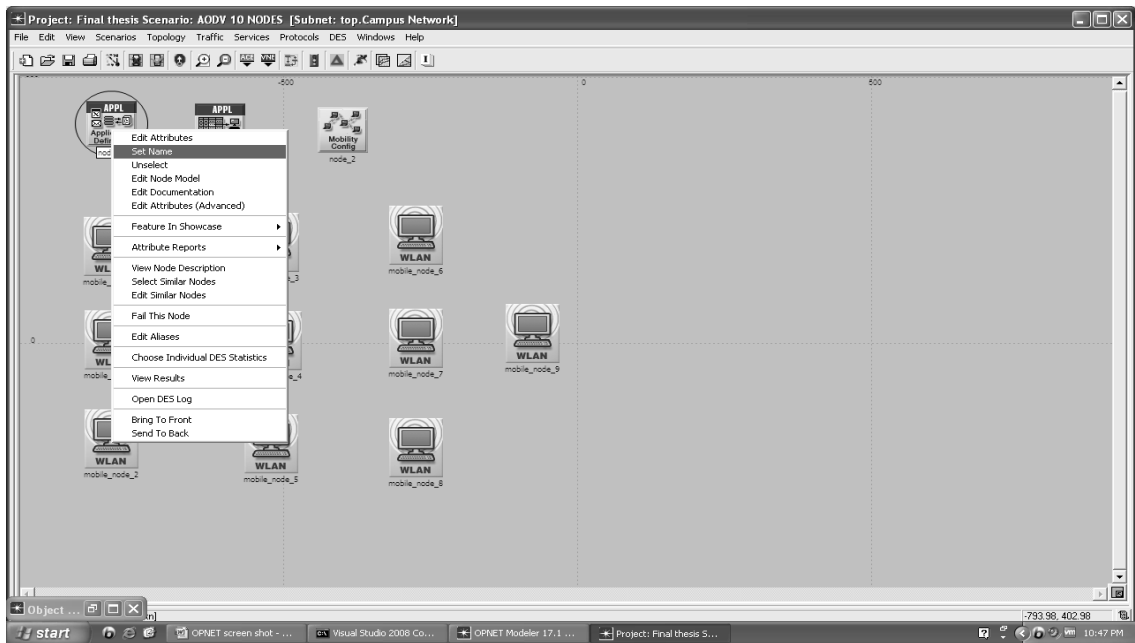


Figure 14



Figure 15

13. Select all in subnet in edit menu.

14. As shown in Figure 16 when all the nodes are selected, go to the protocol menu addressing item then select auto-assign IPv4 addresses (Figure 17)

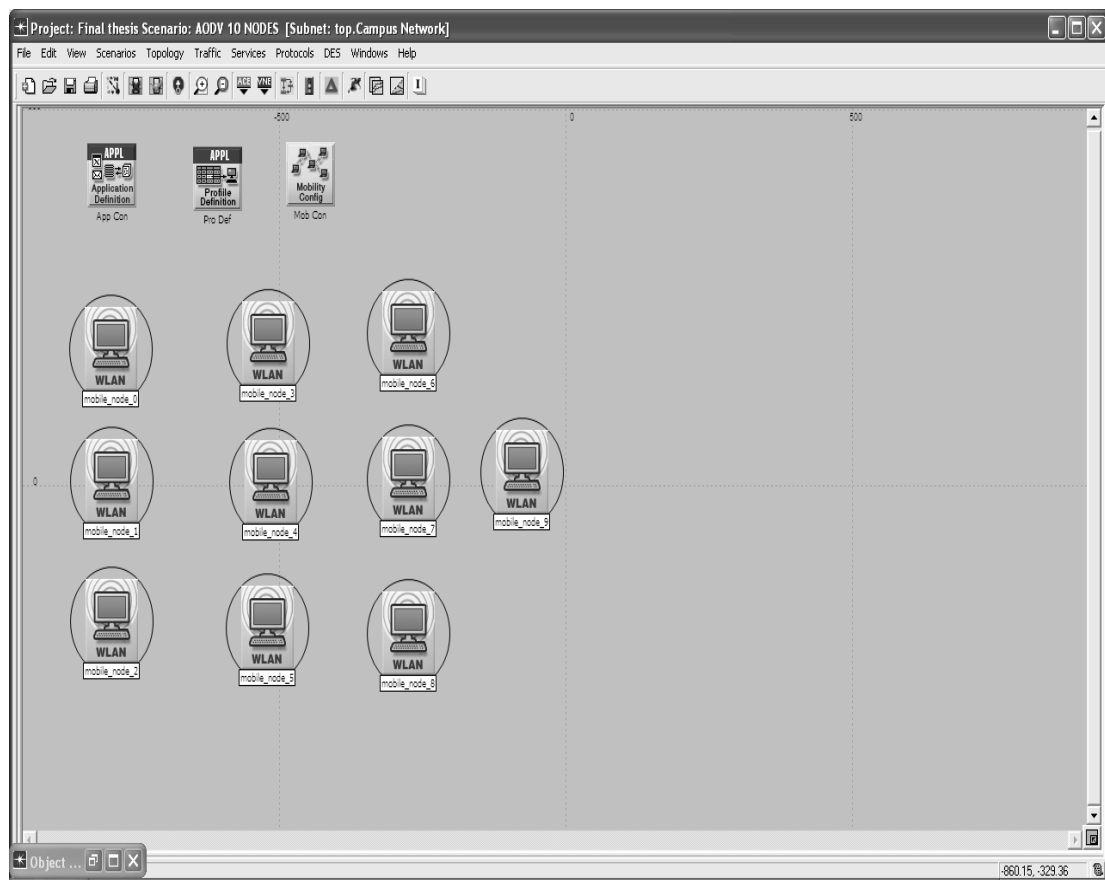


Figure 16

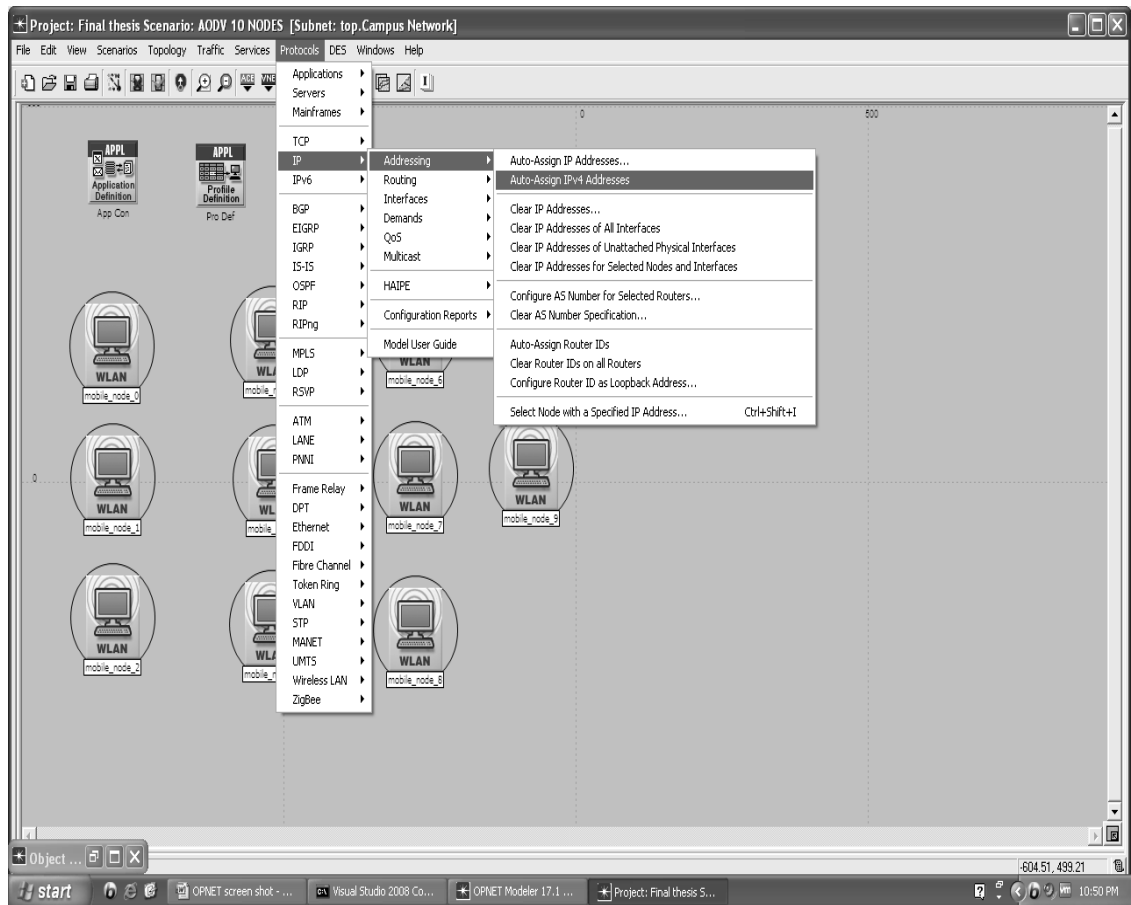


Figure 17



Figure 18

15. Select edit attributes as illustrated in Figure 18 when all the nodes are selected.
16. Choose the proper protocol; as can be seen in Figure 19, OPNET 17.1 has five routing protocols.
17. Don't forget to mark the "apply to selected object" check box and then click OK.
18. Figure 20 shows AODV parameters.

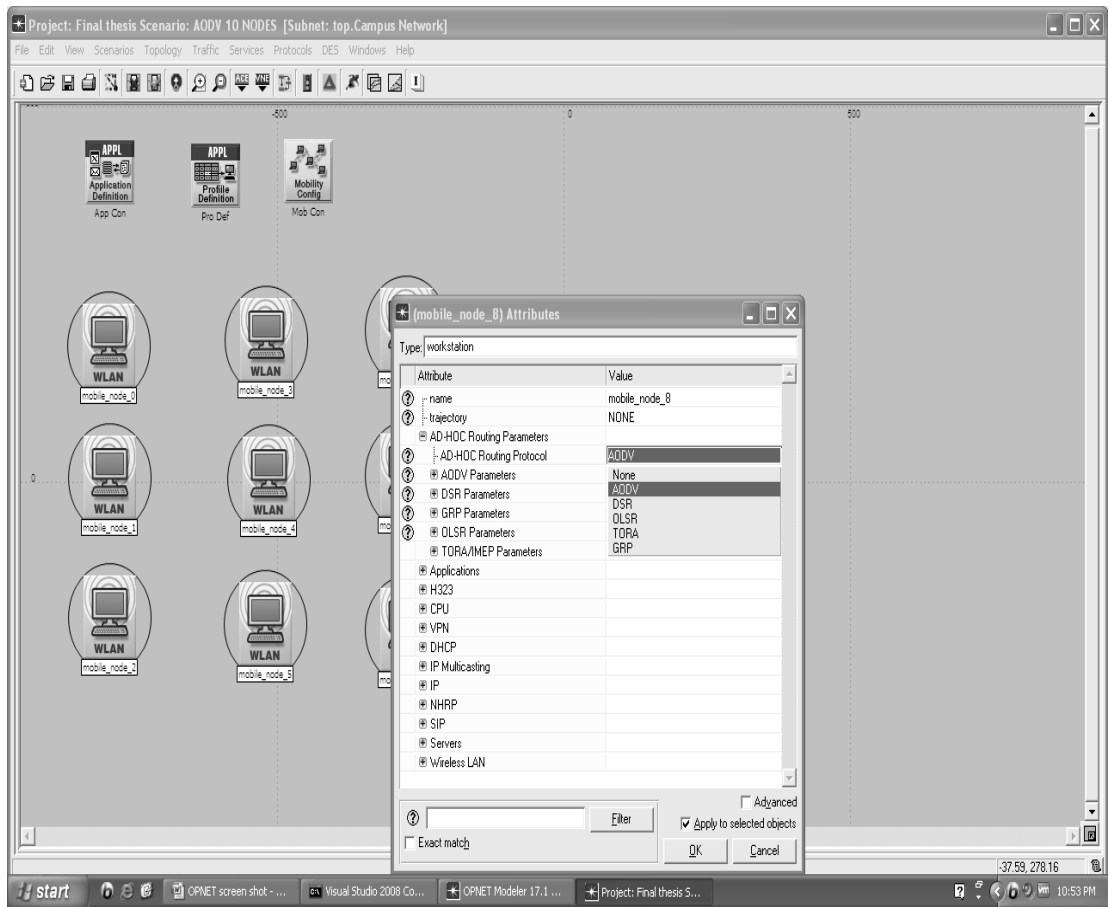


Figure 19

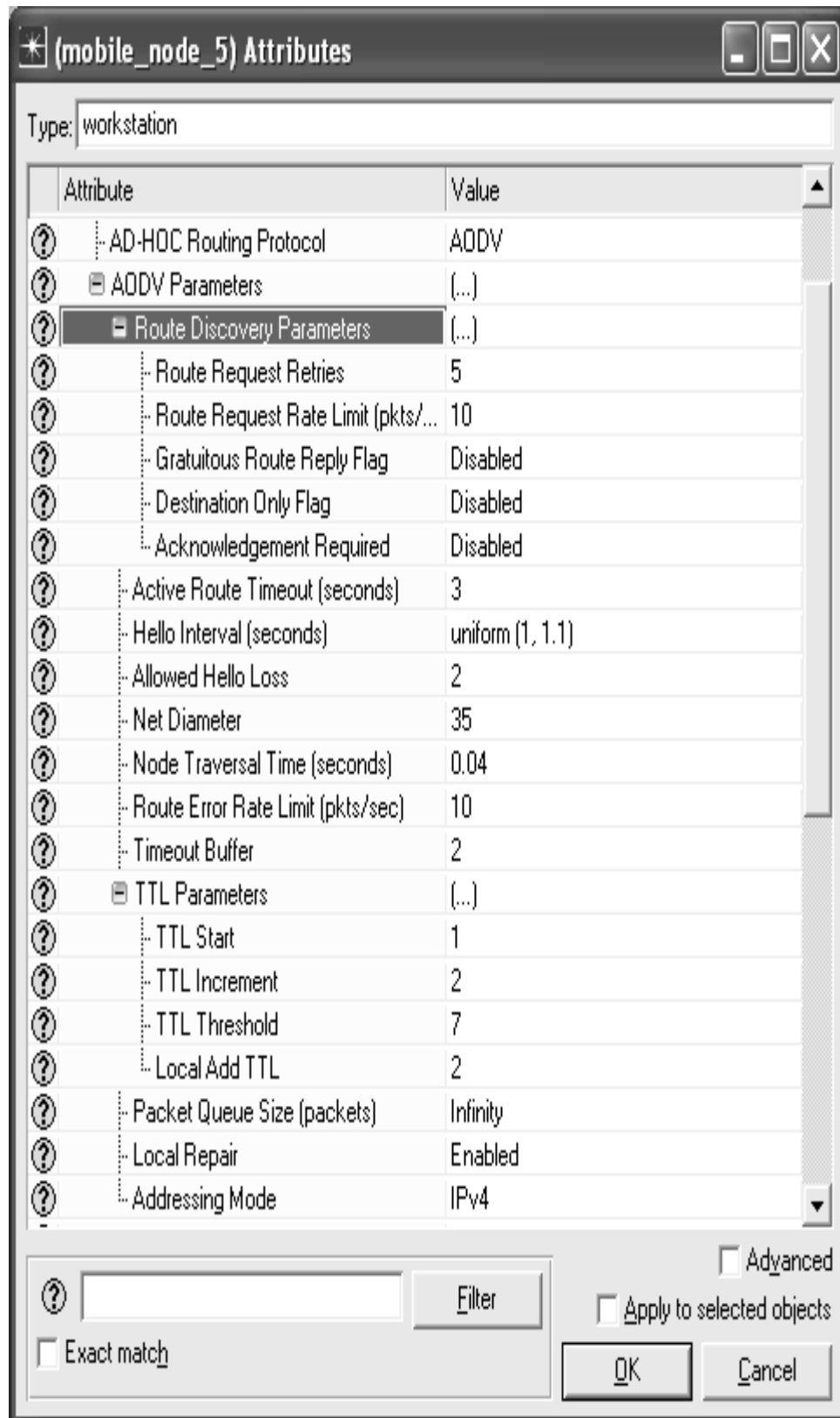


Figure 20

19. Figures 21 and 22 show OLSR parameters.

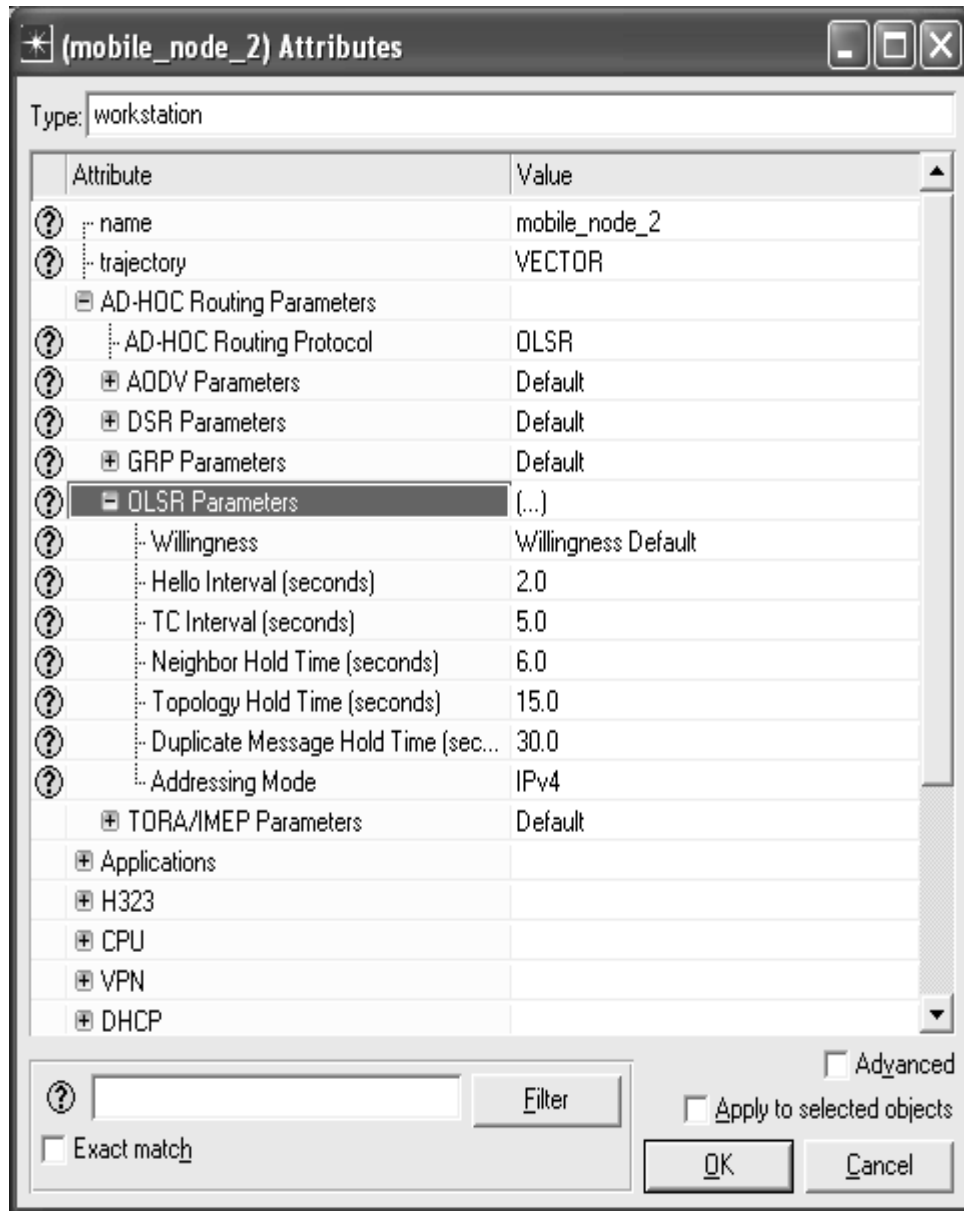


Figure 21

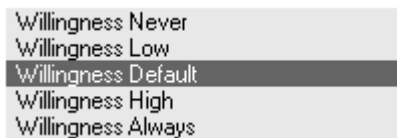


Figure 22

20. Figure 23 show TORA parameters and different mode operation of TORA is illustrated in Figure 24.

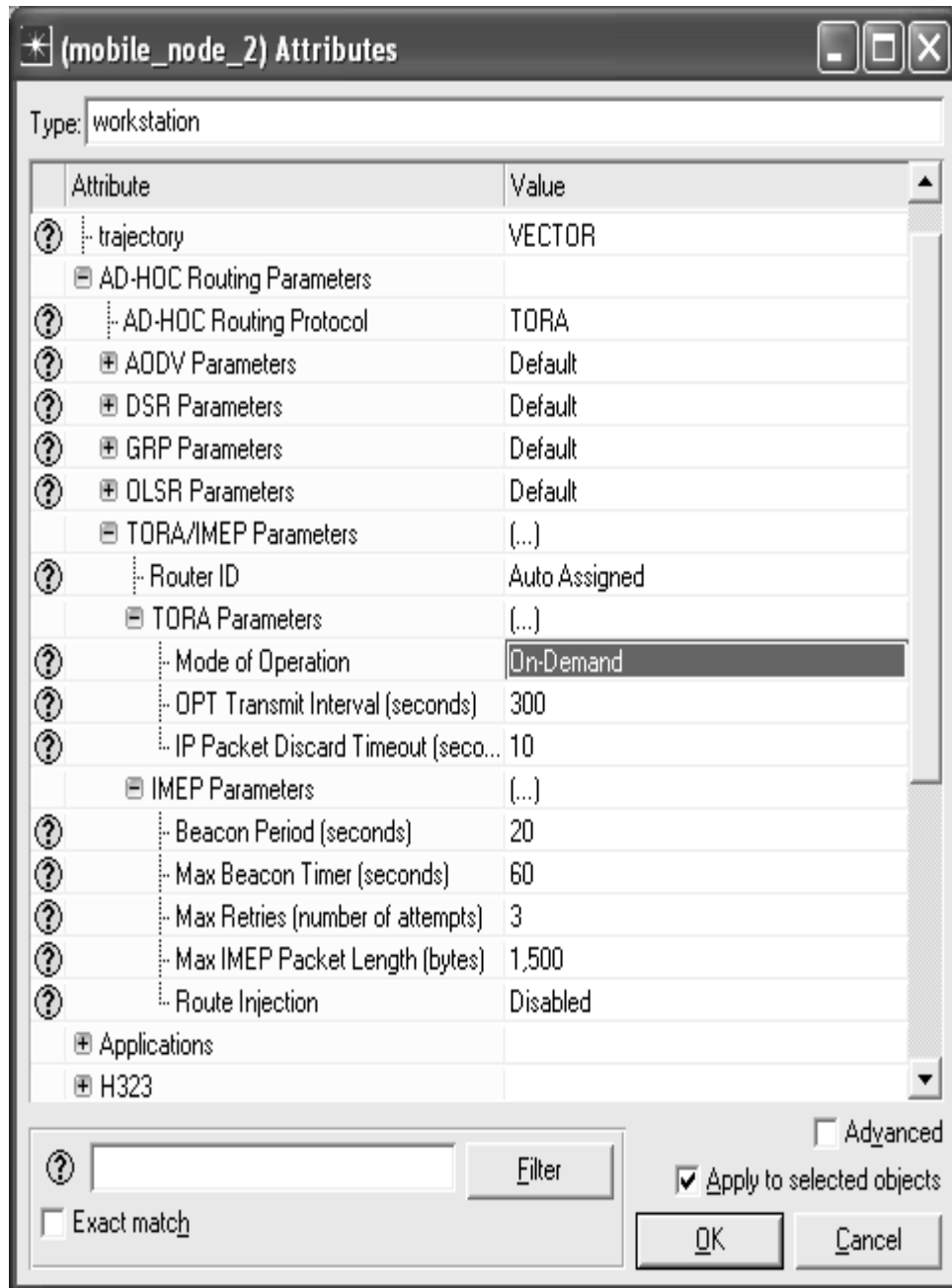


Figure 23



Figure 24

B: Configuration of application

In this part, the application will be setup which will spread out in the profile configuration.

1. Right click on Application Configuration and select edit attributes.(Figure 25)
2. Select number of rows to one.
3. Register the name as an FTP for one of the row.
4. Choose Ftp as description also Low load and click OK. (Figure 26, 27). In the third scenario different parameter of Ftp will be selected.
5. File size can be changed as shown in Figure 28.

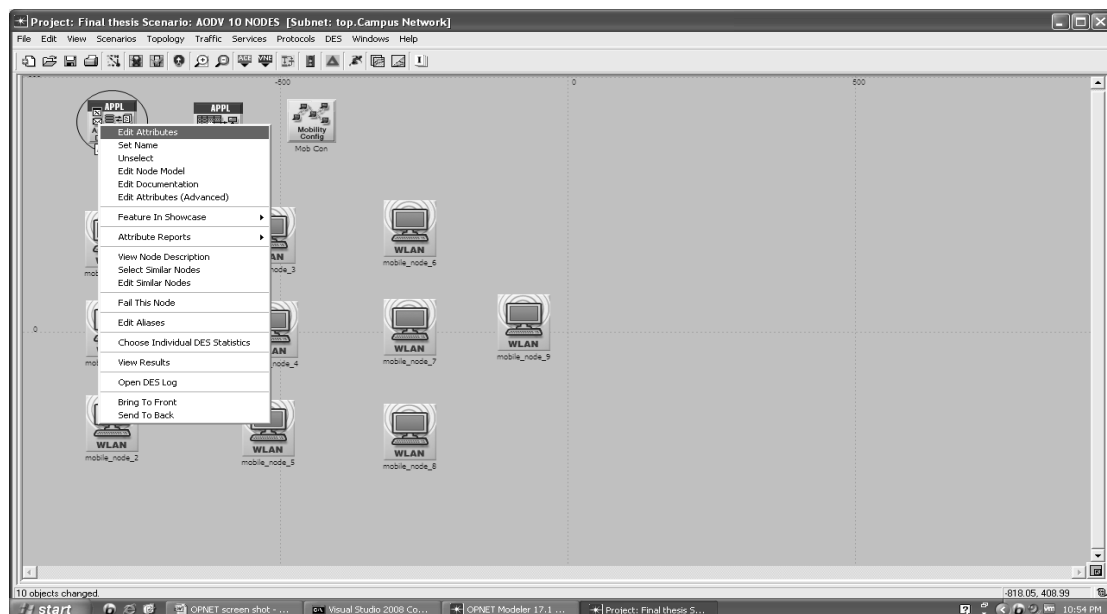


Figure 25

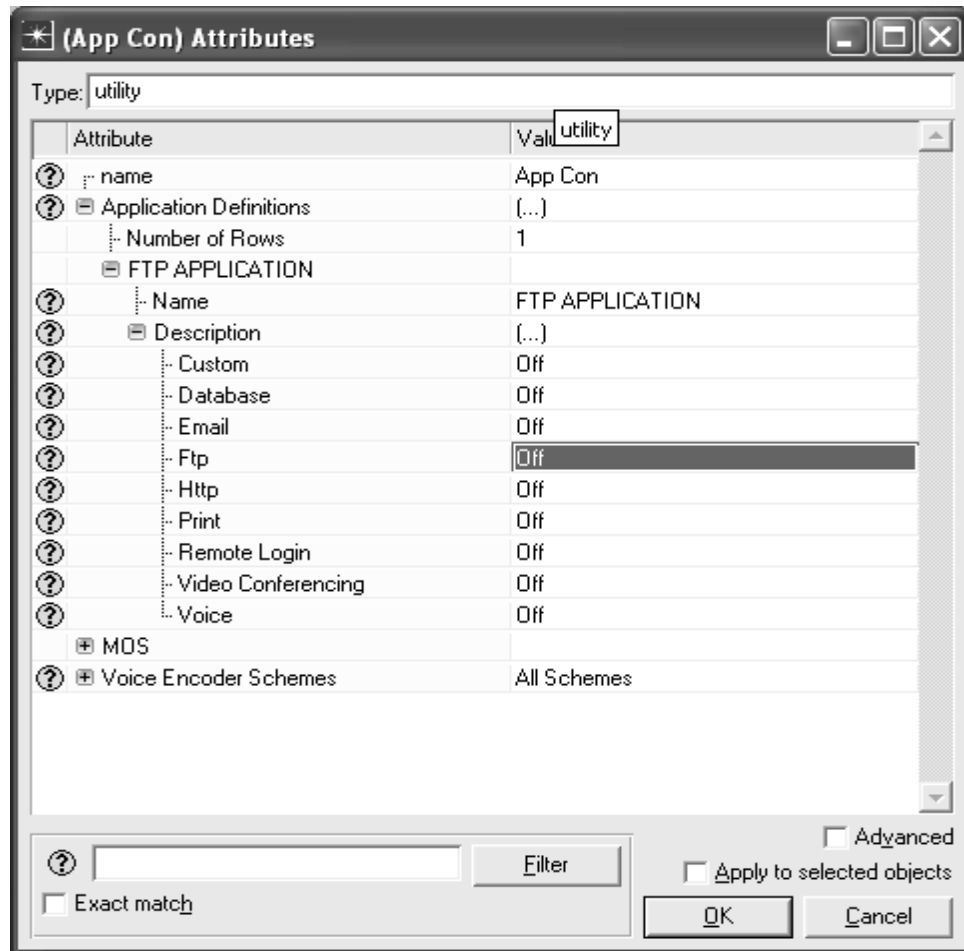


Figure 26



Figure 27

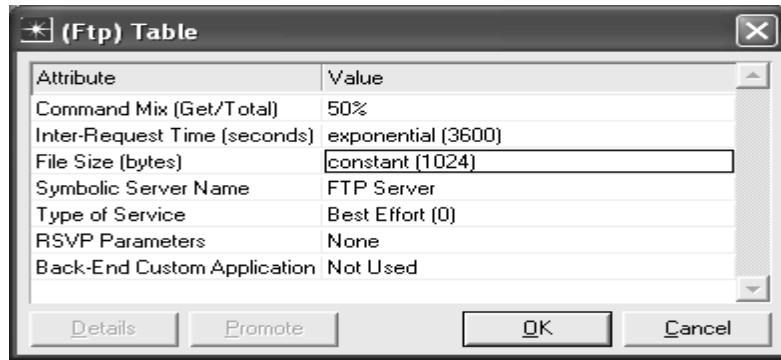


Figure 28

C: Configuration of Profile

This part will specify the traffic pattern followed by the application as well as the configured profiles in this object.

1. Right click on Profile Definition and select edit attributes. (Figure 29)
2. Enter one as a number of rows as shown in Figure 30.
3. Enter the profile name.
4. Register the number of rows to one and choose FTP belong to applications which are selected in last procedure.
5. Set the “start time” offset to constant 100 under FTP and “duration” to End of profile. (Figure 33)

This attribute has two interpretations based on the value specified for the "Operation Mode". 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.

6. As can be seen in Figure 32, OPNET has a different parameter for time; it is available to use whenever it is needed.
7. Belong to “FTP repeatability” set “inter-repetition time (seconds)” to “once at start” (Figure 34, 35).
8. Fix the “start time” to constant 0 and “duration” to “end of simulation”.(Figure 36)
9. Leave the rest as default which is shown in Figure 37.
10. Click OK.

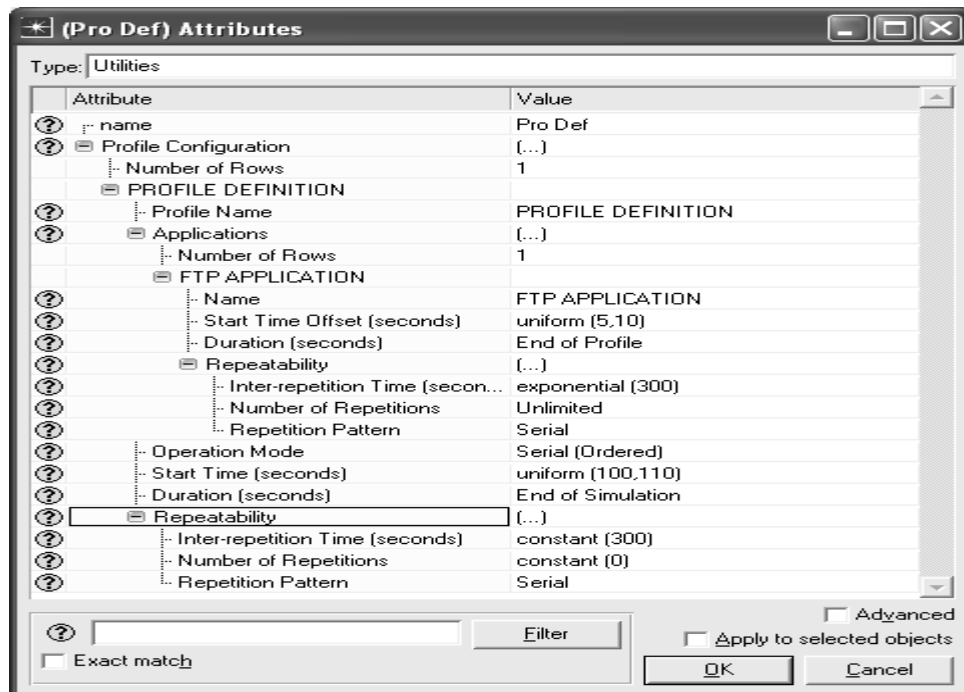


Figure 29

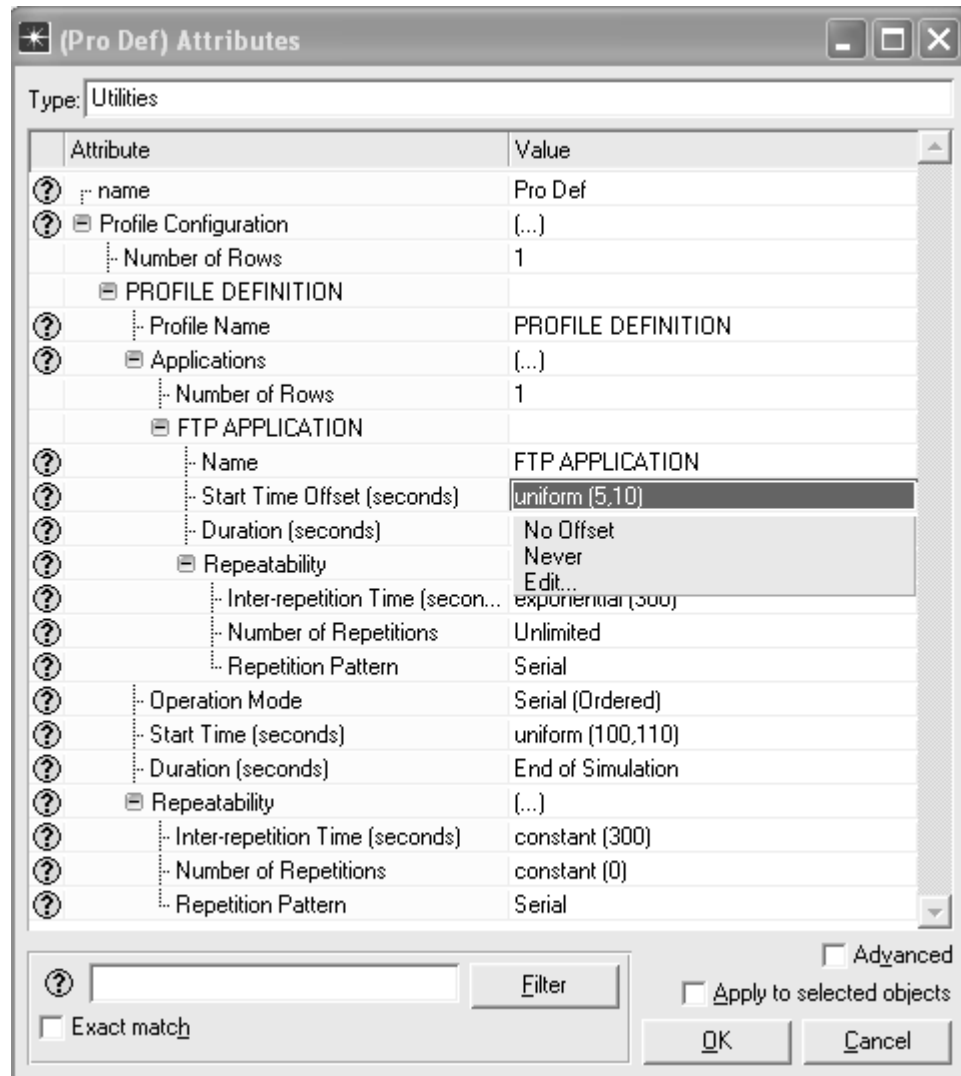


Figure 30

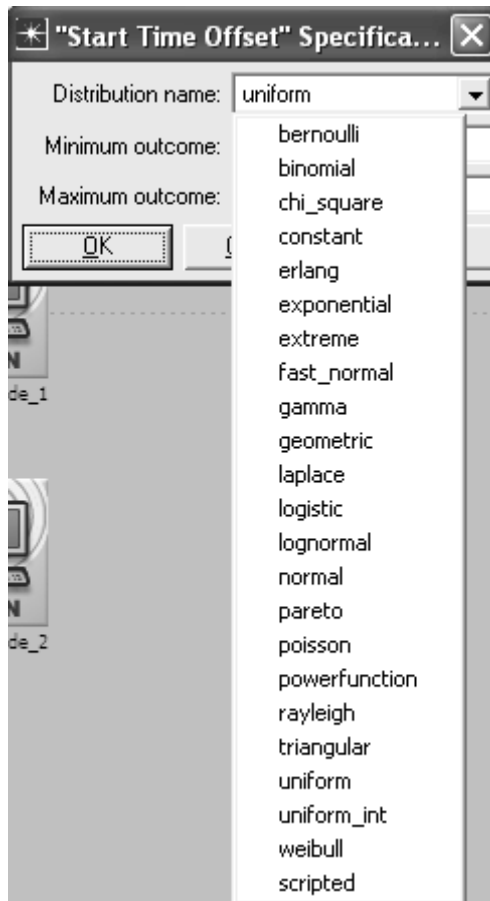


Figure 31

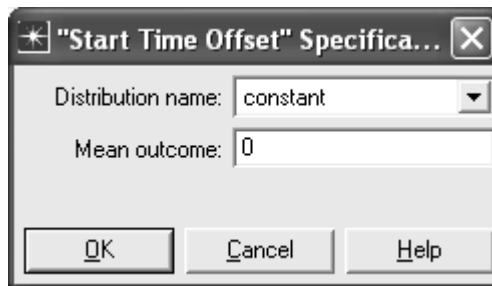


Figure 32

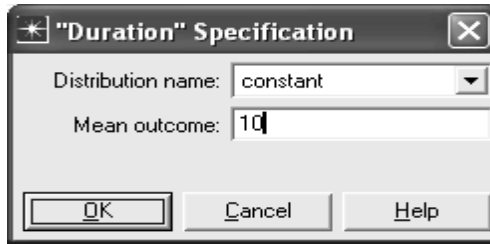


Figure 33

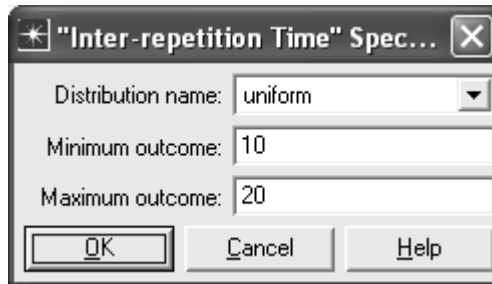


Figure 34

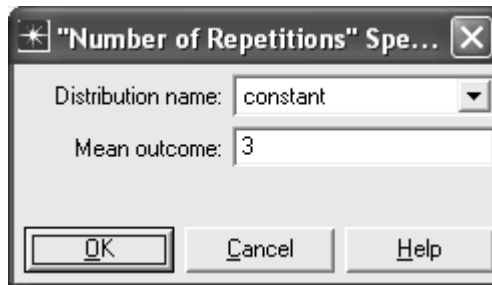


Figure 35

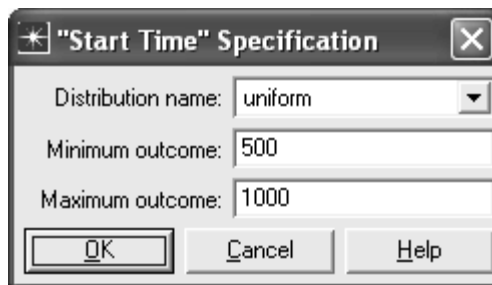


Figure 36

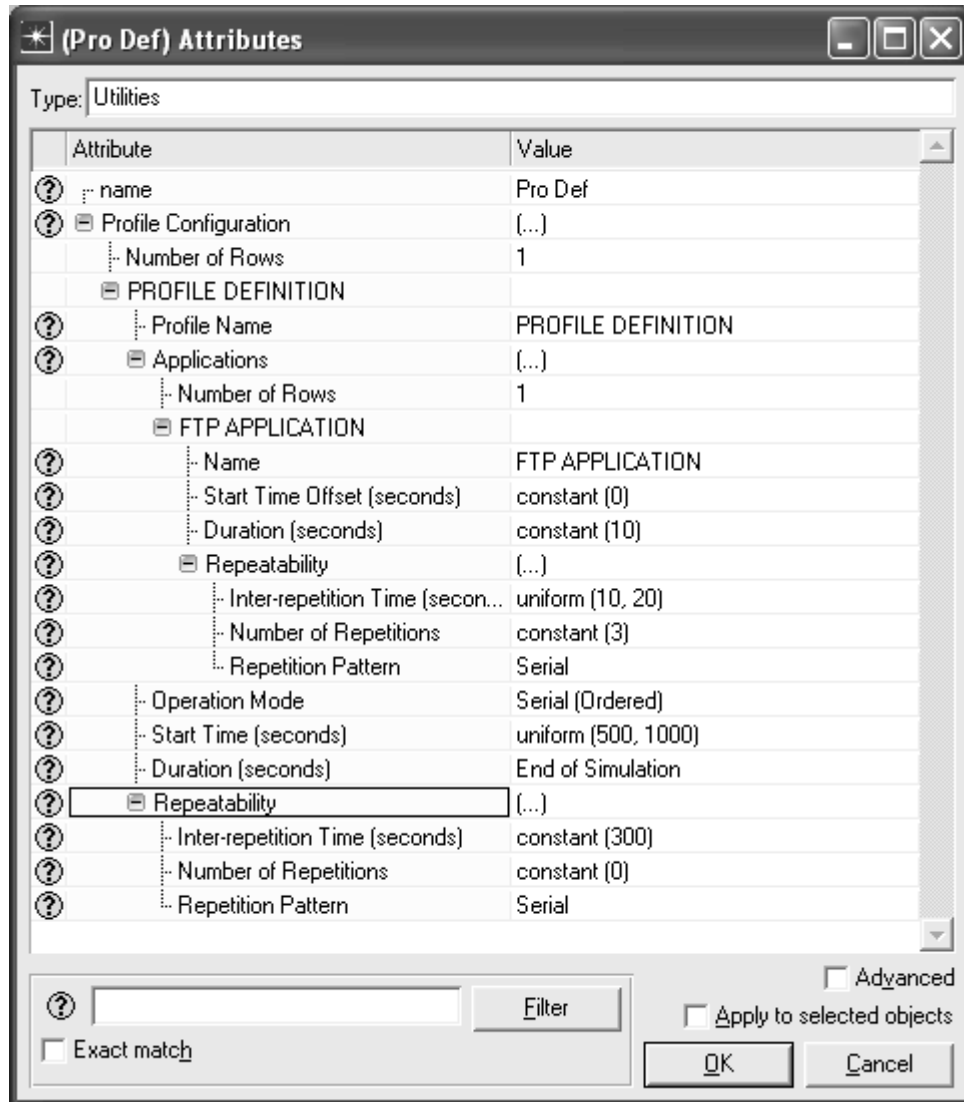


Figure 37

D: Configuration of mobility

In this part, definition of the mobility pattern that the nodes will follow will be explained. The “random waypoint mobility model” was selected for the simulations.

1. Select “edit attributes”.
2. Develop “default random waypoint”.
3. Set “speed” for first scenario to constant 5. Notice that for the second and third scenarios, the “speed” will be 30 and 50.
4. Fix “pause time” to constant 100.
5. “Start time” constant 0.
6. The rest as default. All these steps are illustrated in Figure 38.
7. Select Topology from pull down menu then select Random Mobility to deploy the “mobility profile”. Set mobility profile as shown in Figure 39.
8. Set “the default random waypoint profile” as shown in Figure 40.

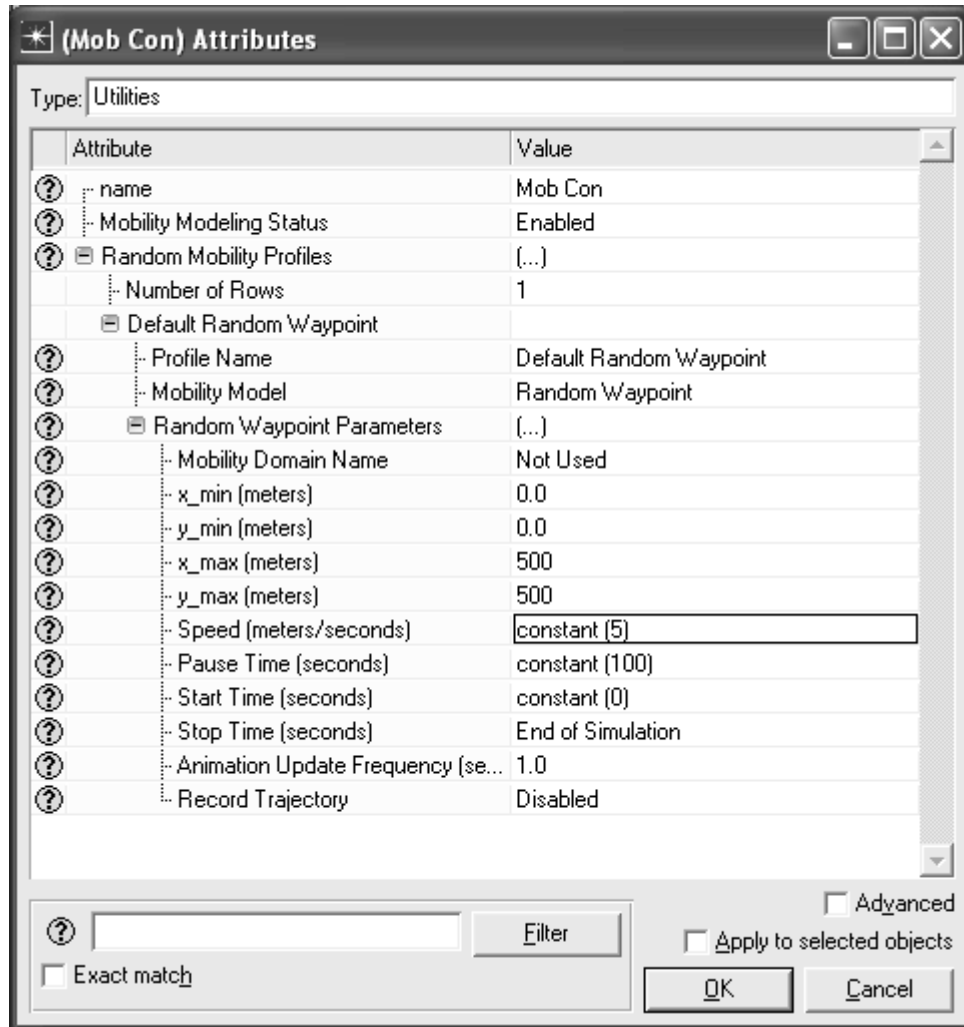


Figure 38

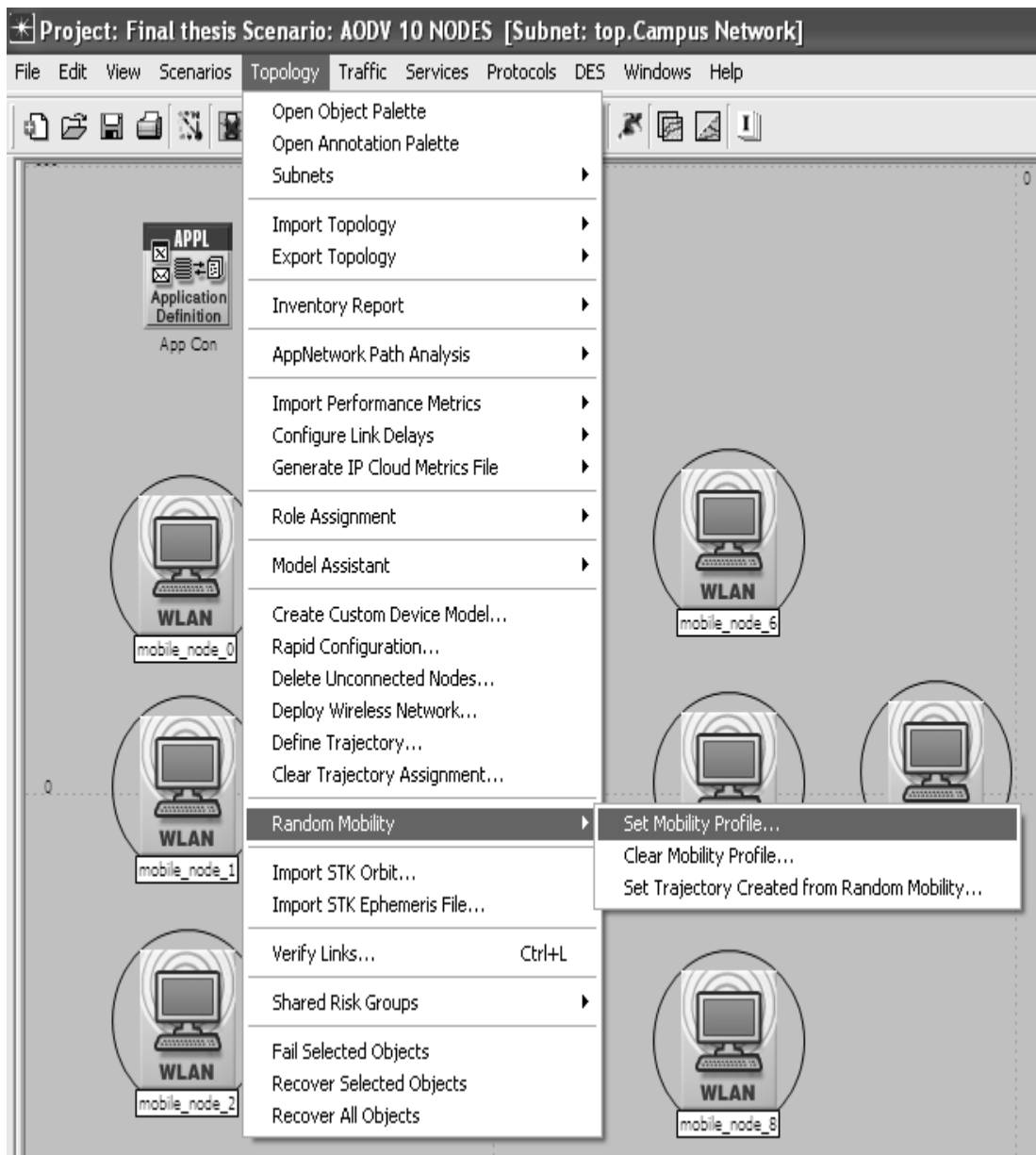


Figure 39

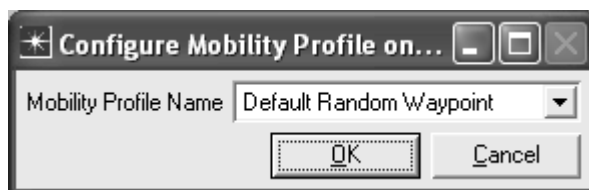


Figure 40

E: Collecting the Statistics of simulation

These procedures show methods of collecting global statistics for all the nodes.

1. Click right in the main window then choose “choose individual DES statistics” as illustrated in Figure 41.
2. Select “global statistics” and choose AODV, TORA_IMEP, OLSR, FTP and wireless LAN (Figure 42).

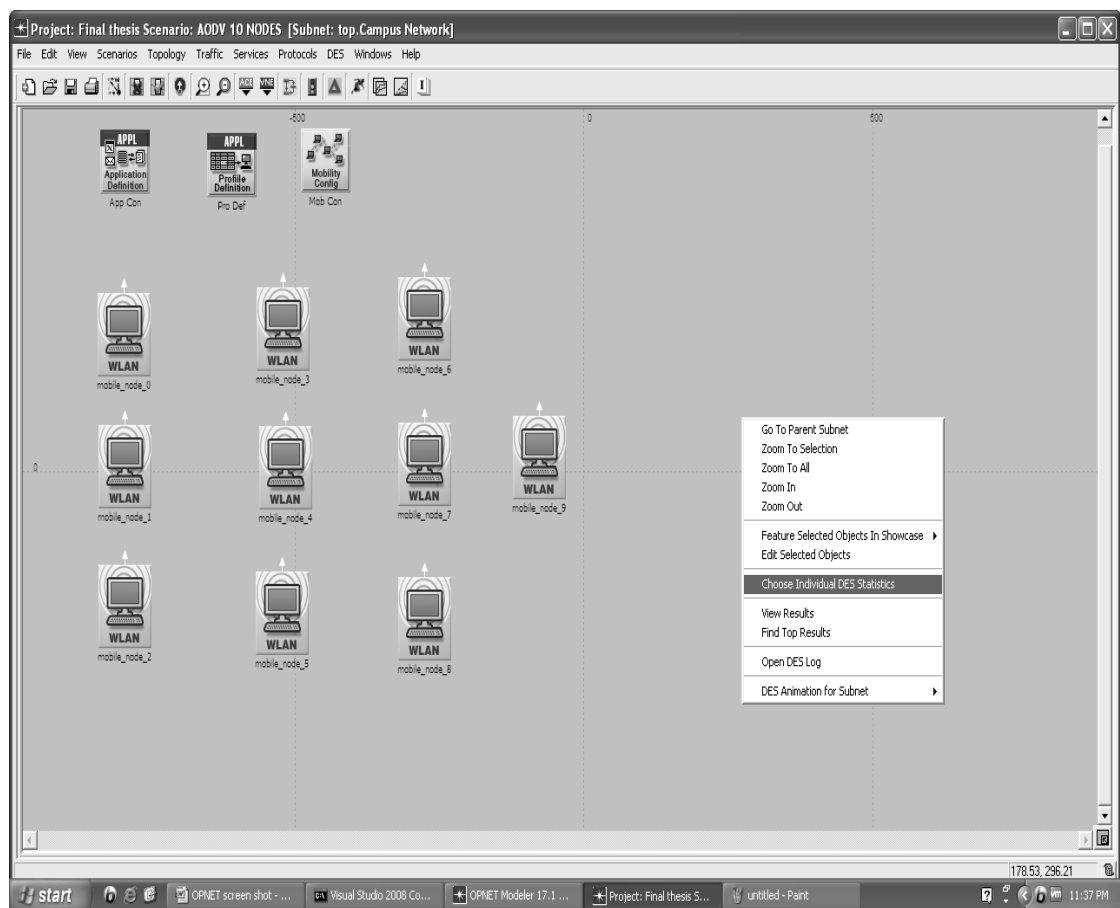


Figure 41

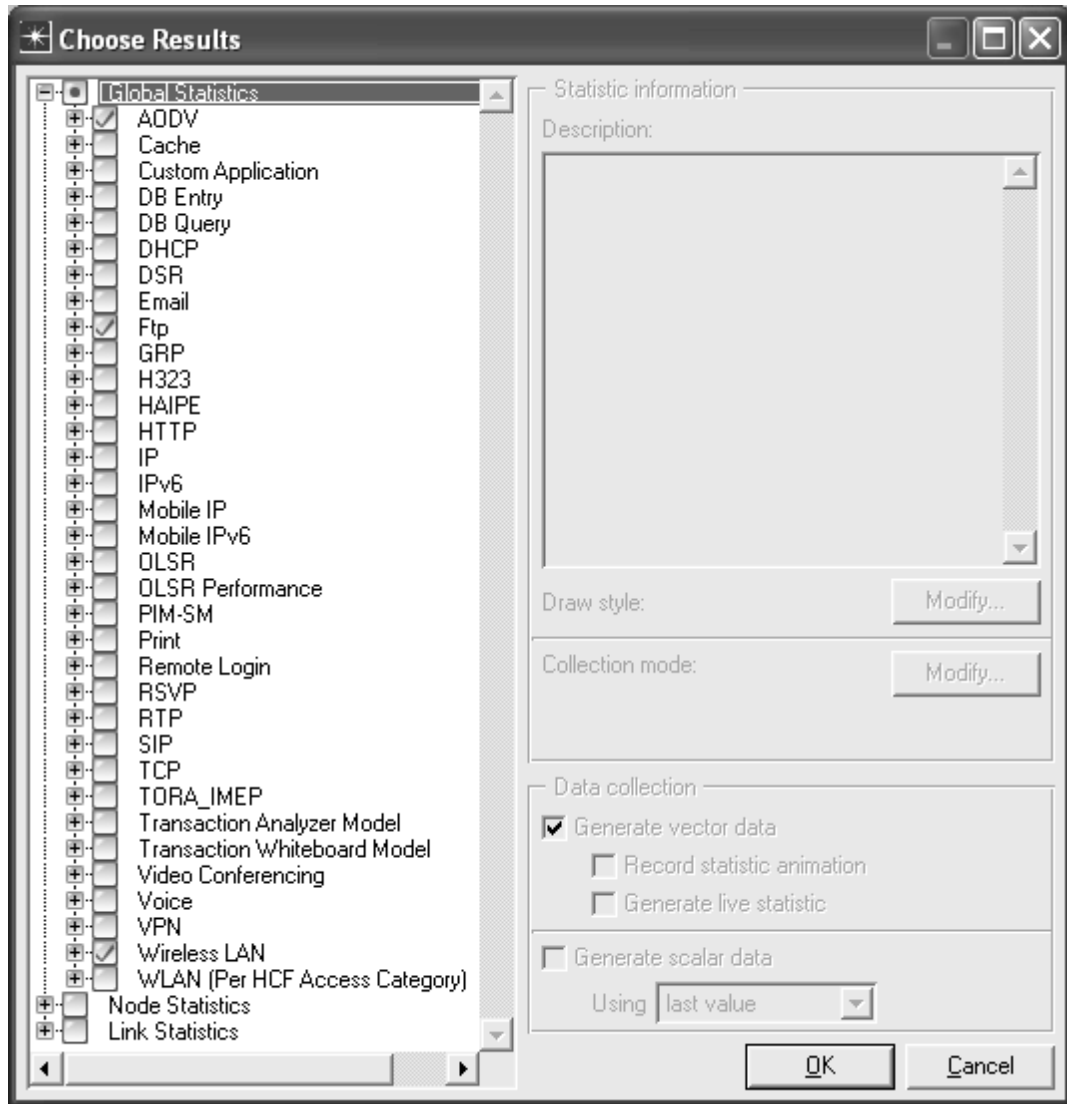


Figure 42

Notice that each one of these statistics has sub tail. For collection mode there is also a modified selection.

F: Duplicate for Scenario

For comparison evaluation and duplicate scenario the following procedures will be used.

1. Go to Scenarios pull down menu and select Duplicate scenarios as shown in Figure 43.
2. Type the new name for scenario
3. Select the number of mobile nodes, speed and all things as appropriate and depending on scenarios like protocols and attribute of them.
4. Save your project.

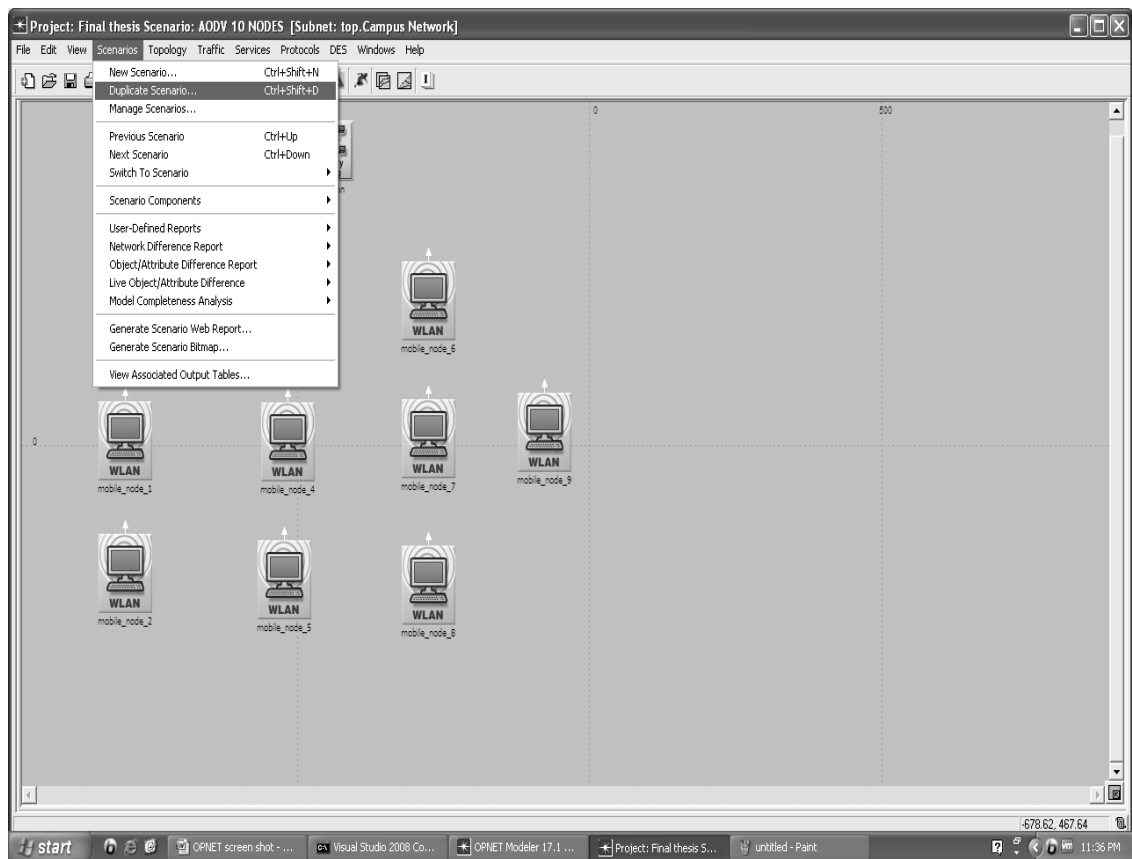


Figure 43

G: Running Simulation

1. To run the simulations, select Scenarios menu and then Manage Scenarios as shown in Figure 44.
2. As shown in Figure 45, select “collect” for all the scenarios.
3. Select the proper “simulation time” for all scenarios. Here 600 sec was selected.
4. Click on OK to run the simulations.
5. During the simulations, as can be seen in Figure 46, it is also possible to select each of them and see the result while simulation is running as shown in Figure 47.

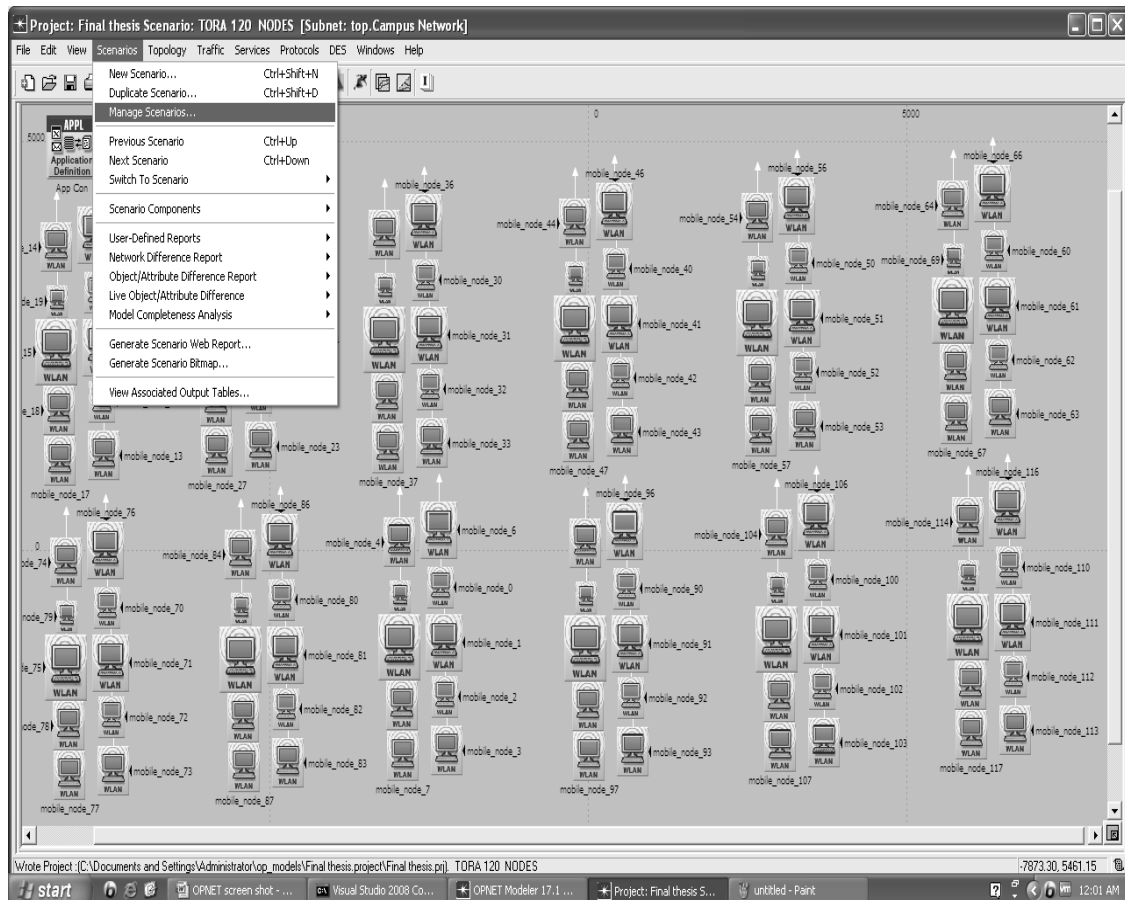


Figure 44

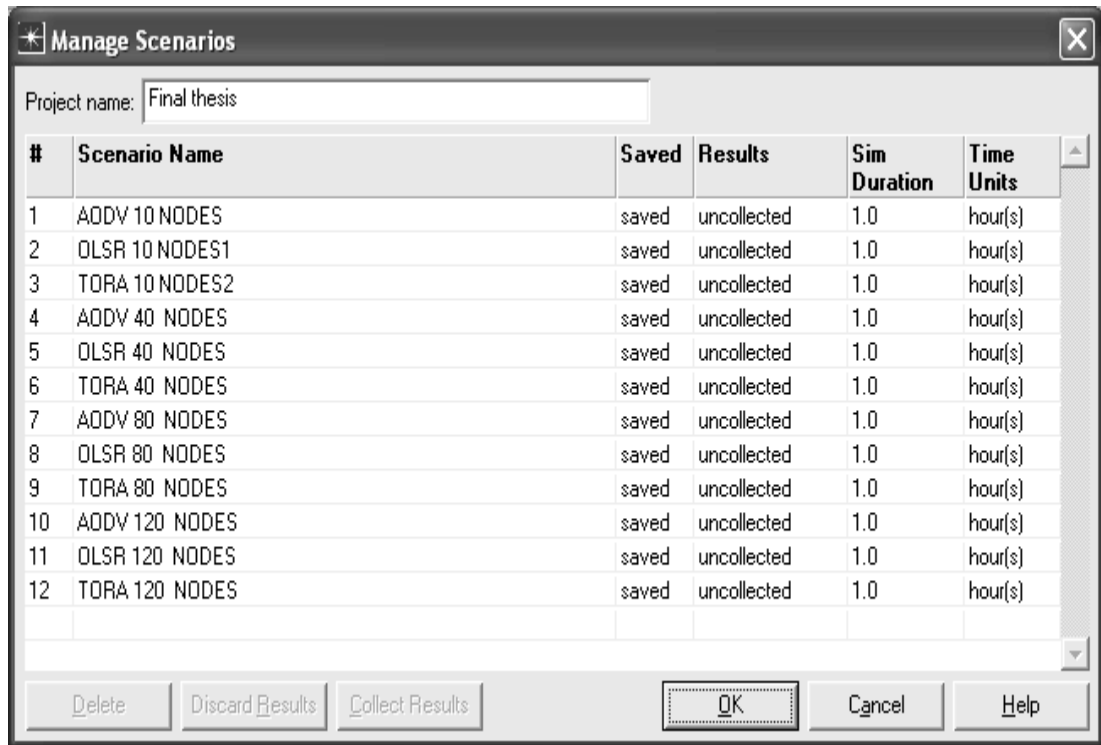


Figure 45

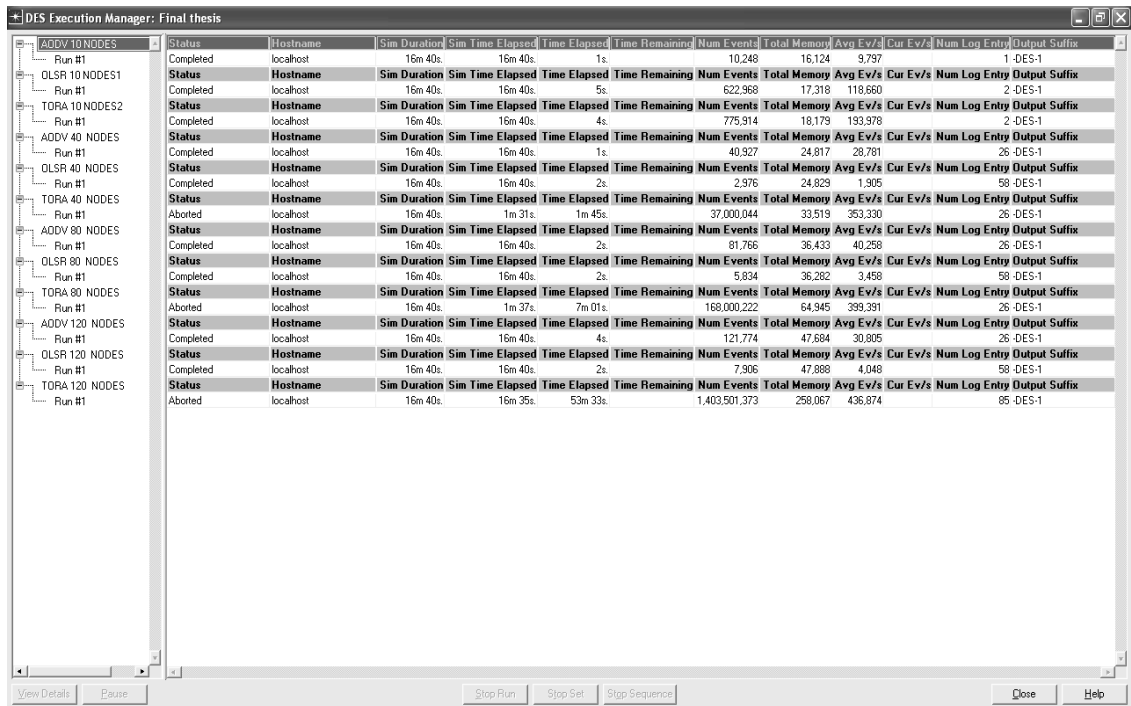


Figure 46

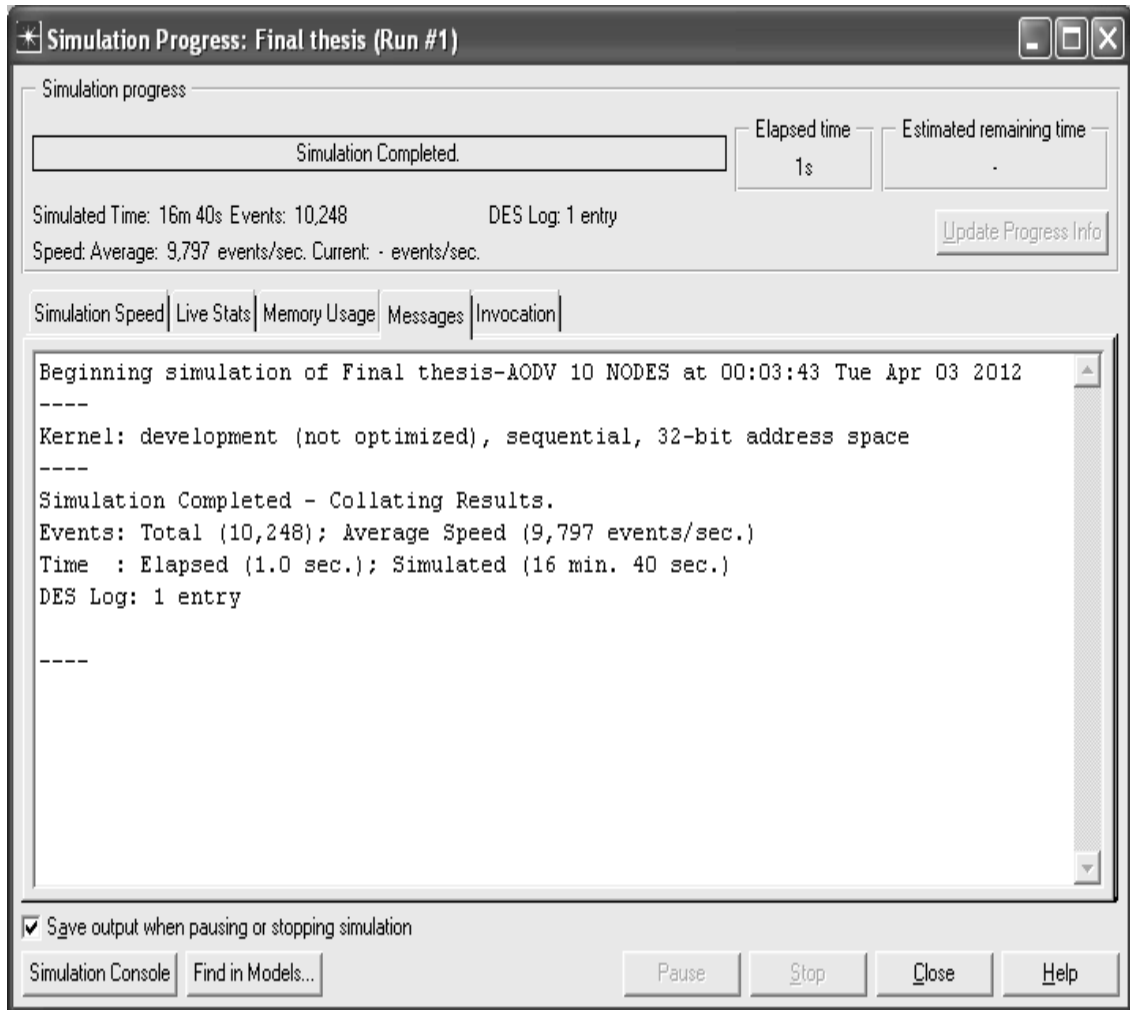


Figure 47

H: Viewing Results

To see the results, the following procedures must to be done:

1. Select “Des menu” then “Results” and go “Compare Results”.
2. Tick the scenarios which are wanted to compare as the results.
3. Below to Global statistics, select the appropriate statistics as to be displayed. All the steps are shown in Figure 48.

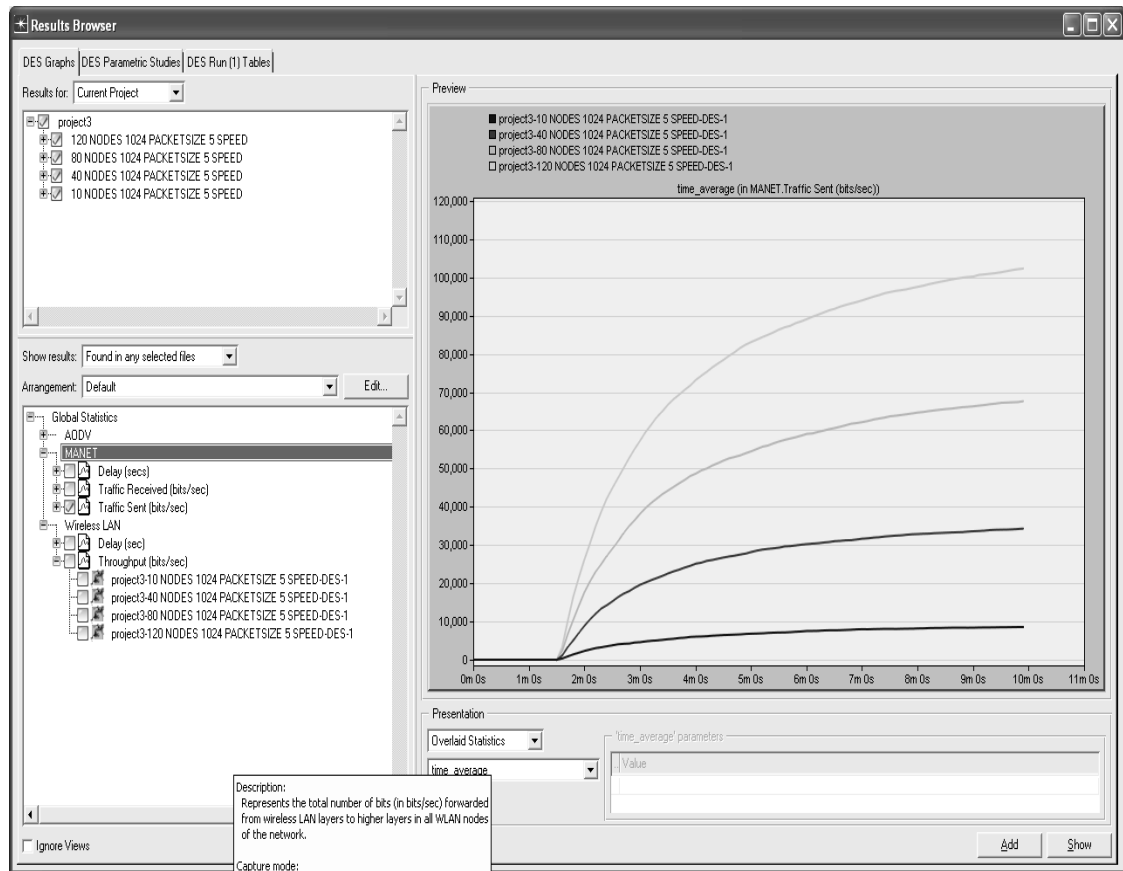


Figure 48

I: Manet_station configuration

As described before in Object Palette Tree, there is manet_station (Mobile Node). The manet_station node model represents a raw packet generator transmitting packets over IP and Wlan. After dragging it on the campus network and right click on the window to change the attribute to appear; as shown in Figure 49.

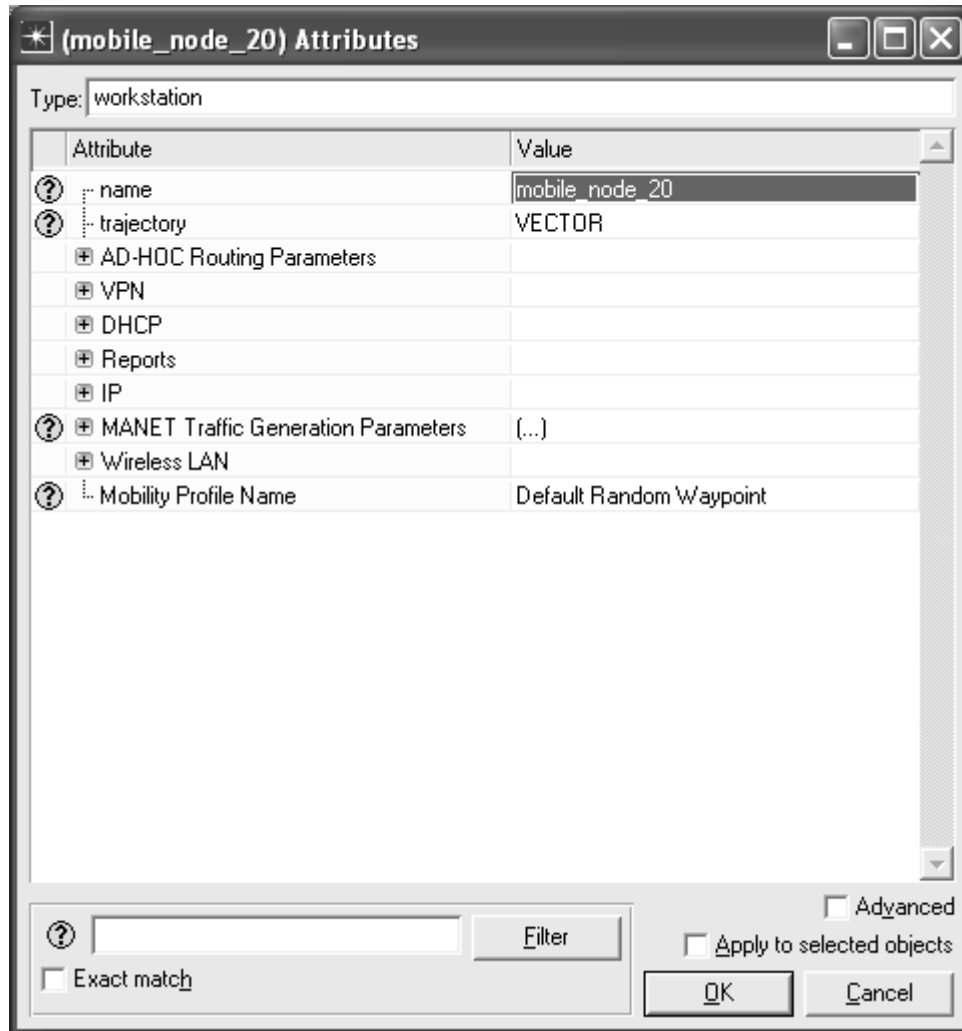


Figure 49

- 1- Set the Ad-hoc routing protocol as considered.
- 2- Set the Manet traffic pattern generation; start time 100, packets inter arrival time exponential 1 and file size to exponential 1024. Leave the rest as default. (Figure 50)
- 3- Notice that Tick apply to selected objects.
- 4- Also, depending on the scenario, it has wireless LAN options to change the parameters. In this scenario, it was selected as default.

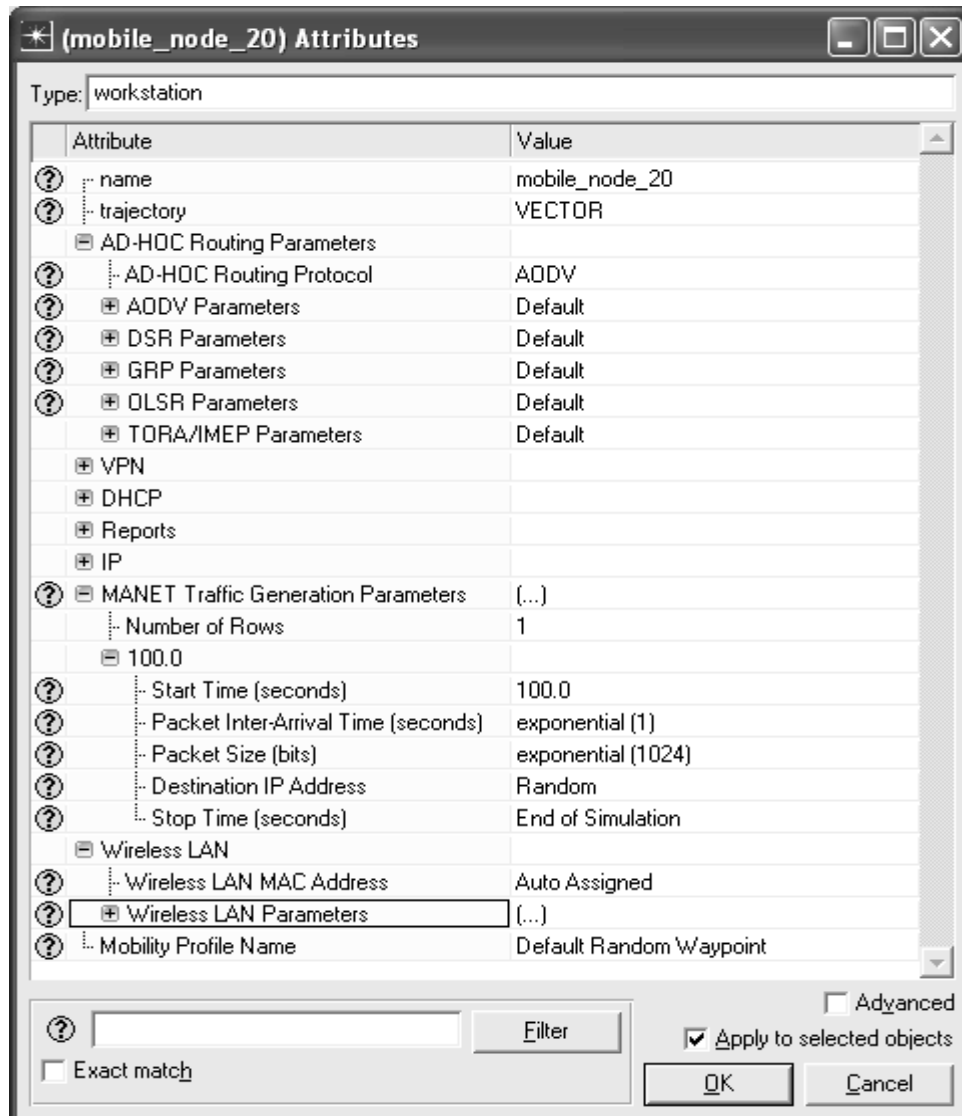


Figure 50