

# **Energy Consumption Analysis for Mobile Ad Hoc Routing Protocols**

**Belma Korkuter**

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  
May 2016  
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

---

Prof. Dr. Cem Tanova  
Acting Director

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

---

Prof. Dr. Işık Aybay  
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.

---

Assoc. Prof. Dr. Ahmet Rızaner  
Co-Supervisor

---

Assoc. Prof. Dr. Ali Hakan Ulusoy  
Supervisor

---

Examining Committee

1. Assoc. Prof. Dr. Muhammed Salamah

2. Assoc. Prof. Dr. Ali Hakan Ulusoy

3. Asst. Prof. Dr. Gürcü Öz

## **ABSTRACT**

The wireless network technology has been continued to become the most popular research field for a couple of years. Mobile Ad hoc Network (MANET), which is a type of a wireless network, is formed by mobile nodes. As is well known, the lifetime of a wireless network is restricted to battery limits of nodes located inside. Because of this reason, chosen routing protocol for a MANET is significant for the lifetime of the network. In this study, energy consumption of Ad hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Optimized Link State Routing (OLSR) and Destination Sequence Distance Vector (DSDV) protocols was analyzed with the performance metrics such as average energy consumption, reception energy consumption ratio, transition energy consumption ratio and idle energy consumption ratio under four different conditions, number of nodes, offered load, movement speed and topological area. It is shown with extensive simulation results that the table-driven protocols such as DSDV have better performance in terms of energy efficiency.

**Keywords:** MANET, AODV, DSR, OLSR, energy consumption

## ÖZ

Kablosuz ağ teknolojisi, günümüzde de akademik anlamda ilgi odağı konular arasında olmaya devam etmektedir. Bir kablosuz ağ çeşidi olan gezgin tasarsız ağlar (MANETs) gezgin düğümlerden oluşur. Bilindiği gibi, bir kablosuz ağın ömrü içinde barındırdığı düğümlerin batarya limitleriyle orantılıdır. Bu sebeple bir MANET için seçilen yönlendirme protokolü, ağın ömrü adına oldukça önemlidir. Bu çalışmada, Ad hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Optimized Link State Routing (OLSR) ve Destination Sequence Distance Vector (DSDV) protokollerinin enerji tüketimleri ortalama enerji tüketimi yüzdesi, alma modunda enerji tüketim yüzdesi, transfer modunda enerji tüketim yüzdesi ve boşa kalma modunda enerji tüketim yüzdesi olmak üzere dört farklı performans ölçütüne bağlı olarak düğüm sayısı yük miktarı, hareket hızı ve topoloji alanı olmak üzere dört farklı koşul altında analiz edilmiştir. Başta DSDV olmak üzere tabloya dayalı yönlendirme protokollerinin farklı senaryolarda enerji konusunda en iyi performansa sahip oldukları sonucuna varılmıştır.

**Anahtar kelimeler:** MANET, AODV, DSR, OLSR, DSDV, enerji tüketimi

To My Son, Zaman

## ACKNOWLEDGMENT

I would like to express my deepest appreciation to my supervisors Assoc. Prof. Dr. Ali Hakan Ulusoy and Assoc. Prof. Dr. Ahmet Rizer for their invaluable support throughout the preparation of this work. Without their motivation and guidance it would not have been accomplished. I feel indebted to them for their endless patience. Their kind wit, grace, and courtesy make them more of family members rather than just academic supervisors.

I would also like to very special thanks to my friend Erfan Amirzadeh Shams whose help and guidance for the simulation of this study. I will forever remain thankful for his contributions.

I can never thank my beloved husband Umut Korkuter enough who has shown limitless understanding, tremendous patience and unconditional love. He is always there for me whenever I need. I would also like to thank my family. They always show their pride to me. It is their praise, strong support, prayers and love that inspired me through this process.

Last but not least, I would also like to special thanks to my son, Zaman. He will be with us very soon. The duration and process of writing this thesis have been an exciting, inspiring, life changing and challenging journey while simultaneously waiting for the birth of him.

# TABLE OF CONTENTS

ABSTRACT .....	iii
ÖZ .....	iv
DEDICATION .....	v
ACKNOWLEDGMENT .....	vi
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF SYMBOLS AND ABBREVIATIONS .....	xi
1 INTRODUCTION .....	1
1.1 Thesis Aim.....	1
1.2 Thesis Significance.....	2
1.3 Thesis Contributions.....	3
1.4 Thesis Organization.....	4
2 BACKGROUND AND LITERATURE REVIEW.....	5
2.1 Routing Protocols for MANETs.....	5
2.1.1 Table-Driven Routing Protocols .....	5
2.1.2 On-Demand Routing Protocols.....	9
2.2 Literature Review about Energy Consumption Analysis for MANETs.....	12
3 ENERGY MANAGEMENT IN MANETS .....	15
3.1 Significance of Energy Management in the MANETs.....	15
3.2 Energy Conserving in the MANETs .....	16
3.2.1 Computation of Energy.....	17
4 SIMULATION ENVIRONMENT.....	20
4.1 Simulation Parameters .....	20

4.2 Performance Metrics .....	25
5 SIMULATION RESULTS .....	29
5.1 Number of Mobile Nodes Varying Scenario .....	29
5.2 Load Varying Scenario .....	35
5.3 Movement Speed Varying Scenario.....	38
5.4 Area Varying Scenario.....	42
6 CONCLUSION AND FUTURE WORK.....	46
6.1 Conclusion .....	46
6.2 Future Work.....	47
REFERENCES.....	48
APPENDICES .....	53
Appendix A: TCL File and Output Files .....	54
Appendix A.1: AODVenergy_10.tcl File.....	54
Appendix A.2: Example of Output Files .....	57
Appendix A.2.1: Example of Mobility File (movement10).....	57
Appendix A.2.2: Example of Traffic File (traffic10).....	61
Appendix B: AWK Script File.....	65
Appendix B.1: Example of AWK Script File (statisticsO10.awk).....	65
Appendix C: Sample Result for 10 MNs for AODV .....	68
Appendix C.1: Energyleft.txt File.....	68
Appendix C.2: Idle.txt File .....	68
Appendix C.3: Reception.txt File .....	68
Appendix C.4: Transition.txt File.....	68
Appendix C.5: Energy10.txt File.....	69



## LIST OF TABLES

Table 4.1: Simulation setup parameters .....	21
Table 4.2: The input parameters for ‘setdest’ .....	24
Table 4.3: Some performance metrics .....	26
Table 4.4: The energy trace formats .....	27

## LIST OF FIGURES

Figure 2.1: Pure flooding vs OLSR flooding .....	7
Figure 4.1: TCL code for random initial positions and destinations of MNs .....	23
Figure 4.2: The shell statement to create movement pattern file .....	23
Figure 4.3: A part of the trace file (out.tr) for energy identification.....	27
Figure 5.1: Effect of the number of MNs on average EC .....	30
Figure 5.2: Effect of the number of MNs on Rx EC ratio .....	31
Figure 5.3: Effect of the number of MNs on Tx EC ratio.....	32
Figure 5.4: Effect of the number of MNs on idle EC ratio .....	33
Figure 5.5: Effect of node density on total EC, EC in idle state, EC in Tx state and EC in Rx state .....	34
Figure 5.6: Effect of the offered load on average EC .....	36
Figure 5.7: Effect of the offered load on Rx EC ratio.....	37
Figure 5.8: Effect of the offered load on Tx EC ratio.....	37
Figure 5.9: Effect of the offered load on idle EC ratio .....	38
Figure 5.10: Effect of the movement speed on average EC.....	39
Figure 5.11: Effect of the movement speed on Rx EC ratio .....	40
Figure 5.12: Effect of the movement speed on Tx EC ratio .....	41
Figure 5.13: Effect of the movement speed on Idle EC ratio.....	41
Figure 5.14: Effect of the topological area on average EC .....	42
Figure 5.15: Effect of the topological area on Rx EC ratio .....	43
Figure 5.16: Effect of the topological area on Tx EC ratio.....	44
Figure 5.17: Effect of the topological area on Idle EC ratio.....	45

## LIST OF SYMBOLS AND ABBREVIATIONS

ABR	Associativity Based Routing
AODV	Ad hoc On-demand Distance Vector
AWK	Alfred Weinberger Kernighan
$B_C$	Channel Bandwidth
CBR	Constant Bit Rate
CORI	Caching Overheard Routing Information
DSDV	Destination Sequence Distance Vector
DSR	Dynamic Source Routing
EC	Energy Consumption
$EC_{idle}$	Energy Consumption in the Idle state
$EC_{Rx}$	Energy Consumption in Reception state
$EC_{tot}$	Total Energy Consumption
$EC_{Tx}$	Energy Consumption in Transmission state
ERAODV	Energy Reverse Ad hoc On-demand Distance Vector
$E_{Tx}$	Required Energy During the Transmission state
$E_{Rx}$	Required Energy During the Reception state
GSR	Global State Routing
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
MAC	Medium Access Control
MN	Mobile Node

MPR	Multipoint Relay
NS2	Network Simulator 2
OLSR	Optimized Link State Routing
$P_{idle}$	Power usage of the MNs in Idle state
$P_r$	Power usage of the Received packet
$P_{Rx}$	Reception Power
$P_t$	Power usage of the Transmitted packet
$P_{Tx}$	Transmission Power
RERR	Route Error
RREP	Route Reply
RREQ	Route Request
Rx	Reception
$S$	Packet Size
SAM	Simple Ad hoc Simulator
SN	Sequence Number
TBRPF	Topology dissemination Based on Reverse-Path Forwarding
TC	Topology Control
TCL	Tool Command Language
TORA	Temporally Ordered Routing Algorithm
$T_r$	Duration of the Time for Reception state
$T_t$	Duration of the Time for Transmission state
Tx	Transmission
UDP	User Datagram Protocol

# Chapter 1

## INTRODUCTION

This chapter provides an introduction to mobile ad hoc routing protocols in general and introduces the purpose, significance, contributions and the organization of this thesis.

### 1.1 Thesis Aim

The most rapidly developing segment of the network industry is wireless communications. Because of this growth, interactions between communications and computing have been increasing which change information access from “anytime and anywhere” into “all the time and everywhere” [1]. Wireless networks can be categorized based on the network architecture as infrastructure (e.g. cellular networks) and infrastructureless (e.g. Mobile Ad hoc Networks (MANETs)).

MANETs or sometimes called mobile mesh networks are collection of wireless Mobile Nodes (MNs) without using any centralized server or predetermined network infrastructure equipment support such as base station [1]. Each MN is free to move unpredictable direction and self-repair, and voluntarily transmits its own transmission [2]. When it is not practical economically or physically in wired network, this type of network is very beneficial.

MANETs have numerous different applications that have been used in many areas where cooperation of more than one MN is necessary, such as online conferences, mobile gaming etc. This cooperation process is known as routing [3].

Transmitting packets from the source to its destination by using the most efficient path [2] is a significant task in MANETs and many routing protocols have been standardized by Internet Engineering Task Force (IETF). Some of the most prominent protocols are Ad hoc On-demand Distance Vector (AODV) [4], Dynamic Source Routing (DSR) [5], Optimized Link State Routing (OLSR) [6], and Destination Sequence Distance Vector (DSDV) [7].

The lifetime of the wireless network greatly depends on limited battery power of the MN. Therefore, selection of routing protocol in MANETs is essential in order to achieve the maximum lifetime of the network. In this work, Energy Consumption (EC) for the abovementioned routing protocols of MANETs is analyzed and compared by using various scenarios in NS2 (Network Simulator 2) environment.

## **1.2 Thesis Significance**

Minimal usage of energy plays an important role in the world technology, especially in MANETs which contain scarce energy resources of MNs. Because of this constraint, EC within the network has to be reduced somehow.

EC is one of the most important performance metrics for MANETs, because it directly relates to the operational lifetime of the network. Energy management in MANETs can become lifesaving in some cases such as emergency situations. During an emergency situation, the first requirements are communication and data

exchanging for situational awareness. When the fixed communication network is not available, MANET is the only solution for this problem in emergency rescue operations (natural or man-made disasters) or military operations (military conflicts) [3]. MANET is restricted by limited battery power of MNs which makes energy usage a serious issue in such kind of situations. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed [8]. Additionally, MANET can be used in military operations as well. Military equipment contains computer equipment and in the battlefield it is needed for relaying and maintaining information related to situational awareness between soldiers, vehicles and military information headquarters [8, 9]. In such kind of situations, energy conservation has an important role which is critical to the success of the tasks performed by the network. Because of this reason, energy conservation should be considered carefully when designing and evaluating MANETs [10].

There is an increasing significance of energy management in MANETs. In order to achieve the most efficient power usage and power saving of MNs, routing protocols are selected according to the conditions.

### **1.3 Thesis Contributions**

There have been some works related with the EC of MANETs under distinct conditions. For instance, in one of the research [2], three different protocols which are AODV, DSR and OLSR are analyzed by using three different scenarios as load, node and speed varying. In this thesis one additional routing protocol (DSDV) and one more scenario (area varying) are analyzed. In order to have a good accuracy,

results are obtained by using an average of 50 runs for each representative scenario in this thesis.

The results of this study are expected to help the researchers to find other innovative methods for proposing more efficient routing protocols under different conditions.

## **1.4 Thesis Organization**

The thesis is divided into six chapters. Chapter 1 gives a brief introduction about MANET protocols proposing the EC analysis with the main aim, significance, contributions and the organization of the thesis. Chapter 2 reviews the related works supporting this thesis. Classifications of MANET protocols are examined and also principles of AODV, DSR, OLSR and DSDV are presented in Chapter 2 as well. In Chapter 3, energy management in MANETs is examined. Chapter 4 presents the environment of the simulation with performance metrics. This is followed by Chapter 5 which is about simulation results and their analysis. Finally, the discussions and conclusions of this this are presented in Chapter 6.



## Chapter 2

### BACKGROUND AND LITERATURE REVIEW

MANET protocols and traditional classification of them are considered in this chapter. Some well-known routing protocols which are AODV, DSR, OLSR and DSDV are presented in details. Furthermore, review of the literature which supports this thesis is given as well.

#### 2.1 Routing Protocols for MANETs

Routing is the process of forwarding data to the destination by using most optimal path [2]. In order to evaluate this optimal path, different routing protocols use different measurements such as the number of hops (e.g. OLSR) and delay (e.g. AODV). According to the needs of the application, one of the routing protocols is selected. Depending on routing strategy, routing protocols can be classified as proactive (table-driven) or reactive (on-demand).

##### 2.1.1 Table-Driven Routing Protocols

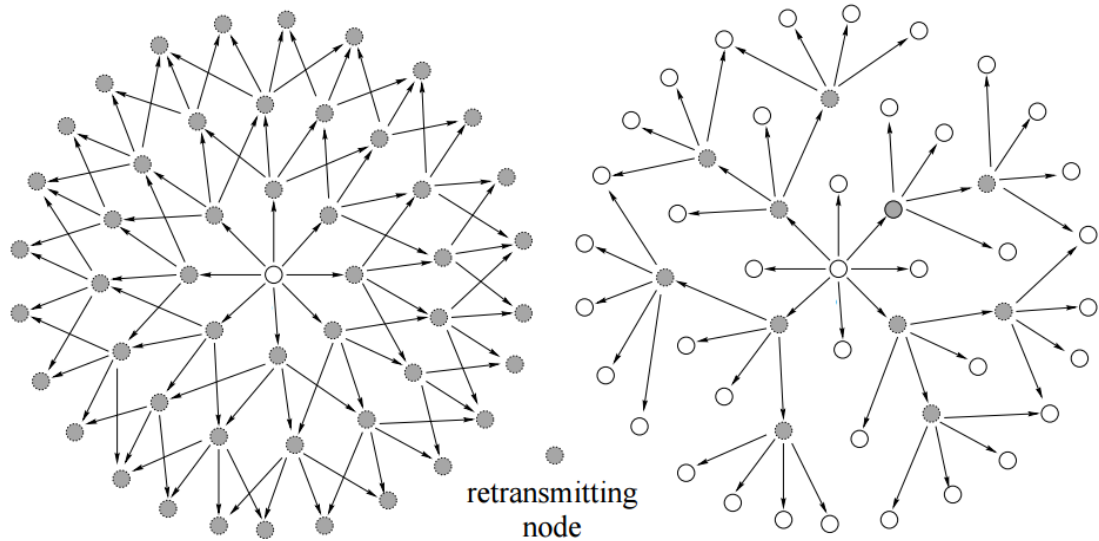
In table-driven routing protocols also called proactive protocols all feasible routes between all source destination pairs are known. The most significant characteristic of proactive protocols is that the routing information is immediately available before use [3]. Each MN inside the network periodically announces update routing information to other MNs in order to compute the optimal path [2]. MNs maintain their routing tables according to these propagations regularly. This information is sent even if no change in topology has occurred or even it is not needed [11]. Since routing information is used in all conditions and also due to pure flooding, it

generates high bandwidth overload and lots of routing control overhead [11]. Because of these reasons, it is not preferred for some networks which have high mobility and a large number of MNs environments [2, 11]. Topology dissemination Based on Reverse-Path Forwarding (TBRPF), OLSR, DSDV and Global State Routing (GSR) are some examples of table-driven routing protocols.

#### **2.1.1.1 Optimized Link State Routing Protocols**

OLSR is one of the table driven protocols designed for MANETs. Because of its table-driven nature, the information about all routes is available before use. Each MN in the network transmits the status of the links, in order to discover its 1-hop and 2-hop neighbors by exchanging “Hello” messages that contain a list of neighbors and its own address at every two seconds as default [2].

Instead of using pure flooding which is not preferred in high MN density and high mobility networks [2], behind of this routing protocol, there is a different strategy to broadcast packets. The Multipoint Relay (MPR) mechanism which minimizes the overhead of control traffic is used in OLSR [12]. MNs select a set of their 1-hop away neighbors as MPRs which help to provide the optimal path in terms of shortest path and also prevent redundant transmissions [12]. Only these MNs can forward Topology Control (TC) messages that contain information about the topology of the network at every five seconds by default [2] to others.



(a) Pure flooding

(b) OLSR flooding

Figure 2.1: Pure flooding vs OLSR flooding [12].

The pure flooding algorithm and the MPR mechanism of OLSR are shown in Figure 2.1 [12]. The darker circles indicate retransmitting MNs. In pure flooding or blind flooding, the broadcast packet is forwarded by every MN in the network which causes huge number of collisions especially in higher mobility scenarios [2, 12]. On the other hand, the broadcast packet is received by all 1-hop neighbors, but only be forwarded by the MPRs to all the 2-hop neighbors in OLSR flooding [2].

As can be seen from Figure 2.1, a broadcast message of MN which is from the center is propagated using pure flooding and it takes 49 retransmissions in order to reach up to only 3-hops. On the other hand, for OLSR flooding it takes 11 retransmissions by using the MPRs [12].

Undoubtedly using MPR mechanism reduces flooding, so it provides less bandwidth and power consumption of MNs are reduced. As a result, the overall network

performance is improved. Because of this reasons, OLSR can be preferred in large and intensive mobile networks [1].

### **2.1.1.2 Destination Sequence Distance Vector Protocol**

DSDV is another example of proactive routing protocols which is one of the first protocols designed for MANETs [7]. In this protocol, each MN within the network maintains the routing table that contains a list of the next hop neighbor to all destination MNs, the number of hops to the destination MNs and Sequence Numbers (SNs) [1]. These SNs are generated by the destination MNs and used to eliminate the stale routes, so this prevents the route loops. Each MN increments and adds its SN when propagating its own routing table and routes that have greater SNs are preferred [1]. If the update is received which has same SN, the update which has least hop count is preferred. Routing information is exchanged between MNs which are immediate neighbors, but not flooding. In other words the route announcements are not rebroadcasted.

DSDV needs regular updates for its routing tables. This can be done in terms of either periodic updates (time driven) or triggered updates (event driven). Each MN periodically announces its updates to its all neighbors every fifteen seconds as default [7, 13]. If a MN does not receive any update from one of its neighbors in three periodic update intervals which is forty five seconds, this MN assumes that its neighbor is lost [13]. On the other hand, when an important change is occurred in between the periodic updates, the MN may broadcast this changed routing table as the event-triggered.

### **2.1.2 On-Demand Routing Protocols**

In on-demand routing protocols also called reactive protocols routes are discovered within the network when there is a demand. If there is no communication between two MNs in order to transmit or receive the data, it is not needed to keep routing information of these two MNs. Route discovery process is started at any time, if a MN in the network has a packet to be transmitted. In this process, the source MN propagates Route Request (RREQ) to the destination. When RREQ is received by the destination, Route Reply (RREP) is sent to the source [2]. The source learns the optimal path by using this RREP message. Once a route between two MNs is discovered, it is continued until the destination MN becomes unachievable or the route is not necessary anymore. Reactive protocols require some delay because of this route discovery process. Temporally Ordered Routing Algorithm (TORA), DSR, AODV, and Associativity Based Routing (ABR) are examples of the reactive type of protocols.

#### **2.1.2.1 Ad hoc On-Demand Distance Vector Routing Protocol**

AODV is one of the well-known and widely used reactive protocols. The route discovery process is started if and only if a MN in the network wants to send a packet and the information about the route of destination MN is unknown [3, 14]. In order to find a route to the destination, the source generates RREQ to its neighbors within its wireless transmission range and they flood this request to other intermediate MNs respectively. Until the packet reaches the destination or an intermediate MN that has a valid route to the destination, this process continues [3].

Each RREQ contains the address of the source and broadcast id and the combination of them are the identifier of this RREQ [1, 15]. The broadcast id that is also called

request id is increased one by one when another RREQ is broadcasted from the same MN [1]. The RREQ includes the destination address, SNs of source and destination to prevent routing loops and provide routing freshness and hop count which represents the number of hops to arrive the destination as well [1, 15].

When RREQ arrives, a RREP is broadcasted by the destination or any intermediate MN that has a fresh route which has a destination SN greater than or equals the one in the corresponding RREQ to the source MN through the reverse path [1, 15]. This path is automatically set up by recording the address of neighbor which it received the first copy of RREQ [4]. RREP contains addresses of the source and destination, the sequence of destination, hop count and also life time which represents expiration or deletion time of the route [1, 15]. When the source receives RREP, the forward route is established in order to start sending data to the destination MN and also optimal path is selected as the lowest latency path [2] without considering hop count in this protocol.

Because of the mobility, the link can be broken. If an intermediate MN or destination MN moves, the neighbor of this MN can sense this movement. If a MN does not receive the Hello message from one of its neighbors, it generates Route Error (RERR) to report this failure to the source and if the route is still necessary, the route discovery process may be restarted [1, 14].

#### **2.1.2.2 Dynamic Source Routing Protocol**

DSR is another reactive protocol. Although it is similar to AODV, the route discovery processes are different [2]. In place of the forward route and reverse route, the DSR uses source route for both discovery and maintenance of routes which are

working together [2, 16]. The source knows the complete hop-by-hop route which is stored in a route cache to the destination and each MN in the network needs to utilize this information of route cache [1]. The source MN learns not only the optimal path in this protocol but also other alternate paths to the destination [17]. It can be useful if the optimal path in terms of the lowest latency path fails.

When a MN has a packet to transmit, first it checks its route cache [3, 16]. If any route to the destination is found, the source MN uses this route otherwise it starts the route discovery process. A RREQ is broadcasted and received by all neighbors respectively. Each intermediate MN keeps the record of the route as a list which includes itself as well [18]. The MN adds itself to the list until the destination and RREP with this list is received by the source [5, 18]. The source can learn routes to each intermediate MN which can also learn routes to all other MN on the route via using a single RREQ-RREP cycle [16]. The feature of Caching Overheard Routing Information (CORI) [5] is used in DSR, in order to store routing information overhearing in control overhead in the cache of MNs [2]. On the other hand, for each destination, only one entry is allowed in routing table [17] of AODV protocol and it can provide limited routing information as compared with DSR [19].

In AODV, the stale routes are replaced with fresher ones by using their SNs and if encountered with different options, the fresher route which has the greatest value of destination SN is chosen [17]. Furthermore, if a route is not recently used, this route is expired. DSR does not have such kind of mechanism to clear stale routes or prefer fresher routes [17]. Because of this deficiency, the stale routes can appear in DSR.

However, the approach of the AODV is not perfect as well, because the valid routes can be erased due to non-utilization of these routes.

Additionally in DSR, any routing advertisement such as Hello messages to detect neighbor is not used, so it provides less bandwidth overhead and conserved energy [14]. In AODV, if Hello message from one of the neighbors of a MN is not received, it generates RERR which contains information of the failed route [4]. However in DSR, a failed route is not appeared and erased before wanted to be used.

## **2.2 Literature Review about Energy Consumption Analysis for**

### **MANETs**

There have been some studies about performance comparison of EC for existing MANET routing protocols [2, 20-27]. In the first instance, EC was used as a metric in NS2 environment to evaluate AODV, DSR, TORA and DSDV in [20] by using five distinct parameters which were the number of MNs, area size, mobility patterns, the number of real traffic sources and pattern of the data traffic. Also, these aforementioned protocols were analyzed by taking EC metric into consideration in [21] by varying the size of the network, constant speed, the pause time of MNs and high speed floating model respectively. According to the both researches, TORA consumes more energy than others under all conditions while the ECs of AODV, DSR and DSDV are very close each other.

These four routing protocols (i.e. AODV, DSR, TORA and DSDV) were analyzed in NS2 environment in [22]. The study conducted was based on some performance metrics such as network lifetime and system lifetime. Network lifetime is the time of a MN in the network that consumes its all power while system lifetime means 20%



of MNs within the network consuming all their power. In this research, packet delivery ratio, routing overhead and end-to-end delay were used as well. The authors used two different parameters which were the number of MNs and pause time. According to the research of [22], DSDV shows better performance in scenarios that related with ECs (i.e. network lifetime and system lifetime) and TORA has the worst performance in these mentioned scenarios among the other compared routing protocols.

In [23], AODV, DSR and DSDV routing protocols with respect to their ECs in different conditions, including node density and speed varying were evaluated and compared with two performance indexes which were total EC and EC depending on the packet type (i.e. MAC, data and routing). The results illustrate that DSDV consumes less energy. Although AODV and DSR are close each other in speed scenario, AODV consumes more energy in both speed and node density scenarios.

In [24], AODV, DSR, Link State and also classic Dijkstra were compared in Simple Ad hoc Simulator (SAM) environment. The authors used different performance metrics, including throughput, average delay, delivery fraction and energy. Especially in moderate load, AODV and DSR have better results with respect to the research. However, simple Link State performs well in heavy load condition.

Moreover, EC of AODV and DSR were also compared in [25], by using NS2 environment. Although both routing protocols have same reactive behavior, the simulation results show that DSR uses less amount of energy under different

scenarios such as traffic source, speed, the number of MNs, the pause time and area size as well.

In [26], EC of AODV, DSDV and a new proposed protocol namely Energy Reverse Ad hoc On-demand Distance Vector (ERAODV) were compared by increasing the number of nodes. The authors explain that DSDV increases the network lifetime compared to AODV, but argue that ERAODV is better than DSDV.

Furthermore in [27], total EC and other metrics such as packet delivery ratio, throughput and the average delay of AODV, DSDV and OLSR were evaluated. The authors explain that AODV consumes more energy under all conditions. On the other hand, the EC of the DSDV is slightly more efficient than OLSR.

In [2], AODV, DSR and OLSR were compared according to their ECs using NS2 environment. The author utilized some parameters including the number of MNs, speed and load. The simulation results illustrate that AODV and DSR are very close to each other under load varying and node density scenarios. On the other hand, OLSR has better performance among the other compared routing protocols.

## Chapter 3

### ENERGY MANAGEMENT IN MANETS

MANET enables untethered communication. In spite of this, it requires portable power sources which are constrained resources such as batteries. This chapter focuses on the questions, why energy management is a crucial issue in the MANETs and how to manage energy in an efficient manner. Additionally, the computation of energy in different states is presented in this chapter as well.

#### **3.1 Significance of Energy Management in the MANETs**

MANET is comprised of the MNs which are dependent on scarce resources and is often preferred especially in emergency situations. In such cases, it may be difficult to charge or replace battery sources. In order to be able to prolong the existence of the overall network, energy management is a must. The energy requirement will increase continuously, in spite of the significant improvement concerning power capacity of the batteries. The issue is not to supply the MNs with advanced batteries, but to preserve and utilize the current batteries efficiently. Energy management has many different meanings. In this study it relates to saving power of the MNs within the network.

Although a MANET does not need the network construction or management, as compared with wired networks or other wireless networks which have fixed base stations or access points, power management in the MANET is more challenging because of the lack of an infrastructure. The networks with infrastructure provide

energy management more efficiently due to the fact that all traffic for the MNs has to go through the specific access points. Each MN in the MANET acts as a router and helps forwarding data which makes the traffic load heavier than other types of networks. Additionally, the absence of centralized administration causes making energy management in a distributed way which is hard to attain [1].

The MANET is very popular research area and many different types of protocols have been proposed based on different perspectives. Various protocols have been proposed in order to satisfy different features such as efficient bandwidth, distributed implementation, optimization of throughput etc. Thanks to the growth of wireless devices, the wireless network is being used much more frequently and due to the energy conserving techniques have gained importance.

### **3.2 Energy Conserving in the MANETs**

Due to the importance of energy conserving in the MANETs, different solutions need to be used in order to cope with the scarcity of power. The most prominent of these solutions can be proposing new energy-efficient routing protocols, or choosing existing routing protocols according to the current network scenario.

Some of the power conservation methods that may be preferred for new routing protocols prevent unnecessary and repeated retransmission within the network, putting the MNs into sleep state instead of putting into idle state when they are not transmitting or receiving as fast as possible, decreasing frequency of sending control packets, reducing control overhead in order to find the optimal path and data transmission and so on [1]. In order to ensure the network makes use of efficient

energy and has a long lifetime, for new routing protocols these techniques can be applied individually on their own or in a combined manner.

Some parameter changes in the current network heavily affect the performance of the routing protocol in terms of energy. As mentioned earlier, network should select a routing protocol according to the network conditions in order to attain the most efficient energy usage of the MNs. The MANET is designed to be scalable. So the network can expand or some other parameters may be changed such as the density or speed of MNs, traffic source, pause time or topological area in the network which force protocols to perform differently.

The studies that were mentioned in Chapter 2 were evaluated and the results of these studies discussed how the energy of the MNs behaves according to the existing protocols under different conditions.

### **3.2.1 Computation of Energy**

A MN within the network consumes its power in different states which are transmission, reception, idle and sleep. If a MN wants to send a packet to other MNs, which is in the ‘transmission state’, or a MN receives a packet from another MN in the ‘reception state’, transmission (Tx) and reception (Rx) energy are required respectively. The MNs in the network require the largest amount of energy in Tx state and this is followed by Rx state.

The required energy during the Tx state ( $E_{Tx}$ ) and Rx state ( $E_{Rx}$ ) are calculated respectively by the following formulas [23]

$$E_{Tx} = (P_{Tx} \times S) / B_C \quad (3.1)$$

$$E_{Rx} = (P_{Rx} \times S) / B_C \quad (3.2)$$

where  $P_{Tx}$  and  $P_{Rx}$  are transmission power and reception power which are power needed to have the specific distance in Watts respectively. These values and the packet size in bits ( $S$ ) are divided by the channel bandwidth which is represented as  $B_C$ .

The EC in Tx state ( $EC_{Tx}$ ) and Rx state ( $EC_{Rx}$ ) are calculated by using following formulas [28]

$$EC_{Tx} = P_t \times T_t \quad (3.3)$$

$$EC_{Rx} = P_r \times T_r \quad (3.4)$$

At the beginning of the communication, each MN has an initial level of energy in Joules which is decreased after each data exchanging. If the energy level of a MN runs out, this MN cannot attend any packet transmission or reception. Additionally for every transmitted and received packet the power usages are defined as  $P_t$  and  $P_r$  (in Watts) respectively. The duration of the time for Tx state ( $T_t$ ) and Rx state ( $T_r$ ) which are related to size of the packet are other parameters needed to calculate  $EC_{Tx}$  and  $EC_{Rx}$ .

Even a MN does not transmit or receive any packet, but is ready for communication in the 'idle state', known as 'waiting state', also requires energy. The EC in the idle state is less than in Tx state and roughly same as in Rx state [1]. On the other hand, the MN does not wait for any packet which is in the 'sleep state', better known as

‘power saving state’; this MN also needs some energy. However, this amount of energy is very few compared to other states, so it can be negligible [28].

The EC in the idle state ( $EC_{idle}$ ) is calculated by using following formula [28]

$$EC_{idle} = P_{idle} \times T_{idle} \quad (3.5)$$

where  $P_{idle}$  (in Watts) is the power of the MN while it is in the idle state. In this mode, the MNs are idle, but still listen to the control messages.

Additionally, total EC ( $EC_{tot}$ ) is computed as follow [28]

$$EC_{tot} = EC_{Tx} + EC_{Rx} + EC_{idle} \quad (3.6)$$

In summary,  $EC_{tot}$  depends on the power needed for transmitting and receiving a single packet, power used in idle state and duration of the time for each state. Other parameters of the energy model, which is used in the NS2 simulation environment, are presented in Chapter 4.

## Chapter 4

### SIMULATION ENVIRONMENT

The parameters of the energy model and also the performance metrics which are used in order to obtain the simulation results are presented in this chapter.

#### 4.1 Simulation Parameters

In order to investigate the performance of the routing protocols (AODV, DSR, OLSR and DSDV) in terms of EC, different parameters are selected in this study. These parameters are the number of MNs, offered load, mobility speed and topological area. The main simulation parameters according to the different scenarios are illustrated in Table 4.1.

The NS2 simulator includes the implementation of AODV, DSR and DSDV routing protocols, but a patch for OLSR is integrated into the NS2 for this study. In order to setup the components of the MANET, the Tool Command Language (TCL) code has been written as shown in Appendix A.1.

The Medium Access Control (MAC) protocol is selected as IEEE 802.11. Under physical layer setting, the radio propagation model is selected as two ray ground, the channel type is wireless and the Omni-directional antenna which is the only supported model of the antenna in the NS2 is used.



As mentioned in Chapter 3, each MN has an initial energy at the beginning of the simulation which is set as 1000 J in this study and the antenna transmits power which is for the transmission range of 250 m is set by the value of  $P_{Tx}$  as 0.281838 W. The values of  $P_t$  and  $P_r$  are the given usage of power for each packet that it transmits and receives and are assigned as 1.4 W and 1.0 W respectively. For the parameters that are not declared in Table 4.1 the default values are used.

Table 4.1: Simulation setup parameters.

Parameters	Number of MN Varying	Offered Load Varying	Movement Speed Varying	Topological Area Varying
Topological Area	1000×1000 m <sup>2</sup>			500×500 - 1000×1000 m <sup>2</sup>
MAC Layer Protocol	IEEE 802.11			
Radio Propagation Model	TwoRayGround			
Antenna Type	OmniAntenna			
Channel Type	Wireless Channel			
Channel Bandwidth ( $B_C$ )	2 Mbps			
Packet Size ( $S$ )	512 bytes			
Initial Energy	1000 J			
Tx Range	250 m			
Number of MNs	10-60 nodes	50 nodes		
Speed	2 m/s	2-30 m/s	2 m/s	
Pause Time	0 s			
Traffic Type	CBR			
Packet Type	UDP			
Number of Source-Destination Pairs	10 connections			
Offered Load	100 kbps	25-150 kbps	100 kbps	
txPower ( $P_t$ )	1.4 W			
rxPower ( $P_r$ )	1.0 W			
idlePower ( $P_{idle}$ )	0.32 W			
sleepPower	0 W			
Simulation Time	300 seconds			

In this study, MNs have never been in the sleep state, so the required power in this state is defined as 0.  $B_c$  which is also mentioned in Chapter 3 is defined as 2 Mbps and  $P_{idle}$  is 0.32 W.

Additionally, the pause time which provides the duration of time of MNs stop before moving again is 0, which means they never stop, and the duration of simulation is 300 seconds in each simulation.

Although the NS2 provides the MNs to move toward the specific coordinate from the given position with the specific speed at a certain time period, entering all initial positions and movements for MNs in the network can be hard, especially if the network contains many MNs. Also the simulation may not provide the reliable results. In an accurate mobility model, each MN has a random initial position which is selected inside the simulation area and moves for the random time, reaches the random destination point, if it is necessary waits for the certain time in second which is known as pause time and then continues to move toward another random target position [21].

In this implementation, the MNs start to move from their initial position which is selected randomly with the specific constant speed in each scenario and their target positions are decided randomly as well to get an accurate simulation result.

In order to define the starting location and the target destination of the MNs randomly in  $1000 \times 1000 \text{ m}^2$  of the square area at the constant speed of 2 m/s, the code in Figure 4.1 is used.

```

#Provide initial location of mobilenodes
#Destination setting procedure

$ns_ at 0.0 "destination"
proc destination {} {
    global ns_ val node_
    set time 50.0
    set now [$ns_ now]
    for {set i 0} {$i < $val(nn)} {incr i} {
        set xx [expr rand()*1000]
        set yy [expr rand()*1000]

        $ns_ at $now "$node_($i) setdest $xx $yy 2";
    }
    $ns_ at [expr $now + $time] "destination"
}

```

Figure 4.1: TCL code for random initial positions and destinations of MNs.

The file of the movement pattern scenario is created by using a shell utility which was developed in C++ called ‘setdest’ as shown in Figure 4.2.

```
setdest -v 2 -n 10 -p 0.0 -s 2.0 -m 2.0 -M 2.0 -t 300 -x 1000 -y 1000 > movement 10
```

Figure 4.2: The shell statement to create movement pattern file.

This shell utility contains two versions. The first (original) version which is the default in NS2 had an issue. If the picked random speed is very small for a MN, this MN may not reach the target destination point. This version provides to define the maximum movement speed, but the minimum speed is 0. In order to cope with this problem the second version which is modified by J. Yoon from University of Michigan [29] is preferred. The second setdest version provides to define the minimum speed and also some other additional features as compare with the original version such as speed type and pause type.

The input parameters shown in Figure 4.2 for ‘setdest’ are explained in Table 4.2. It creates the mobility for 10 MNs with a pause time of 0, the maximum and the minimum speed is 2 m/s as constant and normally distributed. The simulation time is 300 s and the area is 1000×1000 m<sup>2</sup>.

Table 4.2: The input parameters for ‘setdest’.

<b>Parameters</b>	<b>Definition</b>
-v	version of the setdest
-n	number of the MNs in the network
-p	pause time between two movements of a MN
-s	speed type which can be uniform (1) or normal (2) distributions
-m	minimum speed of the MNs within the network
-M	maximum speed of the MNs within the network
-t	time of the simulation
-x	x coordinate of the simulation topography
-y	y coordinate of the simulation topography
movement10	generated file name

The generated file for a case of 10 nodes, which is given in Appendix A.2.1, is generated as the file of the movement pattern scenario and this file is sourced into the TCL script file.

In order to configure movement pattern and the traffic in the TCL file, there are two steps:

Step 1: Declare the movement as variable “set val(mv) "movement10"”

Declare the traffic as variable “set val(tfc) "traffic1"”

Step 2: Source these variables by reference “source \$val(mv)” “source \$val(tfc)”

In order to create the traffic model, User Datagram Protocol (UDP) in the protocol of transport layer and Constant Bit Rate (CBR) for data traffic are used. UDP is

connectionless Internet protocol and preferred for the applications which require fast and efficient transmission. CBR provides to send the data packets with the same rate continually.

The structure of traffic basically depends on two parameters which are packet size and data rate. The parameter of CBR is defined as the packet size ( $S$ ) of 512 bytes and the data rate (offered load) is changed according to the simulation scenarios.

The source and destination pairs are defined in 'trafficX' files as shown in Appendix A.2.2 which are written differently according to the simulation scenarios and the total connection is 10 pairs in the implementations. Other MNs in the network which are not in the source or destination pairs are known as intermediate MNs.

Additionally, other parameters such as topological area, the number of MNs within the network, constant movement speed of the MNs and the offered load are changed depending on the simulation scenario as given in Table 4.1.

## **4.2 Performance Metrics**

In order to measure, evaluate and compare the performance of the mentioned routing protocols in terms of energy, some metrics are considered for the implementation.

Some of these performance metrics are extracted directly from the generated simulation trace files and shown in Table 4.3 and some of them require calculations by using these metrics.

Table 4.3: Some performance metrics.

<b>Metrics</b>	<b>Definition</b>
Energy Left	Remaining energy level of the MNs in Joules
Idle Energy Consumption	Consumed energy of the MNs in idle state in Joules
Tx Energy Consumption	Consumed energy of the MNs in Tx state in Joules
Rx Energy Consumption	Consumed energy of the MNs in Rx state in Joules

Other performance metrics, which require some calculations, are Average EC, Idle EC Ratio, Tx EC Ratio and Rx EC Ratio.

The average EC is calculated as:

$$\text{Average EC} = \frac{(\text{Total number of MNs} \times \text{Initial Energy}) - \text{Total Energy Left}}{\text{Total number of MNs}} \quad (4.1)$$

Idle EC Ratio is the percentage of EC of the MNs in the idle state in proportion to total EC which is computed as:

$$\text{Idle EC Ratio (\%)} = \frac{\text{Total Idle EC}}{(\text{Total number of MNs} \times \text{Initial Energy}) - \text{Total Energy Left}} \times 100 \quad (4.2)$$

Tx EC Ratio is the percentage of EC of the MNs in the Tx state in proportion to total EC which is computed as:

$$\text{Tx EC Ratio (\%)} = \frac{\text{Total Tx EC}}{(\text{Total number of MNs} \times \text{Initial Energy}) - \text{Total Energy Left}} \times 100 \quad (4.3)$$

Rx EC Ratio is the percentage of EC of the MNs in the Rx state in proportion to total EC which is computed as:

$$\text{Rx EC Ratio (\%)} = \frac{\text{Total Rx EC}}{(\text{Total number of MNs} \times \text{Initial Energy}) - \text{Total Energy Left}} \times 100 \quad (4.4)$$

The trace files (out.tr) which are produced by the TCL scripts contain all data about the performance metrics as in Table 4.3 that can be analyzed and obtained from the trace files by using Alfred Weinberger Kernighan (AWK) script file which is given in Appendix B.1. Figure 4.3 is an example part of the trace file and shows the format of the energy representation.

```
N -t 45.902382 -n 26 -e 963.873831 -ei 5.406556 -es 0.000000 -et 5.993452 -er 24.726161
```

Figure 4.3: A part of the trace file (out.tr) for energy identification.

This format begins with the abbreviation of N which represents the energy of the MNs. Other abbreviations of the trace files on energy are defined in Table 4.4.

Table 4.4: The energy trace formats.

<b>Abbreviation</b>	<b>Definition</b>
-t	Time
-n	Name or id of the MN
-e	Total remaining energy level
-ei	EC in idle state
-es	EC in sleep state
-et	EC in Tx state
-er	EC in Rx state

As can be seen in Figure 4.3, the MN of 26th consumes 5.406556 J of energy in idle state, 5.993452 J of energy in Tx state and 24.726161 J of energy in Rx state. Totally it drains its 36.126169 J energy out of 1000 J which is the initial energy of the MN. So the total remaining energy level of this MN is 963.873831 at 45.902382 simulation time.



## Chapter 5

### SIMULATION RESULTS

This chapter presents the results of the simulations which are obtained by using the average of 50 runs for each part of all scenarios. Providing the accuracy of the simulation results highly depend on recurrence. Thus, these simulation results are the consequence of 1,250 correct runs using NS2. As explained in Chapter 4, after executing the simulation for each part of scenario the data from the trace files (out.tr) is obtained with the AWK files.

As mentioned before, four different scenarios which are the number of MNs, load, speed and area are analyzed in terms of EC for different routing protocols in this study. Figures 5.1 – 5.16 represent the comparison between all mentioned protocols on performance metrics. Among these performance metrics used in developing the graphs, the equation for calculating Average EC in Joule is depicted in 4.1 and demonstrated in Figures 5.1, 5.5, 5.9 and 5.13 for each scenario. In order to present the energy distribution ratio in each state (Tx, Rx and idle), the other performance metrics (Tx EC ratio, RX EC ratio and idle EC ratio) are given as percentage in the graphs (Figures 5.2 - 5.8, 5.10 - 5.12) and presented in 4.2 – 4.4.

#### **5.1 Number of Mobile Nodes Varying Scenario**

Different number of MNs is used in MN varying scenario and results are presented in Figures 5.1 - 5.4, in order to observe the effect of MN density in the simulation. For this scenario, the number of MNs is started from 10 MNs and increased to 60 MNs,

the offered load is set as 100 kbps, the movement speed of the MNs is 2 m/s, the area is  $1000 \times 1000 \text{ m}^2$  and the other parameters are given as in Table 4.1.

The average EC is calculated as discussed in 4.1 for all routing protocols and demonstrated in Figure 5.1. It is shown that when the MN density in the network increases, more EC in all routing protocols occurs.

In the maximum MN density environment, which contains 60 MNs, DSDV consumes the least energy among the other protocols with 223 J. Another table-driven protocol OLSR follows DSDV with 245 J. AODV and DSR, which fall into the same routing category, are much closer each other and their ECs are 257 J and 262 J respectively.

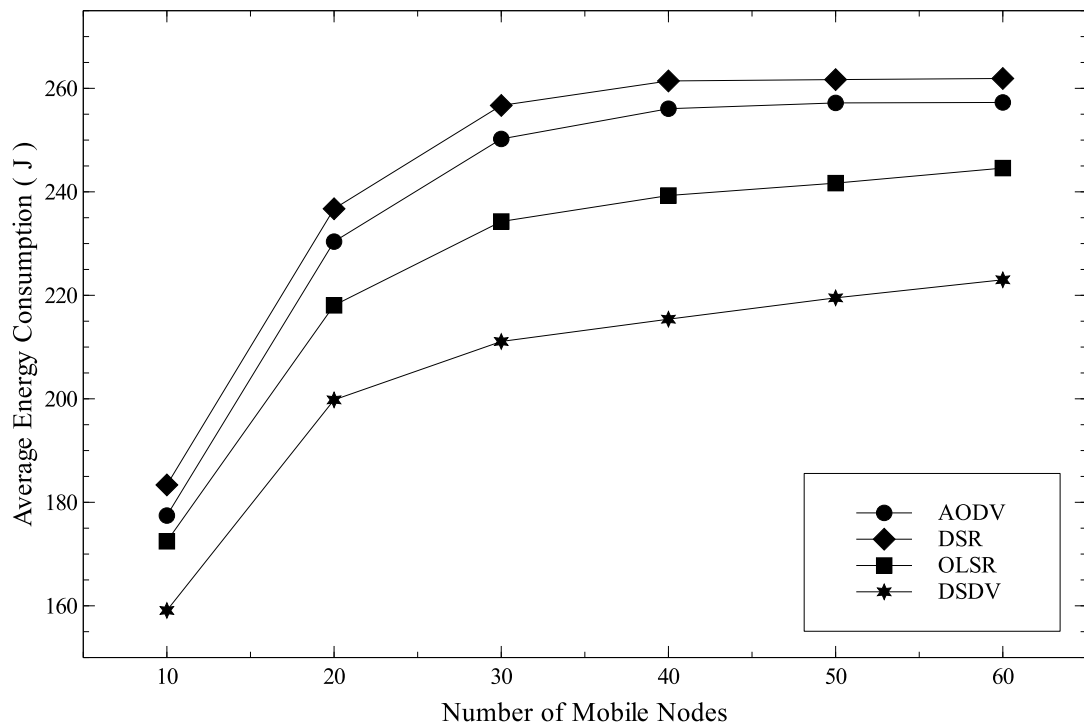


Figure 5.1: Effect of the number of MNs on average EC.  
(load:100 kbps, speed: 2 m/s, area:  $1000 \times 1000 \text{ m}^2$ )

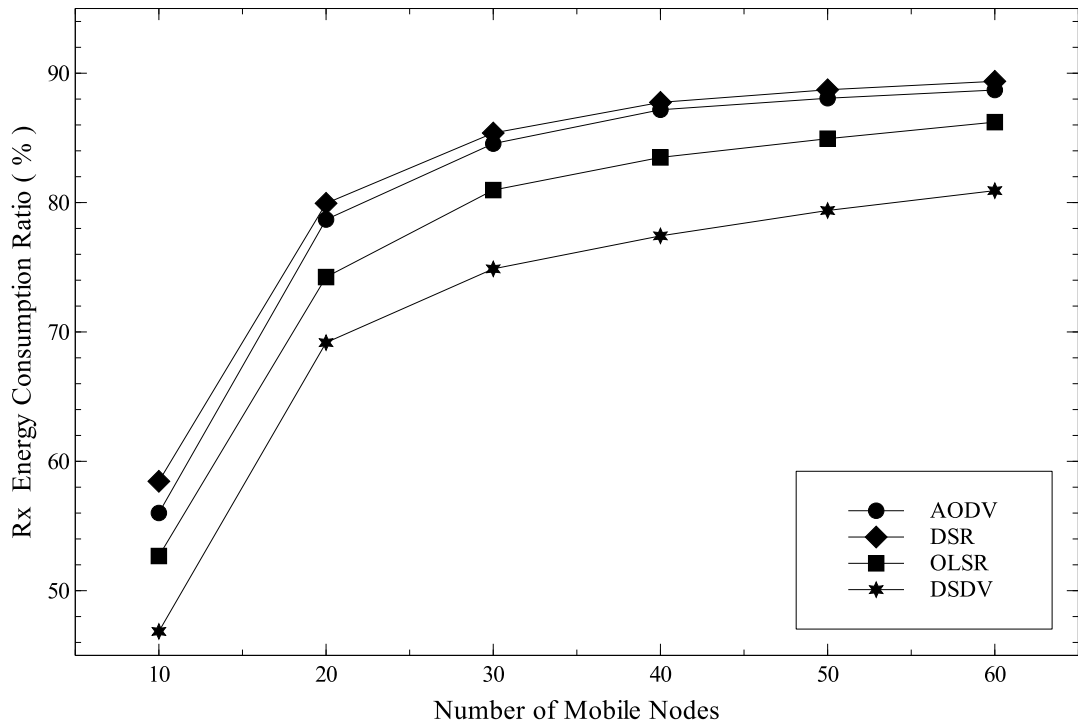


Figure 5.2: Effect of the number of MNs on Rx EC ratio. (load:100 kbps, speed: 2 m/s, area: 1000×1000 m<sup>2</sup>)

According to Rx EC ratio graph as illustrated in Figure 5.2, when the MN density between source and destination increases, the MNs spend more time in Rx state, because these intermediate MNs change their state to Rx mode continuously.

Receiving the data packet process contains two types of activities. These are receiving the actual data packet and receiving control messages from the neighbor MNs such as RREQ which is known as overhearing. In the network which contains a large number of MNs, if one of the MN sends a data packet, more number of intermediate MNs in the transmission range of the source MN sense and receive this packet. Consequently, the MNs consume more energy in Rx state. When the MN broadcasts a routing packet such as RREQ or Hello messages, it is received by many MNs which increases Rx EC ratio, but decreases Tx EC ratio as can be seen from Figure 5.3.

Because of the same reason, the MNs cannot stay idle state for a long time as it is described in Figure 5.4. The optimal path is selected as the shortest path in proactive routing protocols (DSDV and OLSR), which is also mentioned in Chapter 2, in order to reach the destination MN. On the other hand, due to the nature of reactive routing protocols, AODV and DSR prefer lowest latency path without considering hop count, but this path may contain more number of intermediate MNs. Therefore, proactive protocols can stay more time in idle state and they consume more energy there as compared to AODV and DSR.

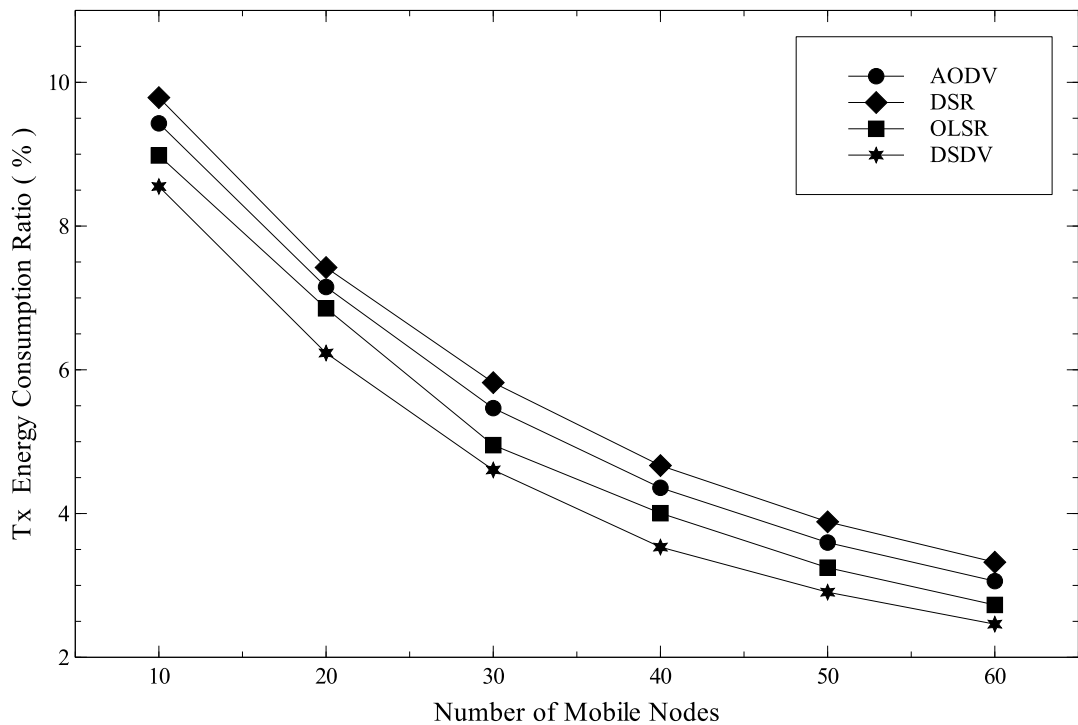


Figure 5.3: Effect of the number of MNs on Tx EC ratio.  
 (load:100 kbps, speed: 2 m/s, area: 1000×1000 m<sup>2</sup>)

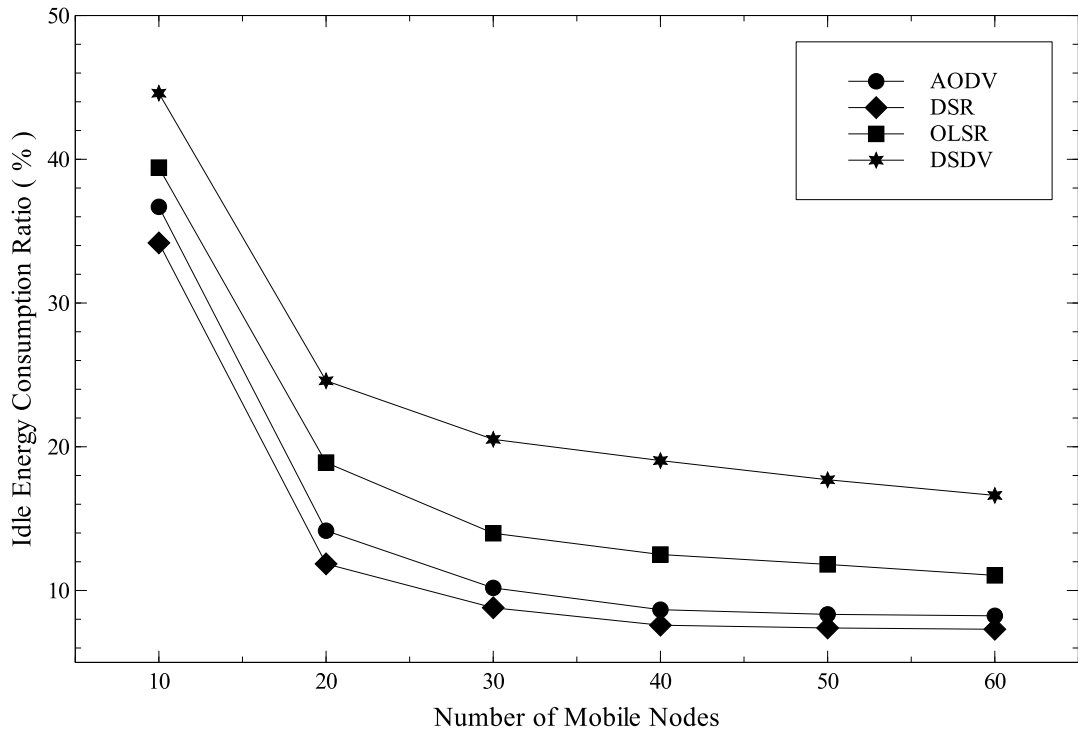


Figure 5.4: Effect of the number of MNs on idle EC ratio. (load:100 kbps, speed: 2 m/s, area: 1000×1000 m<sup>2</sup>)

By examining all results for the MN density scenario, the performance of the DSDV based on the EC is better under high MN density environment. Conversely, reactive protocols consume more energy and their ECs are very close each other.

As mentioned in Chapter 2, there have been some limited works about performance comparison of EC for existing MANET routing protocols. This study depends mainly on one of these studies that is [2] and it presents comparison of three different protocols in terms of EC which are AODV, DSR and OLSR by using distinct conditions (node, load and speed varying). Each simulation scenario is evaluated by averaging 5 runs in the mentioned study [2]. On the other hand, the results of this thesis are obtained by using an average of 50 runs for each scenario to improve the accuracy of the results.

In [2], various numbers of MNs which is changed from 10 to 60 MNs is constructed in order to analyze the effect of MN density in node varying condition and illustrated in Figure 5.5.

Both studies show that total/average EC of all routing protocols is increased when the number of MN density increases. Because MNs consume much more energy for flooding more control overheads generated to the network. As can be seen from Figure 5.5(a), OLSR consumes less energy among the others at 280 J under high MN density environment (60 MNs), while AODV and DSR consume energy at 283 and 284 J, respectively.

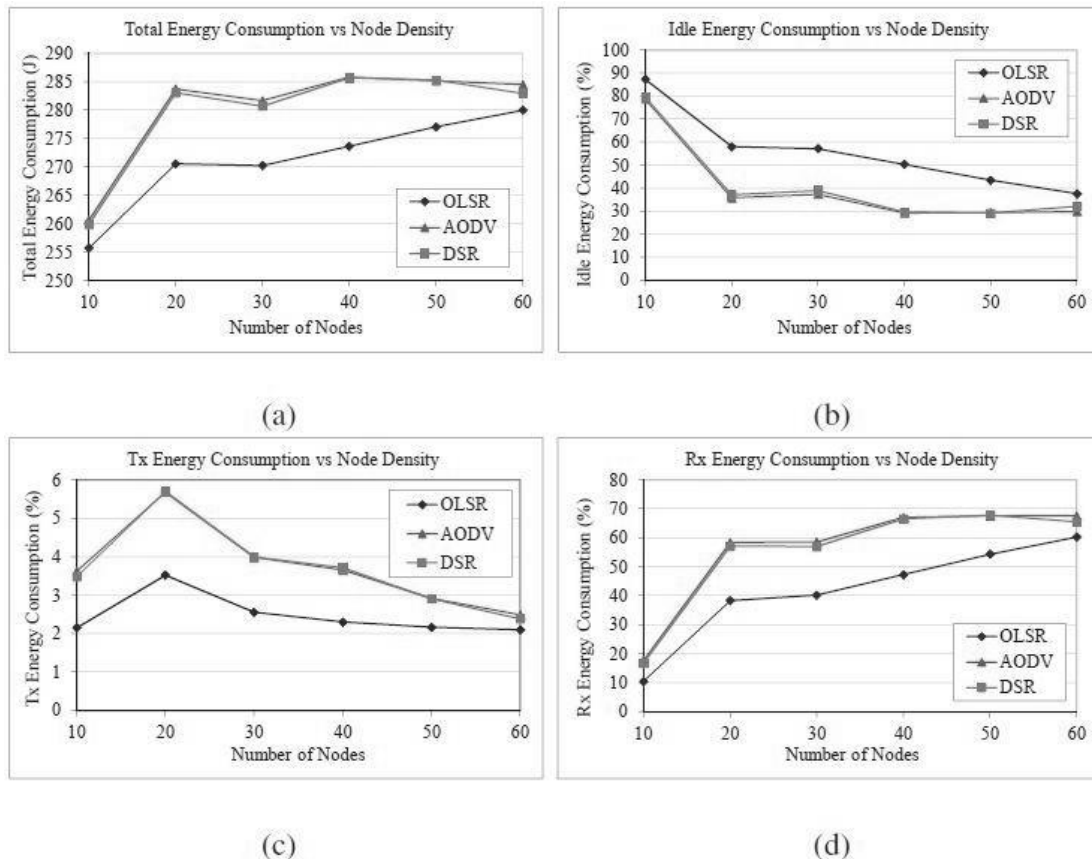


Figure 5.5: Effect of node density on (a) total EC, (b) EC in idle state, (c) EC in Tx state and (d) EC in Rx state [2].

(load:100 kbps, speed: 2 m/s, area: 1000×1000 m<sup>2</sup>)

Likewise, the table-driven protocols (OLSR and DSDV) have better performance as compare with AODV and DSR in our study which is shown in Figure 5.1. According to the simulation results of this thesis, DSDV consumes the least energy with 223 J OLSR follows DSDV with 245 J. AODV and DSR, are much closer each other and their ECs are 257 J and 262 J respectively under high MN density environment. With respect to Tx EC (Figures 5.3 and 5.5(c)) and idle EC (Figures 5.4 and 5.5(b)) graphs in both studies, the MNs spend less time in these two states (idle and Tx) in high MN density environment because they always change their state to Rx frequently. When one of the MN within the network sends a packet, more number of intermediate MNs between source and destination path sense and receive this data which inherently results in increasing Rx EC ratio.

Apparently, the behaviors of all routing protocols are almost same as this thesis. The unknown parameters which can affect energy level (i.e. packet size, receiving range, carrier sense range) and distinct number of repeated simulations cause differences between two studies. All simulation results of mentioned study [2] support the findings of this thesis. Therefore comparisons of these two studies are not given for other scenarios.

## **5.2 Load Varying Scenario**

The performance of all routing protocols on EC by varying load scenario which is changed from 25 kbps to 150 kbps is depicted in Figures 5.6 – 5.9. For this scenario, the simulation environment contains 50 MNs, the movement speed of the MNs is 2 m/s, the range of the area is  $1000 \times 1000 \text{ m}^2$  and the other parameters are shown in Table 4.1. According to the Figure 5.6, the average EC increases with the rise of the offered load because of the density of transmitting and receiving the data packets.

It can also be observed from Figure 5.6, DSDV consumes the least energy with 251 J in the maximum packet exchange density situation. On the other hand, the average EC of AODV, DSR and OLSR are 259 J, 268 J and 270 J respectively when offered load is 150 kbps.

Apparently, more MNs attend in sending and receiving data packets under heavy load environment and they consume more energy in Tx and Rx mode. Because of this reason, the curves of Figure 5.7 and 5.8 slightly climb when the offered load increases. Another consequence of this rise, MNs can stay in idle state for a short period as it can be seen from Figure 5.9.

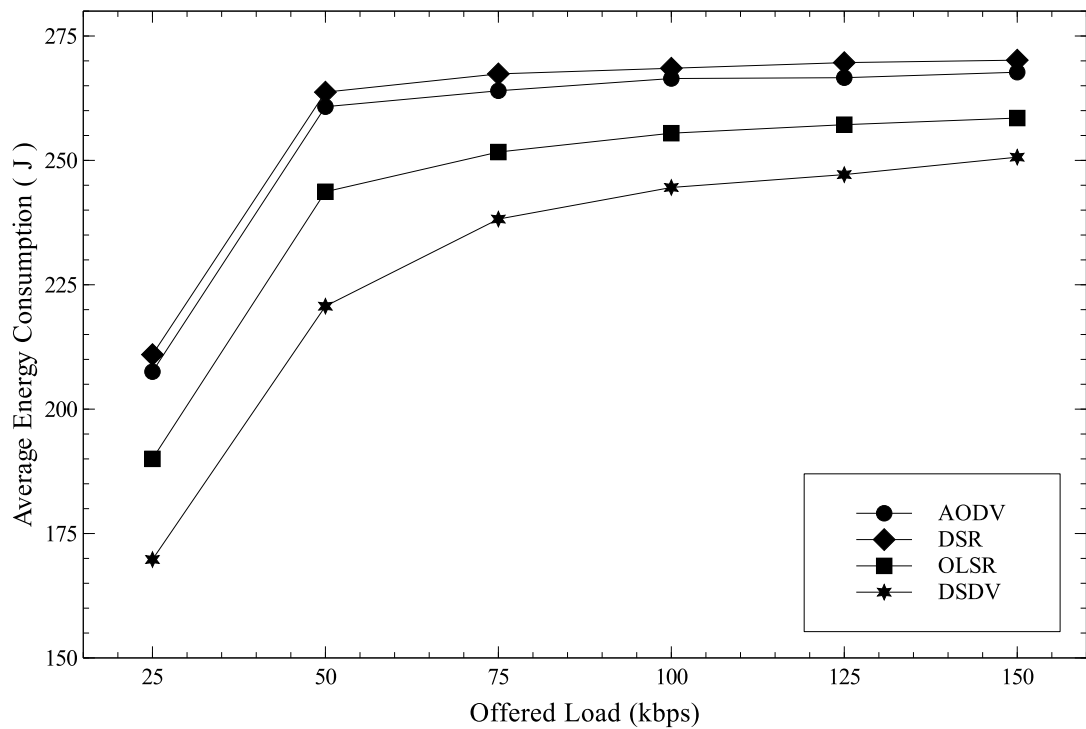


Figure 5.6: Effect of the offered load on average EC.  
(node: 50, speed: 2 m/s, area: 1000×1000 m<sup>2</sup>)



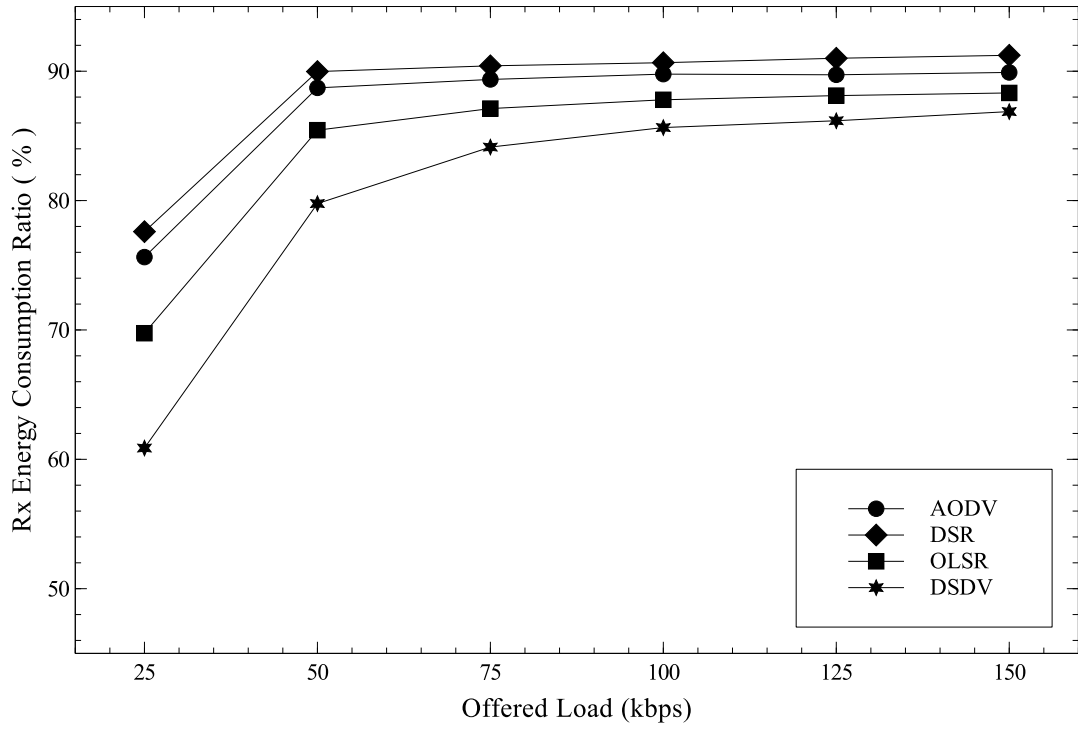


Figure 5.7: Effect of the offered load on Rx EC ratio.  
(node: 50, speed: 2 m/s, area:  $1000 \times 1000 \text{ m}^2$ )

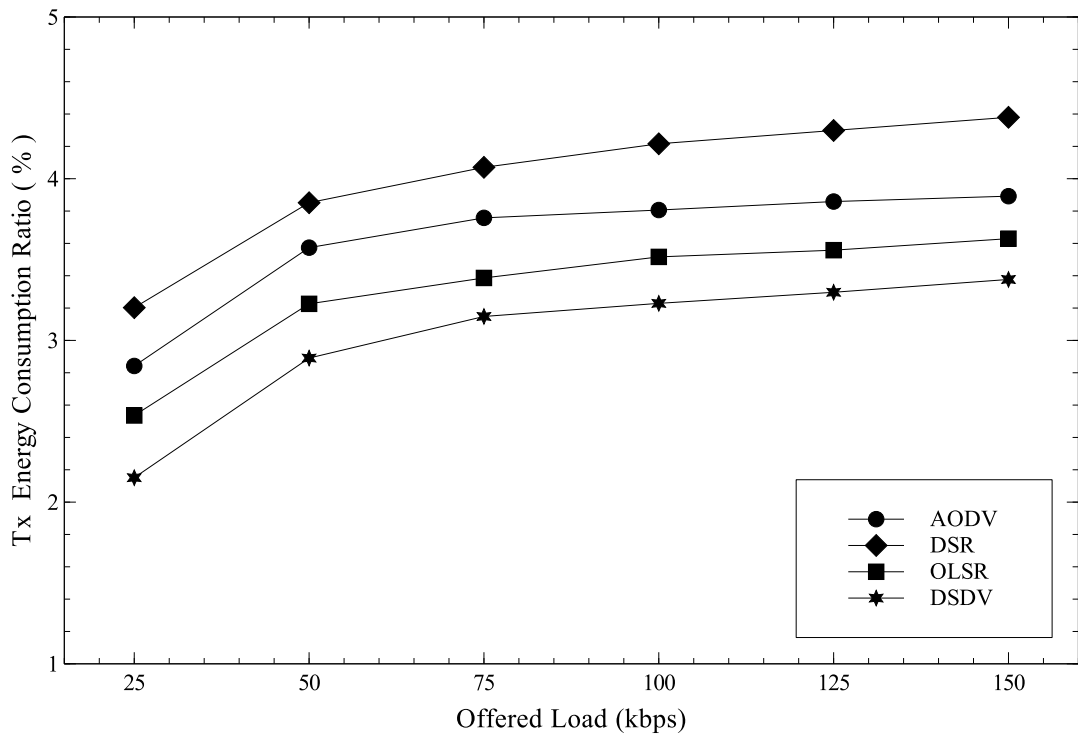


Figure 5.8: Effect of the offered load on Tx EC ratio.  
(node: 50, speed: 2 m/s, area:  $1000 \times 1000 \text{ m}^2$ )

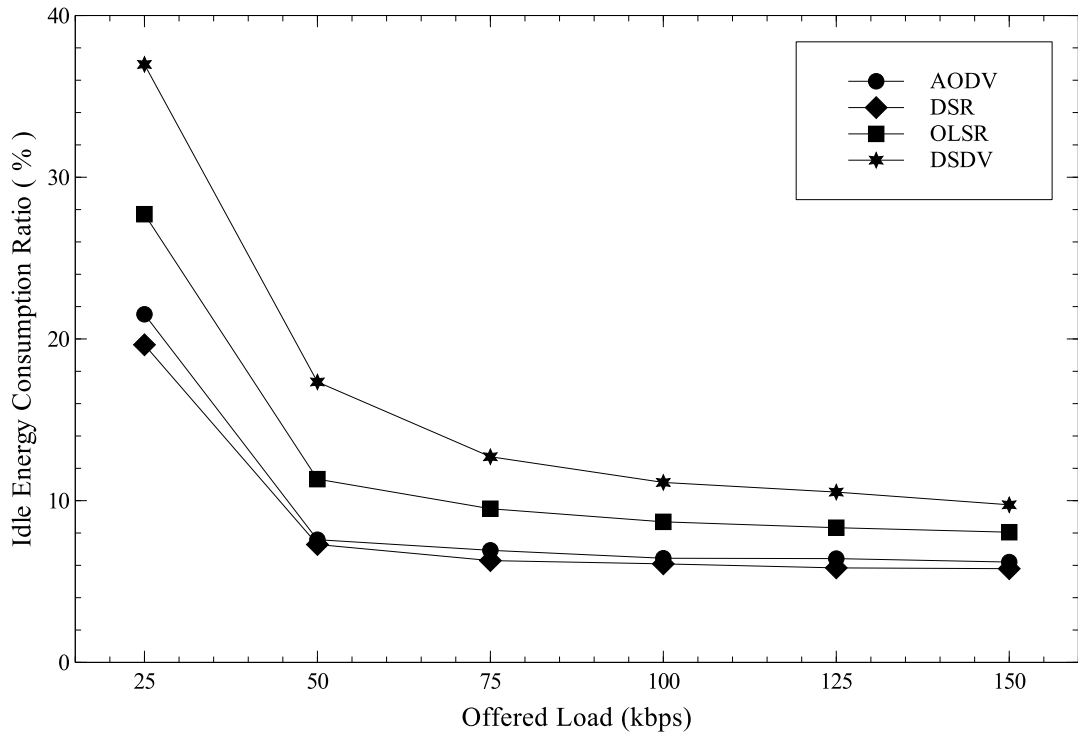


Figure 5.9: Effect of the offered load on idle EC ratio.  
(node: 50, speed: 2 m/s, area:  $1000 \times 1000 \text{ m}^2$ )

Additionally, DSDV is observed as the protocol that has the least Tx and Rx EC ratios. As it is stated before, because of the method of choosing the optimal path, DSDV spends more time in the idle state and consumes more energy among the other protocols.

### 5.3 Movement Speed Varying Scenario

The effect of the movement speed of MNs on EC, which is altered from 2 to 30 m/s, is illustrated in Figures 5.10 – 5.13. This simulation environment contains 50 MNs, the offered load is set as 100 kbps and the area is  $1000 \times 1000 \text{ m}^2$ .

Figure 5.10 highlights the average EC of the routing protocols in this scenario. According to this figure, when MNs in the network move with high speed the average EC of all mentioned routing protocols is slightly decreased.

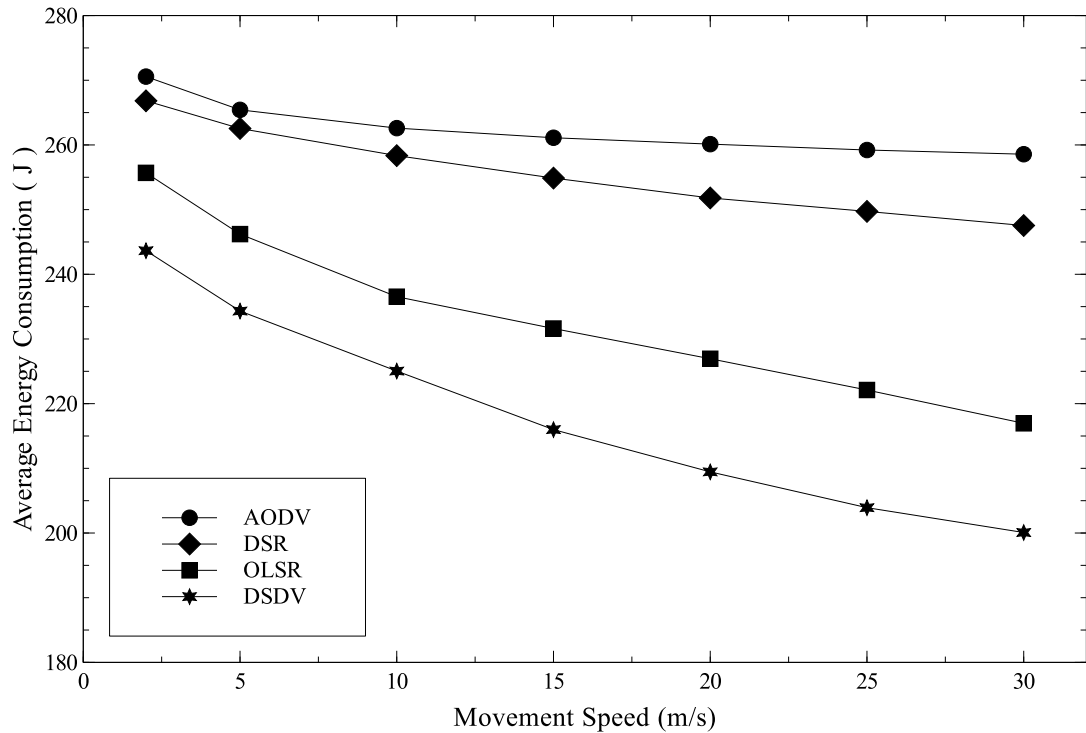


Figure 5.10: Effect of the movement speed on average EC.  
(node: 50, load: 100 kbps, area:  $1000 \times 1000 \text{ m}^2$ )

When MNs within the network move very fast, the network becomes highly dynamic which causes link breaks. Thus, the packet delivery ratio is affected indifferently. As can be easily seen from Figure 5.10, the proactive routing protocols are more sensitive to the movement speed of the MNs. It is because of their low packet delivery ratio under the high MN movement speed environment. Conversely, the average EC of reactive protocols keeps almost stable, especially for AODV (around 260 J).

Additionally, when the MNs move at a speed below 10 m/s, the average EC of AODV and DSR are still worst. Thus, the performance of the proactive routing protocols, especially DSDV, is better regardless movement speed.

As can be seen from the Figures 5.11 and 5.12, Rx EC ratio of all protocols decreases because of the packet loss while Tx EC ratio is almost stable.

Additionally, in the case of link breaks, the MNs have less possibility to receive a data packet from the others or send data to other MNs. So they change their state to idle because of link breaks in the network which inherently results in much more EC to occur in this state as demonstrated Figure 5.13.

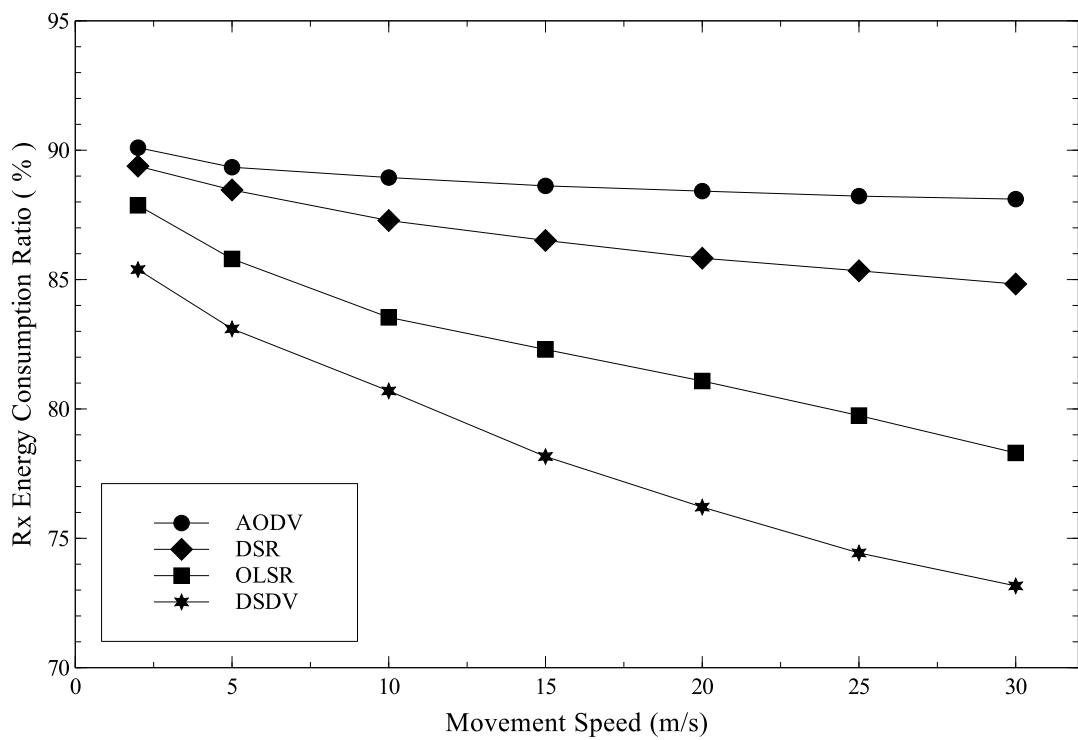


Figure 5.11: Effect of the movement speed on Rx EC ratio.  
(node: 50, load: 100 kbps, area:  $1000 \times 1000 \text{ m}^2$ )

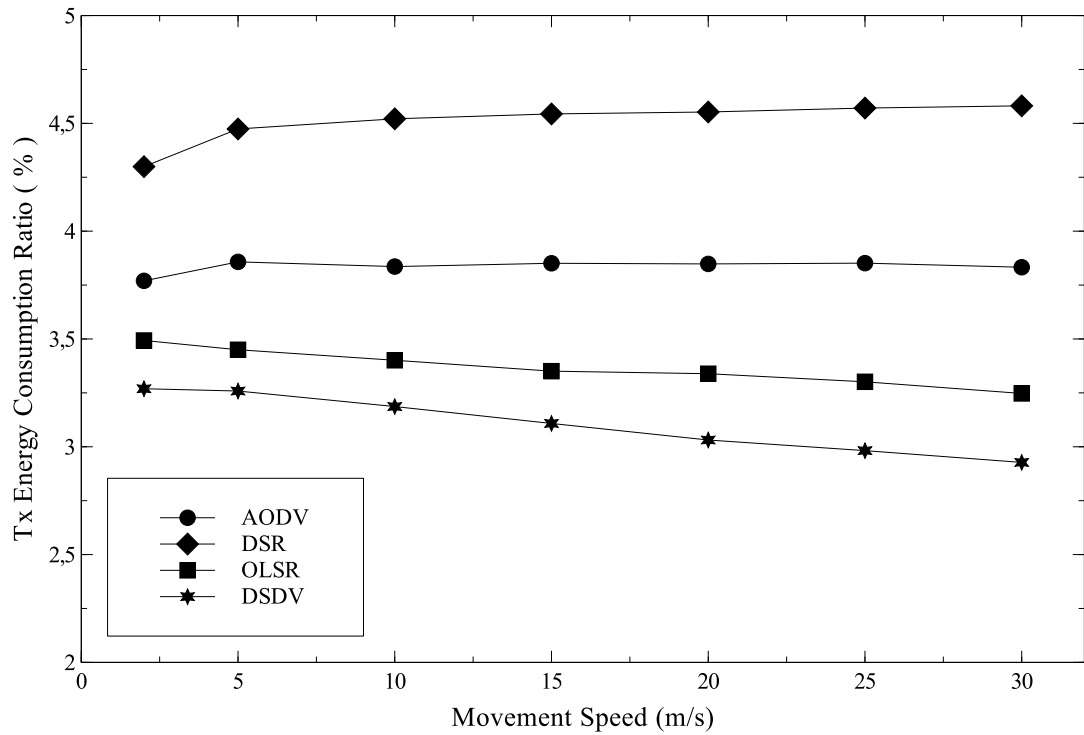


Figure 5.12: Effect of the movement speed on Tx EC ratio.  
(node: 50, load: 100 kbps, area:  $1000 \times 1000 \text{ m}^2$ )

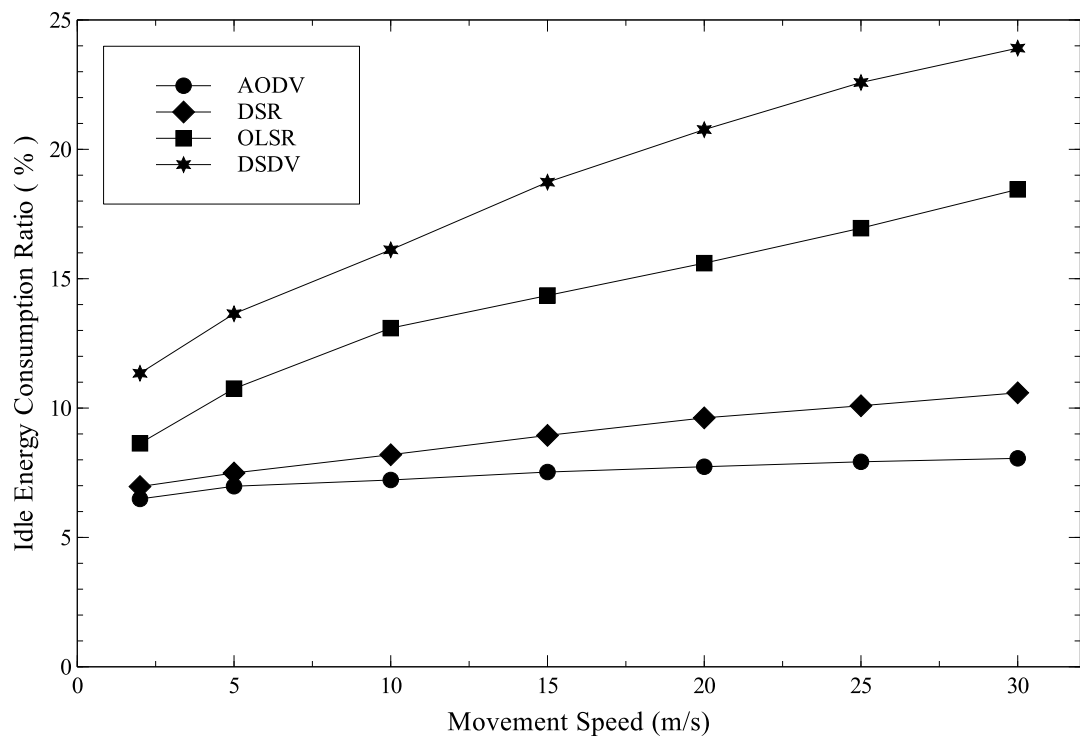


Figure 5.13: Effect of the movement speed on Idle EC ratio.  
(node: 50, load: 100 kbps, area:  $1000 \times 1000 \text{ m}^2$ )

## 5.4 Area Varying Scenario

Figures 5.14 - 5.16 represent the behavior of the mentioned protocols when the simulation area is increasingly changed from  $500 \times 500 \text{ m}^2$  to  $1000 \times 1000 \text{ m}^2$  in order to compose six different range scenarios. The network size is 50 MNs, movement speed of each MN is 2 m/s, the load is 100 kbps and the other parameters are as given in Table 4.1 in this scenario. The average EC comparison of the routing protocols under the different area model is shown in Figure 5.14.

As demonstrated in Figure 5.14, since the area increases, MNs in the network go out of the range each other in other words, they are not in their routing path. Thus, it causes the curves of the average EC graphs go down regularly. By examining this figure, the performance of the DSDV is again better under all (small/large) range.

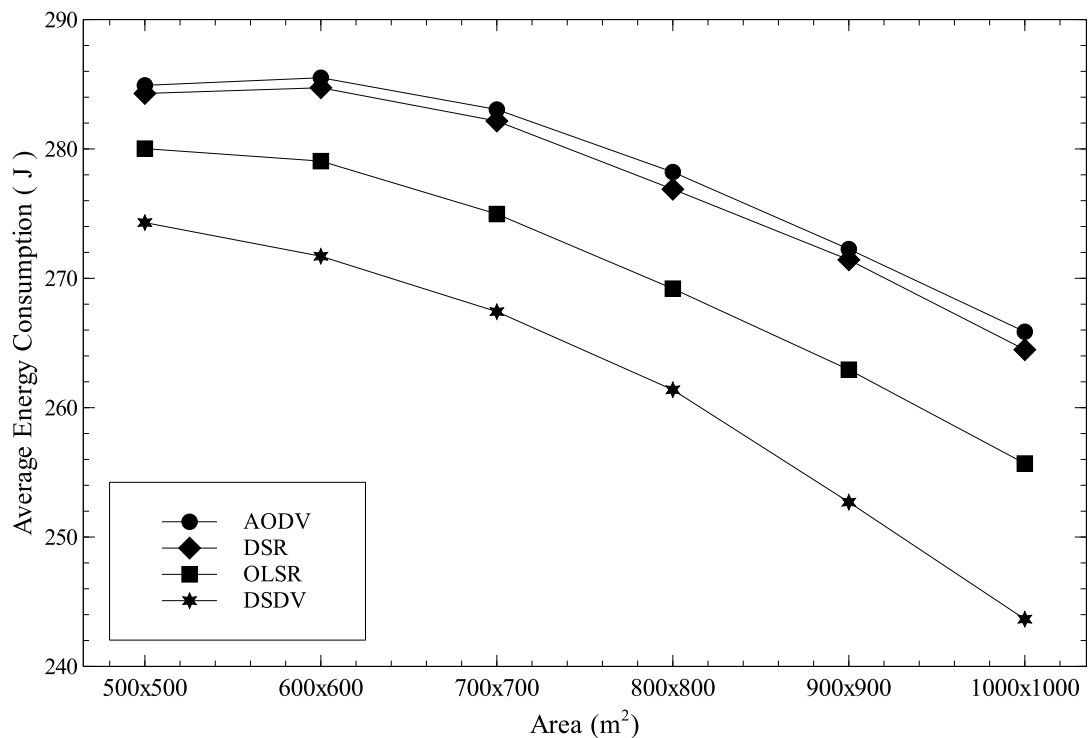


Figure 5.14: Effect of the topological area on average EC.  
(node: 50, load: 100 kbps, speed: 2 m/s)

When the network area becomes larger, the intermediate MNs between source and destination MNs are decreased so the MNs receive fewer data packets such as Hello messages. Thus, the MNs spend less time in Rx state as can be seen from Figure 5.15. On the other hand, all protocols consume more energy in Tx state because the MNs use more RREQ messages when they lose the route to the destination MN due to the path fails more often which is demonstrated in Figure 5.16.

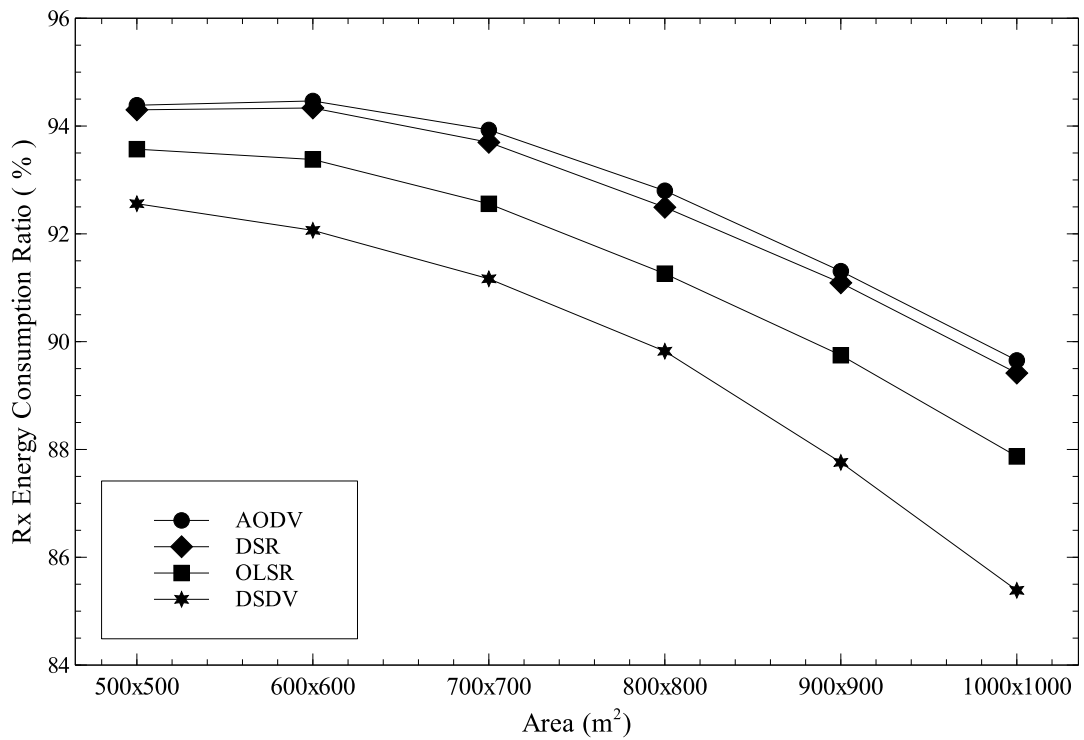


Figure 5.15: Effect of the topological area on Rx EC ratio.  
(node: 50, load: 100 kbps, speed: 2 m/s)

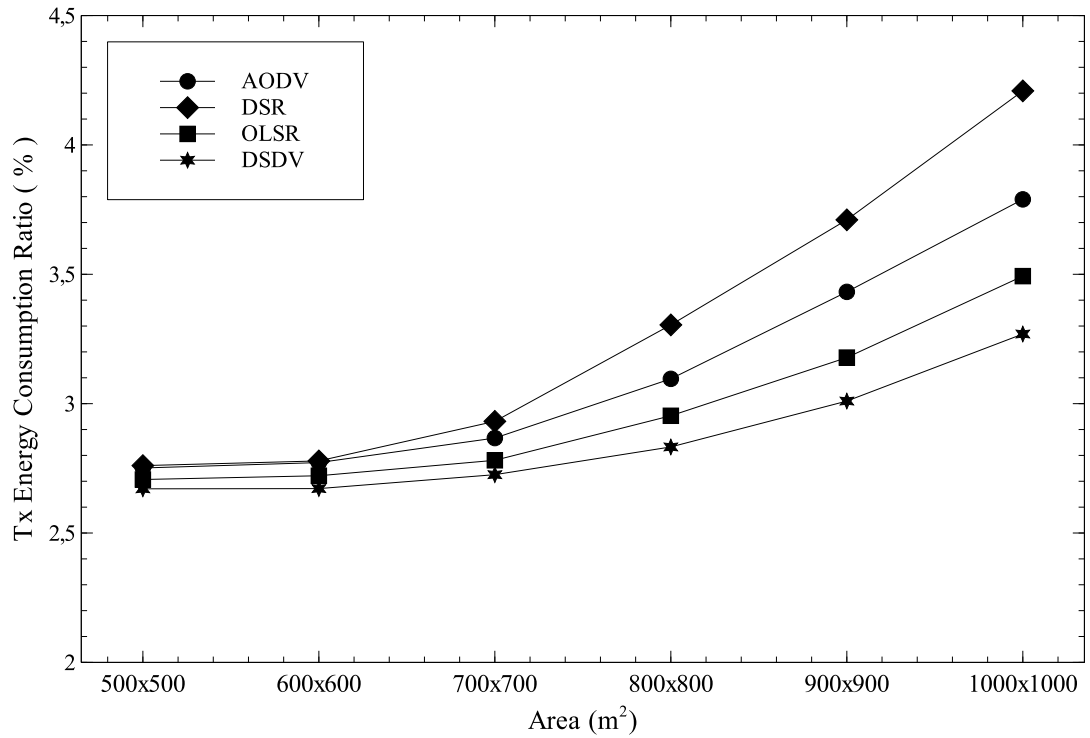


Figure 5.16: Effect of the topological area on Tx EC ratio.  
(node: 50, load: 100 kbps, speed: 2 m/s)

As mentioned before when the terrain of the network becomes larger, it causes a decrease in the intermediate MNs density and also in the failure of the route. Thus, the MNs change their state to idle state continually which consequently results in much more EC ratio in the idle state as illustrated in Figure 5.17.



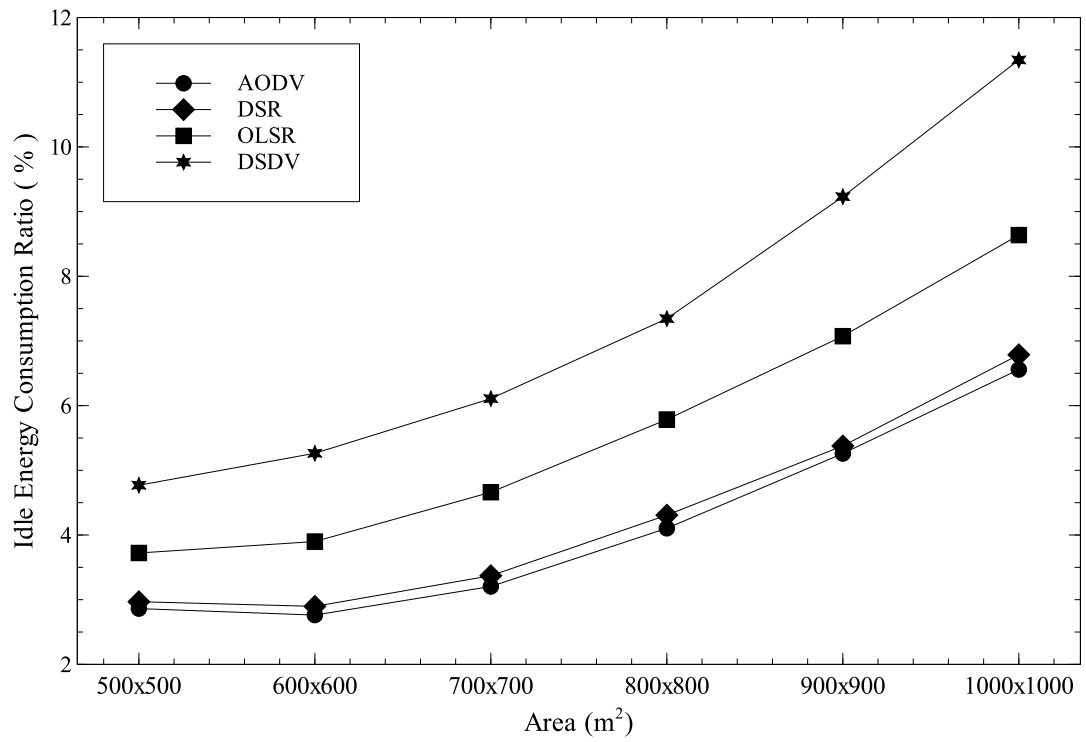


Figure 5.17: Effect of the topological area on Idle EC ratio.  
(node: 50, load: 100 kbps, speed: 2 m/s)

## Chapter 6

### CONCLUSION AND FUTURE WORK

#### 6.1 Conclusion

Energy saving has a primary priority in all technology fields. As wireless network becomes an integral component of the modern communication, energy efficiency in other words green communication is an important issue because of the limited battery of MNs especially for the distributed wireless networks (i.e. without network infrastructure) such as wireless sensor networks and MANETs.

Wireless sensor networking is an emerging technology that has a wide range of applications such as environment monitoring, medical systems etc. Such a network usually consists of a large number of distributed nodes that organize themselves. They are energy constrained. Charging and replacing battery are costly and complex which make energy conservation clearly an important issue. Because of this reason energy efficiency in wireless sensor networks is a hot topic for researchers.

Additionally, the performance investigation of different routing protocols in MANETs has been also very popular topic for the researchers, but only a few of them considered EC as a metric. In this thesis average EC, Idle EC Ratio, Tx EC Ratio and Rx EC Ratio are used as the performance metrics in order to compare the behavior of two reactive (AODV and DSR) and two proactive (OLSR and DSDV) routing protocols under four different scenarios by using NS2 environment.

The EC is proportional to the number of MNs within the network. When the number of MNs increases in the same topological area, the intermediate nodes between source and destination also increase. Consequently, more number of MNs receives the data packet and the EC of all routing protocols increases. Likewise with the rise of the offered load, all protocols act in a similar way and EC of the MNs increases. The density of transmitting and receiving the data packets causes more EC in Tx and Rx states under heavy load environment. Moreover, when MNs move with high speed, EC of all routing protocols is decreased due to the connection failures. Additionally, when the network contains the same amount of MNs and the topological area increases, MNs in the network go out of the routing path that causes EC of all routing protocols go down.

The simulation results show that, the performance of the table-driven protocols especially DSDV have better performance in terms of EC under all conditions as compared to the on-demand protocols. On the other hand, reactive protocols consume more energy level and their performances are very close each other.

## **6.2 Future Work**

As future work, we would like to study on more performance metric and additional routing protocols and also develop a new energy aware routing protocol in order to attain more energy conservation in MANETs.

## REFERENCES

- [1] Sarkar, S. K., Basavaraju, T. G., & Puttamadappa, C. (2008). Ad Hoc Mobile Wireless Networks Principles, Protocols and Applications. Auerbach Publications, pp. 231-234.
  
- [2] Kunavut, K. (2014). Performance and Comparative Analysis of Energy Consumption for Mobile Ad Hoc Routing Protocols. *Proceedings of International Symposium on Communications and Information Technologies*, pp. 76-79, Incheon, Korea.
  
- [3] Pandey K., & Swaroop, A. (2011). A Comprehensive Performance Analysis of Proactive, Reactive and Hybrid MANETs Routing Protocols. *International Journal of Computer Science Issues*, 8(6), pp. 432-441.
  
- [4] Perkins, C. E. Ad hoc on-demand distance vector (AODV) routing, Request For Comments (RFC) 3561, July 2003, Available online at: <http://www.ietf.org/rfc/rfc3561.txt>, last accessed on April, 2016.
  
- [5] Johnson, D., Hu, Y., & Maltz, D. (2007). The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728 (Experimental), Internet Engineering Task Force. Available online: <http://www.ietf.org/rfc/rfc4728.txt>, last accessed on April, 2016.
  
- [6] Clausen T., & Jacquet P. (2003). Optimized Link State Routing Protocol (OLSR), RFC 3626 (Experimental), Internet Engineering Task Force.

Available online:<http://www.ietf.org/rfc/rfc3626.txt>, last accessed on April 2016.

- [7] Perkins, C. E., & Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Computer Communication Review*, 24(4), London England.
- [8] Bakshi, A., Sharma, A. K., & Mishra, A. (2013). Significance of Mobile Ad-Hoc Networks (MANETS). *International Journal of Innovative Technology and Exploring Engineering*, 2(4), pp.1-5.
- [9] Jyoti, N. S. (2012). Comparative Study of Adhoc Routing Protocol AODV, DSR and DSDV in Mobile Adhoc Network. *International Journal of Applied Engineering Research*, 7(11).
- [10] Kafhali, S. E. & Haqiq, A. (2013). Effect of Mobility and Traffic Models on the Energy Consumption in MANET Routing Protocols. *International Journal of Soft Computing and Engineering*, 3(1), pp. 242-249.
- [11] Kara, R., Özçelik İ., & Ekiz, H. (2007). Position Based Hybrid Routing Algorithm in Wireless Ad Hoc Networks. *5th International Conference on Electrical and Electronics Engineering*, pp. 5-9.
- [12] Qayyum, A., Viennot, L., & Laouiti, A. (2002). Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. *In Proceeding of the 35th Hawaii International Conference on System Sciences*, pp. 3866-3875.

- [13] Boppana, R. V., Marina, M. K., & Konduru, S. P. (1999). An Analysis of Routing Techniques for Mobile and Ad Hoc Networks. *High Performance Computing HiPC'99*, pp. 239-245, Springer Berlin Heidelberg.
- [14] Alexandros, K. (2013). A Performance Evaluation of Proactive and Reactive Routing Protocols for Wireless Mobile Ad Hoc Networks using Network Simulator 2. *International Journal of Scientific & Engineering Research*, 4(7), pp. 760-764.
- [15] Kaur, N., Singh, A., Kumar A., & Shurma, M. B. (2011). Comparison and Analysis of RREQ and RREP for Dynamic Wireless Network. *Indian Journal of Computer Science and Engineering*, 2(3), pp. 437-443.
- [16] El-Desoky, A., Sarhan, A. & Arnous, R. (2014). A Simulated Behavioral Study of DSR Routing Protocol Using NS-2. *International Journal of Engineering Research and Applications*, 4(12), pp. 64-71.
- [17] Udayakumar, R., Khanaa, K. P. T., & Geo, A. V. A. (2014). Performance Comparison for DSR and AODV. *Computer Sciences, Engineering and Its Applications*, pp. 36-41.
- [18] Kulkarni, R., Asha S., & Kulkarni, N. (2013). Comparative Analysis Of AODV, DSR Routing Protocols In Mobile Ad-Hoc Networks. *International Journal of Engineering Research & Technology*, 2(7), pp. 283-290.

- [19] Kumar, N., Kumar, V., & Kumar, N. (2014). Comparative Study of Reactive Routing Protocols AODV and DSR for Mobile Ad hoc Networks. *International Journal of Computer Science and Information Technologies*, 5(5), pp. 6888-6891.
- [20] Cano, J. C., & Manzoni, P. (2000). A Performance Comparison of Energy Consumption for Mobile Ad Hoc Network Routing Protocols. *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 8th International Symposium*, pp. 57-64.
- [21] Zhang, J. H., Peng, H., & Shao, F. J. (2011). Energy consumption analysis of MANET routing protocols based on mobility models. *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference*, vol. 4, pp. 2275-2280.
- [22] Khiavi, M. V., Jamali, S., & Gudakahriz, S. J. (2012). Performance Comparison of AODV, DSDV, DSR and TORA Routing Protocols in MANETs. *International Research Journal of Applied and Basic Sciences*, 3(7), pp.1429-1436.
- [23] Kumar, A., Rafiq, M. Q., & Bansal, K. (2012). Performance Evaluation of Energy Consumption in MANET. *International Journal of Computer Applications*, 42(2), pp. 7-12.

- [24] Bertocchi, F., Bergamo, P., Mazzini, G. & Zorzi, M. (2003). Performance Comparison of Routing Protocols for Ad Hoc Networks. *Global Telecommunications Conference*, vol. 2, pp. 1033-1037.
- [25] Barati, M., Atefi, K., Khosravi, F. & Daftari, Y. A. (2012). Performance Evaluation of Energy Consumption for AODV and DSR Routing Protocols in MANET. *Computer & Information Science, International Conference*, vol. 2, pp. 636-642.
- [26] Gouda, B. S., Mandal, A. B., & Narayana, K. L.(2012). Simulation and Comparative Analysis of Energy Conservation Performance Metric for ERAODV, AODV and DSDV Routing Protocols in MANET. *World Congress Information and Communication Technologies (WICT)*, pp. 278-282.
- [27] Samara, C., Karapistoli E., & Economides, A. A. (2012). Performance Comparison of MANET Routing Protocols based on real-life scenarios. *4th International Congress, Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 870-877.
- [28] Jabbar, W. A., Ismail, M., & Nordin, R. (2014). On the Performance of the Current MANET Routing Protocols for VoIP, HTTP, and FTP Applications. *Journal of Computer Networks and Communications*.
- [29] The network simulator-ns2 <http://www.isi.edu/nsnam/ns/>, last accessed on May 2016.



## **APPENDICES**

## Appendix A: TCL File and Output Files

### Appendix A.1: AODVenergy\_10.tcl File

```
Mac/802_11 set dataRate_ 2e6;           # channel capacity is 2 Mbs
Phy/WirelessPhy set Pt_ 0.2818;        # For 250m transmission power range.
Phy/WirelessPhy set bandwidth_ 2e6     # channel bandwidth is 2 Mbs

#=====
#   Simulation parameters setup
#=====
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround  ;# radio-propagation model
set val(netif) Phy/WirelessPhy         ;# network interface type
set val(mac) Mac/802_11                 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
#(changed according to the routing protocols)
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 50                      ;# max packet in ifq
set val(nn) 10                          ;# number of MNs
# (changed according to the scenario as 10, 20, 30, 40, 50, 60)
set val(rp) AODV                        ;# routing protocol
# (changed according to the simulation environment as DSR, OLSR and DSDV)
set val(x) 1000                         ;# X dimension of topography
#(changed according to the scenario as 500, 600, 700, 800, 900)
set val(y) 1000                         ;# Y dimension of topography
#(changed according to the scenario as 500, 600, 700, 800, 900)
set val(stop) 300.0                    ;# time of simulation end
set val(energymodel) EnergyModel        ;# EnergyModel
set val(initialenergy) 1000             ;# Initial energy in Joules
set val(mv) "movement10"               ;#Movement Pattern Connection
set val(tfc) "traffic10"               ;# Traffic Connection

#=====
#   Initialization
#=====
#Create a ns_ simulator
set ns_ [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(nn)]

#Open the NS trace file
```

```

set tracefile [open out.tr w]
#$ns_ use-newtrace
$ns_ trace-all $tracefile

#Open the NAM trace file
#set namfile [open out.nam w]
#$ns_ namtrace-all $namfile
#$ns_ namtrace-all-wireless $namfile $val(x) $val(y)

#Create wireless channel
set chan [new $val(chan)];

#=====
#   Mobile node parameter setup
#=====
$ns_ node-config -adhocRouting $val(rp) \
    -llType      $val(ll) \
    -macType     $val(mac) \
    -ifqType     $val(ifq) \
    -ifqLen     $val(ifqlen) \
    -antType     $val(ant) \
    -propType   $val(prop) \
    -phyType     $val(netif) \
    -channel     $chan \
    -topoInstance $topo \
    -energyModel $val(energymodel) \
    -initialEnergy $val(initialenergy) \
    -txPower 1.4 \
    -rxPower 1.0 \
    -idlePower 0.32 \
    -sleepPower 0.0 \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON \

#=====
#   Nodes Definition
#=====

for { set i 0 } { $i < $val(nn) } { incr i } {
set node_($i) [$ns_ node]
}

if 0 {
#node mobility syntax
#$ns_ at start time "$node no setdest Destination node x y node speed"

$ns_ at 0.0 "destination"

```

```

proc destination {} {
    global ns_ val node_
    set time 50.0
    set now [$ns_ now]
    for {set i 0} {$i < $val(nn)} {incr i} {
        set xx [expr rand()*1000]
# changed according to the scenario as 500, 600, 700, 800, 900
        set yy [expr rand()*1000]
# changed according to the scenario as 500, 600, 700, 800, 900
        $ns_ at $now "$node_($i) setdest $xx $yy 2";
#speed is changed according to the scenario as 2, 5, 10, 15, 20, 25, 30
    }
    $ns_ at [expr $now + $time] "destination"
}
}

```

```

source $val(mv) #source movement pattern by referance
source $val(tfc)#source traffic by referance

```

```

# Define the initial position randomly

```

```

for { set i 0} {$i< $val(nn)} {incr i} {
$ns_ initial_node_pos $node_($i) 20
}

```

```

#Inform all MNs when the simulation end

```

```

for { set i 0} {$i< $val(nn)} {incr i} {
$ns_ at $val(stop) "$node_($i) reset"
}

```

```

$ns_ at $val(stop) "finish"; #stop proc is named finish.

```

```

#=====
# Termination
#=====
if 0 {
proc plotWindow {tcpSource file} {
global ns_
set now[$ns_ now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns_ at [expr $now+$time] "plotWindow $tcpSource $file"}
$ns_ at 0.1 "plotWindow $tcp0 $winFile"
}

```

```

#Define a 'finish' procedure

```

```

proc finish {} {
    global ns_ tracefile namfile

```

```

    $ns_ flush-trace
    close $tracefile
    #close $namfile
    #exec nam out.nam &
    exit 0
}

```

```

$ns_ at $val(stop) "$ns_ nam-end-wireless $val(stop)"
#$ns_ at $val(stop) "finish"
$ns_ at $val(stop) "puts \"done\" ; $ns_ halt"
$ns_ run

```

## Appendix A.2: Example of Output Files

### Appendix A.2.1: Example of Mobility File (movement10)

```

#
# nodes: 10, speed type: 2, min speed: 2.00, max speed: 2.00
# avg speed: -nan, pause type: 1, pause: 0.00, max x: 1000.00, max y: 1000.00
#
$node_(0) set X_ 893.354827728543
$node_(0) set Y_ 868.896607712492
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 632.925997406386
$node_(1) set Y_ 906.678891877925
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 829.181357629067
$node_(2) set Y_ 697.073549495847
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 39.292367448288
$node_(3) set Y_ 766.103486379032
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 64.719316424247
$node_(4) set Y_ 663.905262037249
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 874.531939137411
$node_(5) set Y_ 584.516834835406
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 754.227553000518
$node_(6) set Y_ 613.413311911274
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 135.433146536838
$node_(7) set Y_ 925.010526879269
$node_(7) set Z_ 0.000000000000
$node_(8) set X_ 567.205434194470
$node_(8) set Y_ 817.886594710968
$node_(8) set Z_ 0.000000000000
$node_(9) set X_ 355.686483667881

```

\$node\_(9) set Y\_ 730.130244481166  
 \$node\_(9) set Z\_ 0.000000000000  
 \$ns\_ at 0.000000000000 "\$node\_(0) setdest 804.474439781784 191.575727622898  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(1) setdest 398.664809978167 494.028946561278  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(2) setdest 831.399246801399 143.405707280242  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(3) setdest 533.874969008843 237.743570108587  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(4) setdest 13.908110301217 233.240541656044  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(5) setdest 312.120573111130 37.087570576513  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(6) setdest 636.776773596548 578.402143509045  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(7) setdest 865.565215013354 971.388987276915  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(8) setdest 892.844882493777 394.663861042374  
 2.000000000000"  
 \$ns\_ at 0.000000000000 "\$node\_(9) setdest 88.542487103687 853.039745574879  
 2.000000000000"  
 \$god\_ set-dist 0 1 16777215  
 \$god\_ set-dist 0 2 1  
 \$god\_ set-dist 0 3 16777215  
 \$god\_ set-dist 0 4 16777215  
 \$god\_ set-dist 0 5 2  
 \$god\_ set-dist 0 6 2  
 \$god\_ set-dist 0 7 16777215  
 \$god\_ set-dist 0 8 16777215  
 \$god\_ set-dist 0 9 16777215  
 \$god\_ set-dist 1 2 16777215  
 \$god\_ set-dist 1 3 16777215  
 \$god\_ set-dist 1 4 16777215  
 \$god\_ set-dist 1 5 16777215  
 \$god\_ set-dist 1 6 16777215  
 \$god\_ set-dist 1 7 16777215  
 \$god\_ set-dist 1 8 1  
 \$god\_ set-dist 1 9 2  
 \$god\_ set-dist 2 3 16777215  
 \$god\_ set-dist 2 4 16777215  
 \$god\_ set-dist 2 5 1  
 \$god\_ set-dist 2 6 1  
 \$god\_ set-dist 2 7 16777215  
 \$god\_ set-dist 2 8 16777215  
 \$god\_ set-dist 2 9 16777215  
 \$god\_ set-dist 3 4 1  
 \$god\_ set-dist 3 5 16777215  
 \$god\_ set-dist 3 6 16777215  
 \$god\_ set-dist 3 7 1

\$god\_ set-dist 3 8 16777215  
 \$god\_ set-dist 3 9 16777215  
 \$god\_ set-dist 4 5 16777215  
 \$god\_ set-dist 4 6 16777215  
 \$god\_ set-dist 4 7 2  
 \$god\_ set-dist 4 8 16777215  
 \$god\_ set-dist 4 9 16777215  
 \$god\_ set-dist 5 6 1  
 \$god\_ set-dist 5 7 16777215  
 \$god\_ set-dist 5 8 16777215  
 \$god\_ set-dist 5 9 16777215  
 \$god\_ set-dist 6 7 16777215  
 \$god\_ set-dist 6 8 16777215  
 \$god\_ set-dist 6 9 16777215  
 \$god\_ set-dist 7 8 16777215  
 \$god\_ set-dist 7 9 16777215  
 \$god\_ set-dist 8 9 1  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 0 1 4"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 0 8 3"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 0 9 4"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 1 2 3"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 1 5 3"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 1 6 2"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 2 8 2"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 2 9 3"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 5 8 2"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 5 9 3"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 6 8 1"  
 \$ns\_ at 9.633444831418 "\$god\_ set-dist 6 9 2"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 0 9 16777215"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 1 9 16777215"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 2 9 16777215"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 5 9 16777215"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 6 9 16777215"  
 \$ns\_ at 9.960262518487 "\$god\_ set-dist 8 9 16777215"  
 \$ns\_ at 13.710110978501 "\$god\_ set-dist 3 9 2"  
 \$ns\_ at 13.710110978501 "\$god\_ set-dist 4 9 3"  
 \$ns\_ at 13.710110978501 "\$god\_ set-dist 7 9 1"  
 \$ns\_ at 20.946798305161 "\$god\_ set-dist 3 9 1"  
 \$ns\_ at 20.946798305161 "\$god\_ set-dist 4 9 2"  
 \$ns\_ at 38.041509090675 "\$god\_ set-dist 3 7 2"  
 \$ns\_ at 38.041509090675 "\$god\_ set-dist 4 7 3"  
 \$ns\_ at 44.980640840865 "\$god\_ set-dist 0 1 3"  
 \$ns\_ at 44.980640840865 "\$god\_ set-dist 0 8 2"  
 \$ns\_ at 44.980640840865 "\$god\_ set-dist 1 2 2"  
 \$ns\_ at 44.980640840865 "\$god\_ set-dist 2 8 1"  
 \$ns\_ at 47.496196824390 "\$god\_ set-dist 1 5 2"  
 \$ns\_ at 47.496196824390 "\$god\_ set-dist 1 6 1"  
 \$ns\_ at 52.682702093333 "\$god\_ set-dist 0 1 2"  
 \$ns\_ at 52.682702093333 "\$god\_ set-dist 0 8 1"

\$ns\_ at 61.279008427620 "\$node\_(6) setdest 877.061178774742 221.780924389705  
2.000000000000"  
\$ns\_ at 70.999688061417 "\$god\_ set-dist 5 8 1"  
\$ns\_ at 87.482319263896 "\$god\_ set-dist 1 6 2"  
\$ns\_ at 95.282060692435 "\$god\_ set-dist 0 6 1"  
\$ns\_ at 116.334158090977 "\$god\_ set-dist 3 7 16777215"  
\$ns\_ at 116.334158090977 "\$god\_ set-dist 4 7 16777215"  
\$ns\_ at 116.334158090977 "\$god\_ set-dist 7 9 16777215"  
\$ns\_ at 120.527004341185 "\$god\_ set-dist 3 9 16777215"  
\$ns\_ at 120.527004341185 "\$god\_ set-dist 4 9 16777215"  
\$ns\_ at 130.016324887475 "\$god\_ set-dist 3 4 16777215"  
\$ns\_ at 138.861818670091 "\$god\_ set-dist 0 1 16777215"  
\$ns\_ at 138.861818670091 "\$god\_ set-dist 1 2 16777215"  
\$ns\_ at 138.861818670091 "\$god\_ set-dist 1 5 16777215"  
\$ns\_ at 138.861818670091 "\$god\_ set-dist 1 6 16777215"  
\$ns\_ at 138.861818670091 "\$god\_ set-dist 1 8 16777215"  
\$ns\_ at 147.031170470241 "\$node\_(9) setdest 161.047658067642  
180.632588121011 2.000000000000"  
\$ns\_ at 154.374472087086 "\$god\_ set-dist 1 3 1"  
\$ns\_ at 172.060760148214 "\$god\_ set-dist 5 8 2"  
\$ns\_ at 204.757953912772 "\$god\_ set-dist 2 5 2"  
\$ns\_ at 216.825897927749 "\$node\_(4) setdest 87.301702974074 276.850423446132  
2.000000000000"  
\$ns\_ at 216.910754308402 "\$god\_ set-dist 0 5 16777215"  
\$ns\_ at 216.910754308402 "\$god\_ set-dist 2 5 16777215"  
\$ns\_ at 216.910754308402 "\$god\_ set-dist 5 6 16777215"  
\$ns\_ at 216.910754308402 "\$god\_ set-dist 5 8 16777215"  
\$ns\_ at 231.701022293757 "\$god\_ set-dist 1 5 2"  
\$ns\_ at 231.701022293757 "\$god\_ set-dist 3 5 1"  
\$ns\_ at 237.254231419121 "\$node\_(1) setdest 288.748110658398  
277.005234178409 2.000000000000"  
\$ns\_ at 259.512081978255 "\$node\_(4) setdest 878.642364348660  
511.951782388140 2.000000000000"  
\$ns\_ at 267.001185663376 "\$node\_(8) setdest 701.507039580835  
616.289062236465 2.000000000000"  
\$ns\_ at 270.359776725600 "\$god\_ set-dist 2 8 2"  
\$ns\_ at 276.287897370614 "\$node\_(6) setdest 175.944932409689  
405.382687533181 2.000000000000"  
\$ns\_ at 276.836142210543 "\$node\_(2) setdest 960.267608458098  
343.575807861930 2.000000000000"  
\$ns\_ at 281.435819138721 "\$god\_ set-dist 1 5 1"  
\$ns\_ at 284.674262237205 "\$god\_ set-dist 1 4 1"  
\$ns\_ at 284.674262237205 "\$god\_ set-dist 3 4 2"  
\$ns\_ at 284.674262237205 "\$god\_ set-dist 4 5 2"  
#  
# Destination Unreachables: 54  
#  
# Route Changes: 61  
#  
# Link Changes: 23



```

#
# Node | Route Changes | Link Changes
# 0 | 11 | 2
# 1 | 20 | 6
# 2 | 10 | 3
# 3 | 9 | 6
# 4 | 9 | 2
# 5 | 17 | 6
# 6 | 9 | 5
# 7 | 6 | 3
# 8 | 13 | 8
# 9 | 18 | 5
#

```

### Appendix A.2.2: Example of Traffic File (traffic10)

```

set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(0) set random_ 1
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 0.0 "$cbr_(0) start"
#
# 1 connecting to 3 at time 0.0
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(3) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(1) set random_ 1
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 0.0 "$cbr_(1) start"
#
# 1 connecting to 2 at time 0.0
#
set udp_(2) [new Agent/UDP]

```

```

$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(2) set random_ 1
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 0.0 "$cbr_(2) start"
#
# 1 connecting to 3 at time 0.0
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(3) set random_ 1
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 0.0 "$cbr_(3) start"
#
# 2 connecting to 3 at time 0.0
#
set udp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(4)
set cbr_(4) [new Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(4) set random_ 1
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 0.0 "$cbr_(4) start"
#
# 2 connecting to 4 at time 0.0
#
set udp_(5) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(5)
set null_(5) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(5)
set cbr_(5) [new Application/Traffic/CBR]
$cbr_(5) set packetSize_ 512

```

```

$cbr_(5) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(5) set random_ 1
$cbr_(5) attach-agent $udp_(5)
$ns_ connect $udp_(5) $null_(5)
$ns_ at 0.0 "$cbr_(5) start"
#
# 3 connecting to 4 at time 0.0
#
set udp_(6) [new Agent/UDP]
$ns_ attach-agent $node_(2) $udp_(6)
set null_(6) [new Agent/Null]
$ns_ attach-agent $node_(4) $null_(6)
set cbr_(6) [new Application/Traffic/CBR]
$cbr_(6) set packetSize_ 512
$cbr_(6) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(6) set random_ 1
$cbr_(6) attach-agent $udp_(6)
$ns_ connect $udp_(6) $null_(6)
$ns_ at 0.0 "$cbr_(6) start"
#
# 3 connecting to 5 at time 0.0
#
set udp_(7) [new Agent/UDP]
$ns_ attach-agent $node_(8) $udp_(7)
set null_(7) [new Agent/Null]
$ns_ attach-agent $node_(1) $null_(7)
set cbr_(7) [new Application/Traffic/CBR]
$cbr_(7) set packetSize_ 512
$cbr_(7) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(7) set random_ 1
$cbr_(7) attach-agent $udp_(7)
$ns_ connect $udp_(7) $null_(7)
$ns_ at 0.0 "$cbr_(7) start"
#
# 4 connecting to 5 at time 0.0
#
set udp_(8) [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp_(8)
set null_(8) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(8)
set cbr_(8) [new Application/Traffic/CBR]
$cbr_(8) set packetSize_ 512
$cbr_(8) set rate_ 100kb
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150
$cbr_(8) set random_ 1
$cbr_(8) attach-agent $udp_(8)
$ns_ connect $udp_(8) $null_(8)

```

```
$ns_ at 0.0 "$cbr_(8) start"  
#  
# 4 connecting to 6 at time 0.0  
#  
set udp_(9) [new Agent/UDP]  
$ns_ attach-agent $node_(9) $udp_(9)  
set null_(9) [new Agent/Null]  
$ns_ attach-agent $node_(0) $null_(9)  
set cbr_(9) [new Application/Traffic/CBR]  
$cbr_(9) set packetSize_ 512  
$cbr_(9) set rate_ 100kb  
# rate is changed according to the scenario as 25, 50, 75, 100, 125, 150  
$cbr_(9) set random_ 1  
$cbr_(9) attach-agent $udp_(9)  
$ns_ connect $udp_(9) $null_(9)  
$ns_ at 0.0 "$cbr_(9) start"  
#  
#Total sources/connections: 5/10  
#
```

## Appendix B: AWK Script File

### Appendix B.1: Example of AWK Script File (statisticsO10.awk)

```
BEGIN {
netSize = 10; # changed according to the scenario as 10, 20, 30, 40, 50, 60

packet_sent[netSize] = 0;
packet_drop[netSize] = 0;
packet_recvd[netSize] = 0;
packet_forwarded[netSize] = 0;

# Initial Energy assigned to each node in Joules

energy_left[netSize] = 1000.000000;

total_pkt_sent=0;
total_pkt_recvd=0;
total_pkt_drop=0;
total_pkt_forwarded=0;
pkt_delivery_ratio = 0;
total_hop_count = 0;
avg_hop_count = 0;
overhead = 0;
start = 0.000000000;
end = 0.000000000;
packet_duration = 0.000000000;
rcvnum = 0;
delay = 0.000000000;
sum = 0.000000000;
i=0;
total_energy_remains = 0.000000;
idle_energy_consumed = 0.000000;
trans_energy_consumed = 0.000000;
recesion_energy_consumed = 0.000000;

idleEn[netSize] = 0
transEn[netSize] = 0
recesionEn[netSize] = 0

fail = 0
}

{

# performance metrics and parameters are obtained from trace file
```

```

state      =    $1;
time      =    $3;
node_num   =    $5;
energy_level =    $7;
idleE = $9
transE = $13
recesionE = $15

```

```

node_id    =    $3;
level      =    $4;

```

```
# Energy Consumption
```

```

if(state == "N") {
    for(i=0;i<netSize;i++) {
        if(i == (node_num)) {

                                energy_left[i] = energy_level;

        }
    }
}

```

```

if(state == "N") {
    for(i=0;i<netSize;i++) {
        if(i == (node_num)) {

                                idleEn[i] = idleE;

        }
    }
}

```

```

if(state == "N") {
    for(i=0;i<netSize;i++) {
        if(i == (node_num)) {

                                transEn[i] = transE;

        }
    }
}

```

```

if(state == "N") {
    for(i=0;i<netSize;i++) {

```

```

        if(i == (node_num)) {
                                recessionEn[i] = recessionE;
                                }
                                }
}
}

END {

for(i=0;i<netSize;i++) {

printf("%d %.6f \n",i, energy_left[i]) > "energyleft.txt";
printf("%d %.6f \n",i, idleEn[i]) > "idle.txt";
printf("%d %.6f \n",i, transEn[i]) > "transition.txt";
printf("%d %.6f \n",i, recessionEn[i]) > "recession.txt";

if ( energy_left[i] == 0.0) { fail++; }
total_energy_remains = total_energy_remains + energy_left[i];
idle_energy_consumed = idle_energy_consumed + idleEn[i];
trans_energy_consumed = trans_energy_consumed + transEn[i];
recesion_energy_consumed = recesion_energy_consumed + recessionEn[i];

}
# Calculation of the performance metrics
printf("%.6f\t", (((netSize-fail)*1000)-total_energy_remains)/(netSize-fail));
printf("%.6f\t",idle_energy_consumed/(((netSize-fail)*1000)
total_energy_remains)*100);
printf("%.6f\t",trans_energy_consumed/(((netSize-fail)*1000)-
total_energy_remains)*100);
printf("%.6f\n",recesion_energy_consumed/(((netSize-fail)*1000)-
total_energy_remains)*100);

}

```

## **Appendix C: Sample Result for 10 MNs for AODV**

### **Appendix C.1: Energyleft.txt File**

<b>Node Id</b>	<b>Energy Left</b>
node_0	837.252735
node_1	831.464173
node_2	843.568301
node_3	839.612220
node_4	869.583457
node_5	838.534912
node_6	833.509074
node_7	871.174524
node_8	829.665147
node_9	860.050160

### **Appendix C.2: Idle.txt File**

<b>Node Id</b>	<b>EC of MNs in Idle State</b>
node_0	68.459557
node_1	63.986137
node_2	68.582575
node_3	68.855335
node_4	81.461965
node_5	68.141116
node_6	65.446275
node_7	77.129820
node_8	63.432043
node_9	76.576627

### **Appendix C.3: Reception.txt File**

<b>Node Id</b>	<b>EC of MNs in Rx State</b>
node_0	65.504246
node_1	88.777316
node_2	80.255200
node_3	68.062478
node_4	36.623233
node_5	71.396534
node_6	81.569665
node_7	49.200531
node_8	88.954927
node_9	54.010031

### **Appendix C.4: Transition.txt File**

<b>Node Id</b>	<b>EC of MNs in Tx State</b>
node_0	28.783462



node\_1 15.772374  
node\_2 7.5939250  
node\_3 23.469967  
node\_4 12.331346  
node\_5 21.927438  
node\_6 19.474987  
node\_7 2.4951250  
node\_8 17.947883  
node\_9 9.3631820

### **Appendix C.5: Energy10.txt File**

<b>Average EC</b>	<b>Idle EC Ratio</b>	<b>Tx EC Ratio</b>	<b>Rx EC Ratio</b>
154.558530	45.424310	10.297697	44.277994