Al albayt University

Prince Hussein bin Abdullah Faculty of Information Technology

Department of Computer Science

On Using Route Cache in Route Discovery Mechanism in Mobile Ad-hoc Network

في استخدام المسارات المخزنة في الذاكرة السريعة في آلية استكشاف المسارات في الشبكات الخاصة المتحركة

Prepared by

Mazeed Taha Awad Khazalah 0920901003

Supervisor

Dr. Ismail Ababneh

Co-Supervisor

Dr. Venus Samawi

A thesis submitted to the information technology College of Al albayt University in partial fulfillment of the requirements for the degree of Master in Computer Science.

1st Semester

2012 / 2013

Al albayt University

Prince Hussein bin Abdullah Faculty of Information Technology

Department of Computer Science

On Using Route Cache in Route Discovery Mechanism in Mobile Ad-hoc Network

في استخدام المسارات المخزنة في الذاكرة السريعة في آلية استكشاف المسارات في الشبكات الخاصة المتحركة

prepared by

Mazeed Taha Awad Khazalah **0920901003**

Supervisor

Dr. Ismail Ababneh

Co-Supervisor

Dr. Venus Samawi

Discussion committee members	**Approved**
Dr. Ismael Ababneh	………..............
Dr. Venus Samawi	………..............
Dr. Khaled Batiha	………..............
Dr. Wesam Al Mobaideen	………..............
Dr. Saad Bani Mohammad	………..............

A thesis submitted to the information technology College of Al albayt University in partial fulfillment of the requirements for the degree of Master Computer Science.

Discussed and Recommended (Success / Modify / Reject) on: ……………………

1st Semester

2012 / 2013

Acknowledgements

Those who do not thanks the people do not thank Allah. At first, praise and gratitude towards Allah. I want to thank and appreciation my supervisors who have oversaw my thesis, Professor Ismail Ababneh, and Dr. Venus Samawi for their advices that guides me during the master thesis preparation. They were the source of my inspiration. I derived both insistence and determination from them despite of all the challenges that I encountered. I want also to express thanks for their intensive remarks on the thesis write up, as they have helped develop my writing skills and have also helped me to present my research in a much better way.

I want to thank my parents and all my family for motivating me to pursue higher education. I want also to present deeply thanks to all my friends and colleagues for their continued support and hopeful to complete my thesis on time. Too many thanks goes to Engineer Mohsen Al Khaldi for being helpful tutor who helped me to overcome the obstacles and a lot of technical difficulties.

Finally, special word of sincere thanks goes to my life partner, to whom carrying out my mistakes, and to whom take care of our children while I was busy in my studies most of the time.

Mazeed T. Khaza'lah

Table Of Contents

Content **Page**

Acknowledgments… I
Table of Contents… II
List of Tables… IV
List of Figures… V
List of Abbreviations… VII
Abstract… VIII

Chapter One: Introduction…

1.1 Overview… **1**
1.2 Problem Statement… **2**
1.3 Literature Review… **3**
1.4 Research Objectives… **3**
1.5 Thesis Organization… **4**

Chapter Two: Mobile ad hoc networks (MANET)…

2.1 Overview… **5**
2.2 MANET Definition… **5**
2.3 Applications of MANETs… **6**
2.4 Characteristics of MANETs… **7**
2.5 Routing Management in MANET… **8**
  2.5.1. Proactive Routing Protocols (Table-Driven protocols)… **9**
  2.5.2. Reactive Routing Protocols(on-demand routing protocols)… **10**
  2.5.2.1. Dynamic Source Routing (DSR) protocol… **10**
    a) Route Discovery Mechanism… **12**
    b) Route Maintenance Mechanism… **13**
Chapter Three: Related Work…**15**

Chapter Four: Design And Implementation Of Enhanced-DSR (EDSR)…

4.1 Problem Statement… **20**
4.2 Research Questions… **22**

4.3 Enhanced DSR………………………………………………………………………. 23
4.4 Design of EDSR…………………………………………………………………….. 23
  4.4.1. Module One: EDSR Route Discovery Mechanism…………………………... 24
    a) Design Of Packet Structure For EDSR Route Discovery Process………….. 25
    b) Processing EDSR Route Request…………………………………...……. 25
    c) Processing EDSR Route Reply……………………………………………... 26
  4.4.2. Module Two: Design Of EDSR Caching Module………………..……….. 27
    a) Caching Criteria……...…………………………………………………… 27
Chapter Five: Simulation Environment…………………………………………………
5.1 Simulation Environment……………………………………………………….. 29
5.2 Simulation Tools and Parameters……………………………………………… 29
5.3 Evaluation Metrics……………………………………………………………... 32
5.4 Scenarios Design………………………………………………………………. 33
    a) Scenario one………………………………………………………………… 33
    b) Scenario two………………………………………………………………… 34
    c) Scenario three……………………………………………………………….. 34
Chapter Six: Results Assessment…………………………………………………………
6.1 Threshold selection……………………………………………………………. 36
6.2 Evaluation Results: Packet Delivery Ratio…………………………………….. 40
6.3 Evaluation Results: Control Overhead………………………………………… 47
6.4 Evaluation Results: Average End-to-End……………………………………… 53
6.5 Evaluation Results: Throughput……………………………………………….. 58
6.6 Summary Of Evaluation Metrics Results……………………………………… 60
  6.6.1. Packet Delivery Ratio…………………………………………………...
  6.6.2. Control Overhead…………………………………………………….
  6.6.3. Average End-to-End delay……………………………………………
Chapter Seven: Conclusion and Future Works...……………………………………………
7.1 Conclusion……………………………………………………………………… 64
7.2 Future Works………...…………………………………………………………. 66
References …………………………………………………………………………… 67
Appendices …………………………………………………………………………... 70
Arabic Abstract……………………………………………………………………… 73

List Of Tables

Table 4.1. RREP header format……………………………………………………………25

Table 5.1. Simulation parameters ……………………………………………………………30

Table A.1. Route Request header format ……………………………………………………….70

Table A.2. Route Reply header format ………………………………………………………….71

Table A.3. Route Error header format ………………………………………………………….71

List Of Figures

Figure 2.1. An example of MANET…………………………..……………………………..6

Figure 2.2. Generic Multi-hop Wireless Communications……………………………..………6

Figure 2.3. Topology change in MANETs…………………………..……………………7

Figure 2.4. Classification of MANET protocols ………………………………..………………...9

Figure 2. 5. Unidirectional links …………………………..……………………………11

Figure 2.6. Procedure for DSR route discovery process ………………………………………...12

Figure 2.7. Automatic route shortening…………………………...………………………...14

Figure 4.1. A case of problem statement ……………………………………………………..21

Figure 4.2. General module for Enhanced-DSR ………………………………………………...24

Figure 4.3. Module one (EDSR Route Discovery (RREQ& RREP))……………………………26

Figure 4.4. Pseudo-code for route discovery process……………………………………………....27

Figure 4.5. Module two (EDSR caching management) ……………………….………………..28

Figure 6.1. PDR of 5 sources each sends 1pbs……………………………………….............37

Figure 6.2. Control Overhead of 5 sources each sends 1pbs…..………………………………...38

Figure 6. 3. PDR of 15 sources each sends 4pbs………………………………………………...38

Figure 6.4. Control overhead of 15 sources each sends 4pbs…….……………………………..39

Figure 6. 5. PDR of 5 sources each sends 1pbs………………………………………………40

Figure 6.6. PDR of 5 sources each sends 2pbs……………………………...………………...41

Figure 6.7. PDR of 5 sources each sends 4pbs……………………………………………......42

Figure 6.8. PDR of 10 sources each sends 1pbs……………………………………………43

Figure 6.9. PDR of 10 sources each sends 2pbs……………………………………………44

Figure 6.10. PDR of 10 sources each sends 4pbs…………………………………………......44

Figure 6.11. PDR of 15 sources each sends 1pbs…………………………………………......45

Figure 6.12. PDR of 15 sources each sends 2pbs.…………………………………………46

Figure 6.13. PDR of 15 sources each sends 4pbs…………………………………………….....46

Figure 6.14. Control overhead of 5 sources each sends 1pbs………………………………........47

Figure 6.15. Control overhead of 5 sources each sends 2pbs………………………………........48

Figure 6.16. Control overhead of 5 sources each sends 4pbs………………………………........49

Figure 6.17. Control overhead of 10 sources each sends 1pbs…………………………………..50

Figure 6.18. Control overhead of 10 sources each sends 2pbs…………………………………..50

Figure 6.19. Control overhead of 10 sources each sends 4pbs…………………………………..51

Figure 6.20. Control overhead of 15 sources each sends 1pbs…………………………………..52

Figure 6.21. Control overhead of 15 sources each sends 2pbs…………………………………..52

Figure 6.22. Control overhead of 15 sources each sends 4pbs…………………...…………...53

Figure 6.23. Average end-to-end delay of 5 sources each sends 1pbs……………...………...54

Figure 6.24. Average end-to-end delay of 5 sources each sends 2pbs………………...……...55

Figure 6.25. Average end-to-end delay of 5 sources each sends 4pbs………………….........55

Figure 6.26. Average end-to-end delay of 10 sources each sends 1pbs…………………........56

Figure 6.27. Average end-to-end delay of 10 sources each sends 2pbs………………...…......57

Figure 6.28. Average end-to-end delay of 10 sources each sends 4pbs………………...…......57

Figure 6.29. Average end-to-end delay of 15 sources each sends 1pbs…………………........58

Figure 6.30. Average end-to-end delay of 15 sources each sends 2pbs…………………........59

Figure 6.31. Average end-to-end delay of 15 sources each sends 4pbs…………………........59

Figure 6. 32. Throughput of 5 sources each sends 1pbs……………………………………....60

List Of Abbreviations

ABR	Associativity Based Routing
ACK	Acknowledgment
AODV	Ad Hoc On Demand Distance Vector
CBR	Constant Bit Rate
DSR	Dynamic Source Route
EDSR	Enhanced DSR
EEDSR	Energy Efficient DSR
GLOMOSIM	Global Mobile Information Systems Simulation
IEEE	Institute Of Electrical And Electronic Engineers
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MDSR	Modified DSR
NS-2	Network Simulator-2
PPS	Packets Per Second
PDR	Packet Delivery Ratio
RERR	Route Error
RFC	Request For Comments
RREP	Route Reply
RREQ	Route Request
RWP	Random-Waypoint
SSR	Signal Stability Routing
TORA	Temporally Ordered Routing Algorithm
UDP	User Datagram Protocol

Abstract

Dynamic Source Routing (DSR) is one of the Ad hoc routing protocols. Propagation of route discovery and route maintenance packets along the network is costly. Route caching is used to speed up route discovery and to reduce propagation of route requests. When an intermediate node receives route request packet, it replies from its cache if it has a cached route to the destination. Although route caching enhances the performance of DSR protocols, but stale cache entries will lead to performance degradation, when we are near the destination. In this case, the probability of obtaining a fresh route to the destination is higher without using route cache.

The question is, when to use the cache, and when to stop using it. In other words, what is the suitable threshold value that makes using cache more effective. To answer this question, we carried out an empirical study to determine suitable threshold value that supports the decision of using route cache. Based on this study, it was detected that half of the network diameter is the most suitable threshold value.

Another feature, concerning route caching, is the ability of caching any heard routes including multiple routes per destination. This feature may cause a problem since cache has limited size. We investigate that suitable cache size that caches the most fresh routes instead of caching any routes without using any caching criteria. The question is, which route should be cached and which route should be removed. To answer this question, we keep only two routes for specific destination. These two routes are selected based on two caching criteria: freshness, and hop count. Initially, we ensure that the route satisfies freshness criterion, then the comparison is done based on hop count criterion. We achieved freshness criterion by selecting the route that is generated by complete broadcasting of route request.

GLOMOSIM simulator is used to evaluate the suggested Enhanced DSR (EDSR). In simulation, we used several parameters for evaluating the performance of the algorithms: pause time, number of CBR (Constant Bit Rate) sources, and transmission rate. The pause time represents the amount of time a node stays without moving. The number of CBR sources represents the number of sources sending data packets to their destinations. We have evaluated EDSR in comparison with Basic DSR for different three scenarios, each has three cases. The scenarios randomly chosen by permutation these three parameters. Simulation results show significant improvement over Basic DSR up to 22% in packet delivery ratio, up to 63% in control overhead, and up to 27% in average end-to-end delay.

Chapter 1
Introduction

1.1. Overview

During the last decades, portable computing and wireless devices (such as cell phone, laptops… etc) are invented and widespread. This widespread uprising affects the field of information technology, and substitute the platform-based applications by network-based applications. Users became capable to perform many information activities at the same time from the same place. Mobile devices characteristics (such as small, cheap, fast… etc) made wireless network to be easier way to exchange information among interconnected nodes. Wireless networks have two types; infrastructure and infrastructure-less networks. These two types are specified depending on using infrastructure or not. Mobile Ad-hoc Networks (MANETs) are a type of infrastructure-less networks *(I. Chlamtac et.al, 2003).*

A MANET is a collection of mobile nodes connected by wireless links. These mobile nodes are able to move independently in any direction, change their links to other devices frequently, and work as virtual routers by forwarding traffic not related to their own use. Forwarding of such packets is handled using multi-hop wireless communication. Multi–hop wireless communication means that there can be multiple hops in the route between a source and a destination. If any node needs to communicate with another node out of its communication range, then it uses intermediate nodes to rely the message hop-by-hop. Using multi-hop wireless communication means that MANETs do not need infrastructure in order to transfer information between nodes. MANETs may operate by themselves or may be connected to the larger Internet *(J. Broch et al, 1998), (I. Chlamtac et.al, 2003).*

The network topology of a MANET is frequently changed due to node mobility and limited transmission range. So that maintaining routes in MANETs is a big challenge. Several routing protocols have been proposed to handle this problem. MANET Routing protocols are classified mainly into three types: proactive, reactive and hybrid. In contrast to proactive protocols that requires frequent update of network information, reactive protocols maintain routes only when needed. Therefore, reactive protocols reduce the overhead and routing load, but increase the transmission delay. Hybrid routing protocols combine the advantages of both of the other two types *(Perkins et al, 2001)*. Different reactive protocols have been proposed. The most popular reactive protocols are: Dynamic Source Route (DSR), Ad hoc On Demand Distance Vector (AODV), Associativity-Based Routing (ABR), look for Signal Stability Routing (SSR), and Temporally Ordered Routing Algorithm (TORA). Several on-demand protocols employ a route cache that records links that this node has learned or has previously discovered. In this work, the concentration is on DSR protocol.

1.2. Problem Statement

DSR is a reactive protocol (*Johnson et al, 1996*) designed specifically for ad-hoc networks. As other on-demand protocols, DSR has two mechanisms for maintaining routes: ***Route Discovery*** and ***Route Maintenance***. Several features are used to optimize DSR performance. These features include caching links learned by overhearing, caching multiple routes per destination, and replying from cache by intermediate nodes. Cache entries may become invalid due to nodes moving out of wireless transmission range of each other. An invalid route reply from an intermediate node that contains a stale route can be very costly in terms of time and bandwidth *(Biswal, et.al, 2011)*. Using a cache that contains a misleading route increases the probability of having a broken link during the data sending process (*Johnson, et.al, 1996),(RFC4728).*

Aggressive caching of overheard routes can considerably improve performance. Unluckily, spreading cached routes can significantly increase the possibility of cache cross-pollution, since stale routing information in one node's cache can be cached by other nodes. Even when a node has really learned that a link is broken, it is still possible for that node to once more hear this stale information (*Hu et al, 2002).*

As declared above, using stale cached routes can degrade performance-rather than optimize it. Therefore, we need to use a mechanism to determine when we can use cached entries. DSR needs to be developed to solve problems such as: when a node that receives a route request should respond with a route picked from its cache, and when to broadcast the route request. Another problem concerning route caching is the ability of caching multiple routes per destination. Aggressive caching of heard routes without filtering means that consumed of limited cache size.

1.3. *Literature Review*

Many researchers have given attention to cache problems. Different strategies have been proposed for solving these problems. (*Lou and Fang, 2002)* proposed an adaptive link cache to remove stale routes from cache. *(Hu and Johnson, 2002)* used an epoch number to sequence links at the sending side. They prevented any link that was invalid at a previous time from re-entering caches. (*Yu, 2006)* proposed a technique that used a cache table with no fixed size. In this technique only reachable nodes that had cached a broken link would be informed. It allowed DSR to quickly drop down stale routes from cache. (*Adane, 2008)* proposed an active communication energy efficient routing protocol (EEDSR) that aims to prolong network lifetime by reducing energy consumption and balancing energy load. The protocol uses a delay forwarding technique to solve the broadcasting storm problem. It discovers routes based on some energy metric. When an intermediate node has a cached route to destination, an intermediate node unicasts RREQ to destination using a cached route instead of unicast reply to source node. (*Rizvi et al, 2008)* proposed a scheme that used a reverse direction search method. They propose that when an error occurs, the packet would be sent back to the previous node, and then this node attempts to find an alternate route in its cache. In this scheme, no feedback would be sent to the source node. **(***Neelakntapp et al, 2009)* proposed an adaptive route selection scheme for DSR that depends on node stability. The stability of a node is measured by the number of packets it has successively delivered to its neighbors.

1.4. *Research Objectives*

In this work, we present a new mechanism that restricts the use of cache during the route discovery process, and restricts aggressive caching of overheard routes. This mechanism aims to:

- Reduce the cost of route requests by reducing the use of stale cached routes (when the route request is near the destination). This mechanism uses an empirical threshold value. Nodes closer to the source (i.e. less than threshold distance) are allowed to reply from cache.
- Make cache entries as small as possible, to investigate the limited size for route cache. We achieved this aim by allowing nodes to cache only two routes per destination, and to

cache routes that came from complete Route Request broadcasts only to ensure caching fresh routes.

The general objective of this thesis is to improve the performance of DSR. In the course of this research, the performance of the basic DSR was analyzed via simulation based on several evaluation metrics: packet delivery ratio (PDR), control overhead, Average End-to-End delay and Throughput. Then, DSR was modified, as explained above, and the performance of the proposed Enhanced DSR was compared to that of DSR using those metrics.

1.5. Thesis Organization

The rest of the thesis is organized as follows:

In ***chapter two***, we discuss the background information to understand the subject matter of this thesis work. It consists of theoretical concepts for our work. This chapter covers Ad-hoc networks, mainly MANETs and MANET's routing protocols. This chapter is the corner stone of this work.

Chapter three addresses related research. This chapter is the foundation of this work's problem definition and includes the related work on using cached routes.

Chapter four discusses the main work of the thesis. In this chapter, we present the design and implementation of Enhanced-DSR in more depth. We discuss the mechanism of when to use cached routes, the nature of routes that are allowed to be cached, and the number of route(s) per destination that are allowed to be cached.

In ***chapter five***, we give the description of the system and the evaluation metrics that we used to evaluate our Enhanced-DSR.

In ***chapter six*** we presents the simulation results of Enhanced-DSR using GLOMOSIM-2.3, and compare it with Basic DSR based on the three evaluation metrics.

Chapter seven concludes the work of this thesis. We represent a conclusion, where we summarize our contributions and main findings.

Chapter two

Mobile ad-hoc networks (MANET)

2.1. Overview

During the last decades, high prevalence of wireless communication and portable computing devices (such as cell phone, laptops… etc) led to essential changes in telecommunication, data networking, and information technology field. Users became capable to perform many information activities at the same time from the same place. The nature of mobile devices (such as small, cheap, convenient… etc) made wireless network the easier way to exchange information among interconnected nodes (Chlamtac et.al, 2003).

Wireless networks are classified into two types: infrastructured networks, and infrastructure-less networks. In infrastructured networks, nodes can communicate only with base stations. A base station has a fixed location. It acts as coordinator for access to transmission channel(s) within its coverage area. Every infrastructured network has at least one base station (Adane, 2008). Setting up this type of wireless network takes time and can be costly (Chlamtac et.al, 2003). Infrastructure-less networks do not need any pre-established infrastructure. Nodes can communicate with each other without using any base station or network coordinator. So that each node acts as router and host at the same time. A MANET is an infrastructure-less network (Adane, 2008).

2.2. MANET Definition

A MANET is a collection of mobile nodes connected to each other by wireless links, as shown in Figure (2.1). These mobile nodes are capable of moving independently in any direction, and can act as virtual routers by forwarding traffic not related to their own use. Forwarding of packets is typically performed using multi-hop wireless communication. Multi–hop wireless communication means that there can be multiple hops in the route between source and destination. If any node needs to communicate with another node out of its communication range, it uses intermediate nodes to relay the message hop-by-hop *(Boyer et.al, 2001)*. As shown in Figure (2.2), there are **n** hops that need to transmit data from source to destination. Using multi-hop wireless communication dispenses MANET from using any infrastructure to transfer information between nodes *(Chlamtac et.al, 2003)*. MANET may operate by them-selves (self-configuring), or may be connected to the larger Internet.

Figure 2.1. An example of MANET

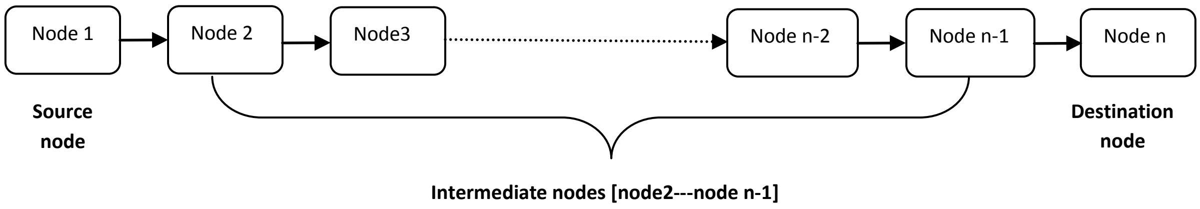


Figure 2.2. Generic Multi-hop Wireless Communications (J. Boyer, 2001)

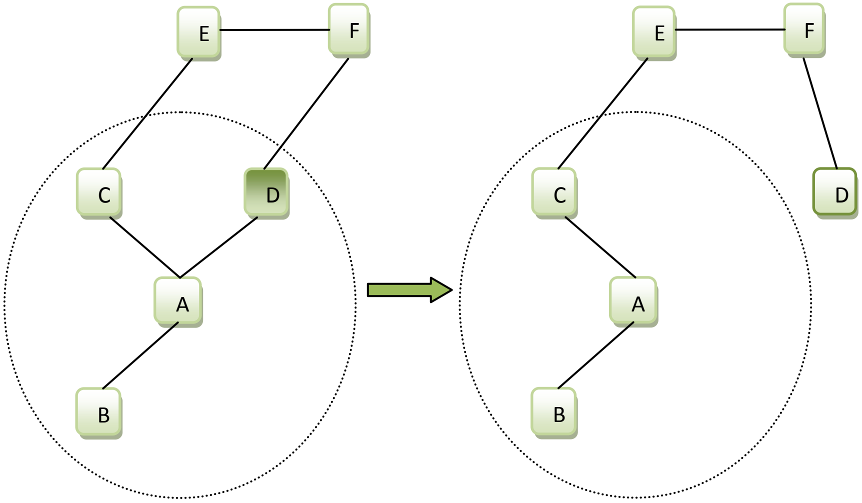
2.3. Applications of MANETs

Due to its flexibility, a MANET is attractive for several applications. MANETs are suitable in cases where infrastructure is hard or take a long time to establish. It is often used in commercial business environments which need to arrange collaborative computing outside offices rather than inside. MANETs are also suitable to provide catastrophic and emergency management services applications, such as in disasters and earthquakes, which need to reestablish communication quickly in areas where there is no infrastructure. Search and rescue activities widely use MANETs also. The military also uses MANET to perform their communications *(Chlamtac et.al, 2003).*

2.4. Characteristics of MANETs

MANET has several characteristics that discriminate it from other networks. These characteristics should be taken into consideration as guideline when designing new routing protocol. These characteristics are as follows:

1. ***Dynamic topology***: since mobile nodes move independently, network topology is frequently changed unpredictably. So that, each node can move out of range of another node. This cause link breakage as shown in the example illustrated in Figure (2.3). Initially, there is a direct link between node (A) and node (D) since (D) is within the range of (A) (as shown in Figure (2.3, a)). When node (D) moves out of node (A) range, the link between them is broken, and network topology becomes as shown in Figure (2.3, b). The network is still connected since node (A) can reach (D) through (C, E, and F). Keeping network consistence needs frequently up-to-date network information. This frequent updates imposes high degree of network overhead *(Adane, 2008).*



(a) (b)

Figure 2.3. Topology change in MANETs: nodes A, B, C, D, E, and F compose a MANET. The circle represents the propagation range of node A. initially, the network has the topology in (a). When node D moves out of A's range, the network topology changes to the one in (b) (Zhou and Haas, 1999).

2. ***Transmission range constrained:*** each node has a limited radio transmission range that allows it to communicate directly with those nodes within its transmission range. But when a node needs to communicate with another node out of its range, it uses multi-hop communication *(Chlamtac et.al, 2003).*

3. ***Energy-constrained*****:** a node in a MANET has a battery. Since each node acts as host and router at the same time, the node acts as a receiver all the time. Therefore, batteries are quickly consumed. When the receiver is off, then the node is not a member of the MANET, which means that the network is partitioned *(Adane, 2008)*. Designing a routing protocol should consider reducing energy consumption.

4. ***Network security*****:** security must be taken into account depending on the application that uses the MANET. For applications that need a good security, such as hostile environment, security mechanisms must be applied to make useful use of MANETs as in wired networks, which have good security *(Adane, 2008).*

5. ***Variation in link and node capabilities*****:** nodes may differ in most capabilities such as processing capabilities. In addition, due to variation in node transmission capabilities, asymmetric links may result. This asymmetric links needs protocols to manage the heterogeneity of networks *(Chlamtac et.al, 2003).*

2.5. Routing Management in MANET

A MANET can be categorized based on various parameters such as: symmetric and asymmetric links, traffic characteristics, routing methods, and some other metrics such as time and reliability constraints.

MANET topology is frequently changed due to node mobility. Therefore, maintaining routes in such network is a big challenge. Several routing protocols have been proposed to deal with this problem. MANET routing protocols are categorized in general into three types based on several parameters such as route discovery cost, or information update mechanism for network. These three types are ***proactive***, ***reactive and hybrid*** protocols (see Figure 2.4).

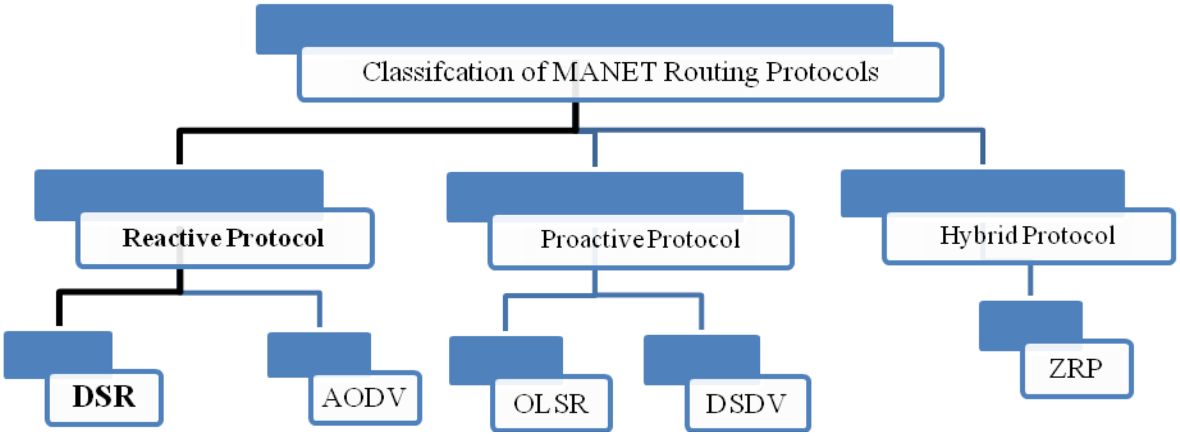


Figure 2.4. Classification of MANET protocols

The basis of the routing problem is answering the question: How do I **best** get from here to there? Networks may be static or dynamic. Routing in static networks is so simple. We can form a direction table for each node. This table is established once for each node. However, the problem occurs in dynamic networks, where links can be broken and network topology changes in unpredictable ways, such as in MANETs (Levchenko et.al, 2008).

Due to its special characteristics (such as node mobility, frequent topology changes, energy constraints, low channel bandwidth and high channel error rates…. etc), a MANET needs to special autonomous protocols to maintain routes between senders and receivers. Maintaining routes in such networks is a big challenge since protocols developed for wired networks do not work well in MANETs or they may give wrong results (Adane, 2008). Several routing protocols have been proposed to maintain routes in MANETs. These protocols could be classified mainly into three types based on several characteristics as mentioned above.

2.5.1. Proactive Routing Protocols (Table-Driven protocols)

In proactive routing protocols, each node stores the most recent information for routes to any node in the network. This information is stored in one table or more (Gani et al, 2009). Keeping a consistent view of the network topology is the main goal of proactive protocols. This consistency is achieved by propagating information throughout the whole network. Two ways of information propagation are available: periodic propagation and event-driven propagation (triggering update) *(Chlamtac et.al, 2003)*. In periodic propagation, each node broadcasts its route table to the network periodically. However, in event-driven propagation, any change in a node's neighbors must be broadcasted to the network. The change may be a link breakage. Proactive protocols consume much energy and increase traffic control due to periodic broadcasts of route tables. In case of large networks (network with a large number of nodes), the table sizes become huge. This causes a challenge (Adane, 2008). The most popular proactive routing protocols are Destination-Sequenced Distance-Vector (DSDV), Optimized Link State Routing (OLSR), and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) (Gani et al, 2009).

2.5.2. Reactive Routing Protocols(on-demand routing protocols):

In contrast to proactive protocols, reactive protocols look for routes only when needed. These protocols reduce overhead and routing load as compared with proactive protocols. On the other hand, the transmission delay is increased (Das et al, 2001). Route discovery and route maintenance are two mechanisms that are used to obtain routes in reactive protocols. Before the route discovery process, a source node does not actually know the route to destination. Route maintenance proceeds after a route is broken to fix the error (Johnson, 2003).

The most popular reactive protocols are: Dynamic Source Route (**DSR**), Ad hoc On Demand Distance Vector (**AODV**), Associativity Based Routing (**ABR**), Signal Stability Routing (**SSR**), and Temporally Ordered Routing Algorithm (**TORA**) (Das et al, 2001). In our work, the DSR protocol is considered.

2.5.2.1. Dynamic Source Routing (DSR) protocol

DSR is a reactive protocol designed specifically for Ad-hoc network (Johnson et. al, 1996). Like other on-demand routing protocols, DSR establishes routes when a node has data to transmit to another node within the DSR network. Explicit **source routing** is a key distinguishing feature of DSR. Source routing means that a source node already knows the complete hop-by-hop route to the destination before actual data transmission. Using source routing achieves loop-free routing (since no duplicate nodes are allowed). It also avoids the need for periodic updates of routing information of intermediate nodes listed in the source route. The source route is carried in the data packet header. The other key optimizations of DSR over other on-demand protocols are the ability of handling unidirectional links (see Figure 2.5), support of internetworking between different types of wireless networks, and using the caching principle which allows routes to be cached in node caches *(Johnson et .al, 1996).*

DSR optimizes performance as compared with other on demand protocols ((Johnson et.al 1996), (Adane 2008). It limits used bandwidth by avoiding the periodic table updates. Overhead also is reduced in spite of topology changes. DSR does not have any periodic routing messages for neighbors. Therefore, in the worst case, the overhead cannot exceed what is needed to track a route currently in use *(Johnson et.al 1996).*

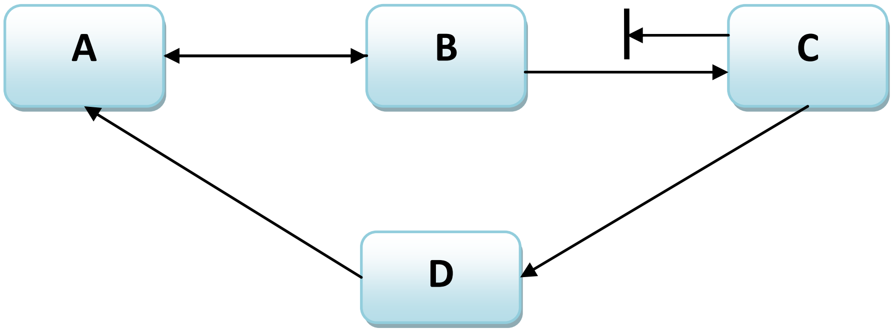


Figure 2.5. Unidirectional links: node A can transmit to node B and B to C, but node C cannot transmit to node B, it must use a different route to A.

DSR works well when the network's diameter is small or medium, since the complete source route is included in the packet header. When the network is large, the packet header can be longer than the packet payload (Johnson et al, 1996). The following sections present a brief description of the Route Discovery and the Route Maintenance mechanisms.

a) Route Discovery Mechanism

Route discovery is used when source node (**S)** has data to send to a desired destination (**D)** and no routing information is available at source node (**S).** Route discovery mechanism is used to find the appropriate route to reach destination.

As in the other on-demand routing protocols, Route Discovery utilizes two types of packets: **Route Request Packet (**RREQ**)**, and **Route Reply Packet (**RREP**).** The combination of source node IP (initiator), Request ID, and destination node IP (target) carried in the two packet types are unique throughout the network. In route discovery, when there is data to be sent, S will follow the steps in Figure (2.6) (Neelakantappa et al, 2009).

```
Checks the route cache
If (there exists a fresh enough path for desired destination)
{Use it}
Otherwise
{    Broadcast a route request (RREQ) packet  }
Intermediate or destination node receiving the RREQ checks
If (Itself is a destination)
{   Send a route reply (RREP)  }
Otherwise
{
If (there exists a fresh enough path for desired destination)
{Send RREP}
Otherwise
{Broadcast a route request (RREQ) packet) }}
```

Figure 2.6. Procedure for DSR route discovery process

In DSR, the process of returning RREP to source node is as follows: When the desired destination receives RREQ, it copies the route record of RREQ to the RREP packet. The desired destination then unicasts the RREP packet to the source node by reversing the route record provided that links are bidirectional. If links are unidirectional, a route discovery from destination to source is carried out so as to obtain a route back to the source node, and the source to destination route is piggybacked with the RREP packet.

Several additional route discovery features have been proposed. These include:

1. Caching Overheard Routing Information: Since packet header (of both RREQ and RREP) contains source route record from source to destination, any node may overhear the forwarded packets and then it may cache this route for future use. A single route discovery operation may generate many replies. These replies allow any node to learn and cache multiple routes to a specific destination. These multiple routes act as alternatives to each other when any route is broken during packet forwarding. Caching multiple routes for a specific destination reduces the overhead that results from regenerating a new RREQ each time a route breaks.
2. Replying to Route Requests using Cached Routes: Instead of performing a complete RREQ until reaching the destination, an intermediate node can respond from its own cache if it has a route to destination.
3. Preventing Route Reply Storms: Bandwidth waste and collision may occur due to responses from intermediate nodes. This can be handled by enabling promiscuous mode that allows a node to hear all packets. It does not reply if it hears a shorter path than it has. Each node delays its RREP for a random time.
4. Route Request Hop Limits: RREQ may be either non-propagating or propagating. In non-propagating mode, the route request hop limit is one hop. Therefore, non-propagating mode is un-expensive since D is either a neighbor of S or a neighbor of S has a route to D. If a non-propagated RREQ does not find a route to D, then a propagating RREQ is sent where the route request hop limit is unlimited.

b) *Route Maintenance Mechanism*

Route maintenance is used when a route to a destination **(D)** is obtained, and when a source node **(S)** actually sends a data packet to **D.** The only packet type for route maintenance is the **Route Error packet (**RERR**).** If any link on the route to the destination is broken, the node detecting this error will send RERR to **S**. At this time, **S** drops any routes containing this broken link from its cache. Each node that hears this RERR also drops down this broken link from its cache. **S** uses another cached route (if exists), or generates a new RREQ packet.

Several route maintenance features have been proposed and implemented in on-demand protocols in addition to the basic operation of route maintenance. These features include:

1. Packet Salvaging: This feature means that after a node discovers a link-failure, it sends RERR to S. Then it tries another cached route to D if one exists. Otherwise, the packet will be discarded. This node replaces the suffix of the packet source route with the salvaging node's cached route. It then transmits the packet to the destination. A counter that determines how many times a packet has been salvaged is maintained to avoid loops.
2. Automatic Route Shortening: Routes could be shorter than that in source route. This may happen when a node hears a packet in another order than that defined in the source route record. In this case, source could shorten the route. In Figure (2.7), the source route between S and D is [S, K, O, D]. If node O can hear node S when transmit packet to K, then O should declare S to shorten route to be [S, O, D].

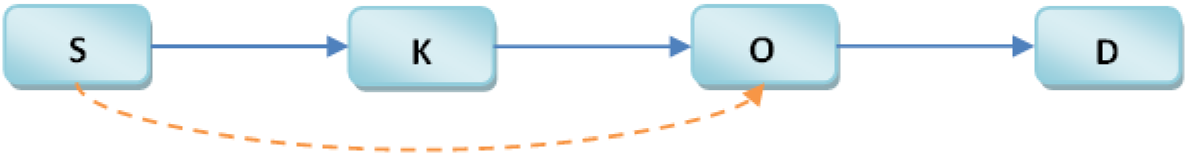


Figure 2.7. Automatic route shortening

3. Gratuitous Route Errors: RERR will be piggybacked in new RREQs that are generated after RERR.

Chapter Three

Related Works

Many researchers gave attention to caching management and maintaining routes in DSR. This section presents a brief description of some optimizations that have been proposed to handle caching and route discovery in DSR.

Lou and Fang (2002) studied the effects of using two caching strategies with one of on-demand routing protocol, DSR. These two strategies are path cache and link cache.

In path cache, full path between specific source and specific destination is cached separately. It is easy to implement. In link cache, all links between two nodes in the view point of specific node are cached separately. Graph data structure is used to represents link cache. In link cache, the route returned to source node is composed of individual links that lead at all to the destination.

In their work, They focused on link cache since it is rarely studied. Then they proposed a new scheme to remove stale routes from cache in DSR. This scheme was called Adaptive Link Cache. The aim of this scheme is tracking the optimal link lifetime under different mobility levels. The scheme is composed of link cache and timer-based stale link expiry mechanisms. They have studied the effects of different link lifetimes on the performance of the routing protocol.

In their work, they uses three parameters to estimate link life time. These three parameters represent the times where link entered cache, the last time that this link is used, and the expected time that this link may live. Link may be removed from link cache either when it’s time to live is expired, or when node received a RERR indicating that this link is broken. After link removed from link cache, the statistical link life time is collected using these three parameters.

A comparison between this scheme that uses link cache and path cache structure was performed in their work. Using the GLOMOSIM simulator, they showed that using Link cache without a time-out mechanism decreases the performance of DSR, while using Adaptive Link Cache enhances the overall network performance by reducing overhead in the case of high traffic load. The weakness of this work is using a heuristic in link lifetime prediction.

Hu and Johnson (2002) proposed a new approach that aims to reduce the invalid cache information spread through overhearing of invalid links in protocols such as DSR. They achieved this goal by preventing any link that was invalid at a previous time from re-entering caches.

They used an epoch number to sequence links at the sending side. This sequencing is generated depending on route error messages. After each route request for any link followed by a route error, the epoch number of this link is incremented by one. When two links conflict at cache, the node should choose link with higher epoch number.

They analyzed this approach theoretically, and they proved that this approach doesn't directly increase the packet overhead. However, they proposed approach ensures that stale information cannot override fresh information.

Yu (2006) gave attention to the staleness cache problem. He proposed a new distributed, adaptive, proactive on-demand algorithm that maintains a new cache structure called cache-table. The cache table maintains the information needed to update cache entries. This cache-table does not have a fixed size; it increases as a new route is discovered, and decreases as an invalid route is dropped. Each cache-table contains two types of information. The first information type is the arrangement of node in route (downstream or upstream). The second one determines which neighbors have determined the link through route reply.

Provided that each node stored all neighbors that cached that link in its cache. A technique of updating cache proactively was used. In this technique, only reachable nodes that had cached a broken link are informed (i.e., the algorithm guarantees informing all reachable nodes that had cached a broken-link, therefore it allowed DSR to quickly drop down stale route from cache). Using NS-2 simulator, simulation results show that the proposed algorithm improves overhead, delivery latency, and packet delivery ratio in both promiscuous and non-promiscuous mode.

Promiscuous mode means that the interface of the node are allowed to deliver every received packet to the network driver software without any destination address filtering. Non-promiscuous mode node does not allowable to deliver any packet without filtering it based on destination address *(RFC4728).*

Adane (2008) proposed an active communication energy efficient routing mechanism as improvement of routing protocols. This mechanism is implemented with DSR (any other routing protocol could implement it). It is named Energy Efficient Dynamic Source Routing (EEDSR). This protocol aims to reduce energy consumption/packet and to balance energy load. This may lead to long network lifetime. It also aims to increase packet delivery ratio (PDR).

In this mechanism, routes are discovered based on energy metric. Unlike DSR, which uses a fixed transmission power, EEDSR uses a suitable transmission power based on link cost. Route cost is included in the RREQ header to determine the path with least path cost. The destination node replies to RREQ with the least path cost.

This mechanism implements a delay forwarding technique, where each node waits for a certain period before rebroadcasting the RREQ. Therefore, it presents a solution to the broadcast storm problem. As in DSR, an intermediate node is allowed to reply from its cache. However, in EEDSR, an intermediate node unicasts RREQ to destination using a cached route instead of unicast reply to source node. This may increase route freshness.

NS-2 simulator was used to evaluate the suggested mechanism performance and to achieve work objectives. The simulation results show that, EEDSR outperforms the Basic DSR, in limited battery energized nodes. Also, EEDSR improves the network lifetime in comparison with DSR. In addition, EEDSR has better PDR than DSR.

Rizvi et al (2008) suggested a new scheme that modifies both route discovery and route maintenance in the DSR protocol. They commented on the behavior of DSR in the case of link failure.

This behavior indicates that, When any link goes down, DSR protocol transmits back RERR packet to the source node to generate new RREQ packet for the same data packet. the node that triggering error locates an alternate route (if exist) to salvage packet or discard it (if not exist). Rizvi et al believe that this behavior degrades the overall network performance. Therefore, a new scheme was proposed.

This scheme aims to minimize the overhead corresponding to frequent packet retransmission, and to maximize network throughput. In the proposed scheme, a reverse direction search method was used. In the reverse direction method, when an error occurs, the packet would be sent back to the previous node, which tries to find an alternate route in its cache.

In this scheme, no feedback would be sent to source node. They supported their hypothesis using a mathematical model. They proved that their scheme improves time delay and number of packet loses.

Gani et al (2009) present a modified DSR protocol (MDSR). They benefit from using the ACK reply path as a backup route when the original route is broken. Two flags are added to the RERR packet. The first flag is to determine beginning a local route repair. The second flag is used to check if this local repair is done successfully. They used a TCP_BUS method to avoid the problem of unnecessary resending of lost data packet.

The source node uses the ACK path until the original path is repaired successfully. Otherwise, a new route discovery must be performed. A source node doesn't use the ACK path for a long time because it tries to find another route that may be better than the ACK path.

NS-2 simulation results showed that using MDSR reduces the average end-to end delay, and it increases the packet delivery rate as compared with DSR.

Neelakntapp et al (2009) proposed a new adaptive route selection scheme for DSR depending on node stability and hop count.

The stability of a node is measured by the count of packets that it has successfully delivered to its neighbors. More stable routes are selected using only node history record. When source node receive more than one route for the destination, it calculates a value (V) depending on the selection criteria for the received routes. The route with the smallest value (V) will be selected to send data.

Using the GLOMOSIM simulator, they found that using the adaptive route selection scheme suffers less extra routing overhead, and much less average packet delivery delay with better throughput as compared with the traditional DSR.

Biswal et al (2011) studied the route reply problem of DSR. They present an improved Route Reply (RREP) method. The problem statement that this work has studied is high collision probability one-hop away from the source node. This collision occurs due to the congestion of multiple RREPs for the same RREQ at about same time. They added a delay time for an RREP packet when it is one-hop away from source node.

NS-2 simulator was used to evaluate the suggested scheme. The simulation results indicate that this scheme is better than the traditional DSR in decreasing the probability of collision.

Chapter Four

Design and Implementation of Enhanced-DSR (EDSR)

Existing version of DSR has several criticisms on using caching during route discovery mechanism. In this chapter, we will discuss these criticisms and then we will discuss the design of EDSR protocol which proposed in this work to deal with these criticisms.

4.1. Problem Statement

Several features have been added to the first version of DSR protocol to improve its performance. The features that concerns route caching are: caching any links received or overheard from forwarded packets, caching multiple routes for specific destination, and the ability of reply from cache of both destination and intermediate nodes.

Any two nodes may move out of wireless transmission range of each other. Cache entries may become invalid (stale) quickly due to node mobility. Staleness cache entries problem occurs because a node is not notified when one of its cache entries became invalid until it uses this entry to send packets. Using invalid routes may increase the probability of lost packets during data sending process. So that, route reply from an intermediate node’s cache that may contain stale routes can be very costly in terms of time and bandwidth (Biswal, 2011).

In Basic DSR, any overheard routes that spread to the whole network may be cached without using any criteria to filter these routes. This feature can upgrade network performance. unfortunately, stale routing information in one node's cache can easily cached by another nodes. Although a node may be notified that a link is broken, it is still possible for that node to cache a route that contain this broken link if it hear it once more. This means that staleness problem still exists even with link failure notification. Such routes spreading can significantly increase the possibility of cache pollution (Hu and Johnson, 2002).

In term of reaching a broken link, Basic DSR reacts in the procedure that discussed in section 2.5.2.1(b). This procedure may increase network overhead and delivery latency if new route discovery process is needed to be generated to handle link failure (Johnson and Maltz, 1996).

Another issue corresponding to caching is the limited cache size. If any routes is aggressively cached, we do not investigate cache size. It may consumed by caching invalid entries.

In Basic DSR route discovery process, source node broadcasts the RREQ packets to its neighbors. Provided that DSR protocol has a neighbor table used to keep track of neighbors. As each node has X neighbors, along route from source to destination, there are Y hops. In this case, the total number of RREQ packets broadcasted from source to destination (R) equal to X multiply by Y RREQ packets. This number of RREQ packets broadcasting for each route discovery process results a higher degree of overhead in Basic DSR(*Sultana et.al, 2010).*

Sometimes, when we are near the destination, reply from cache is critical. Cached route may be stale. At this case, broadcasting of route request may catch fresher route than when using cached route. Consider the case shown in figure (4.1).

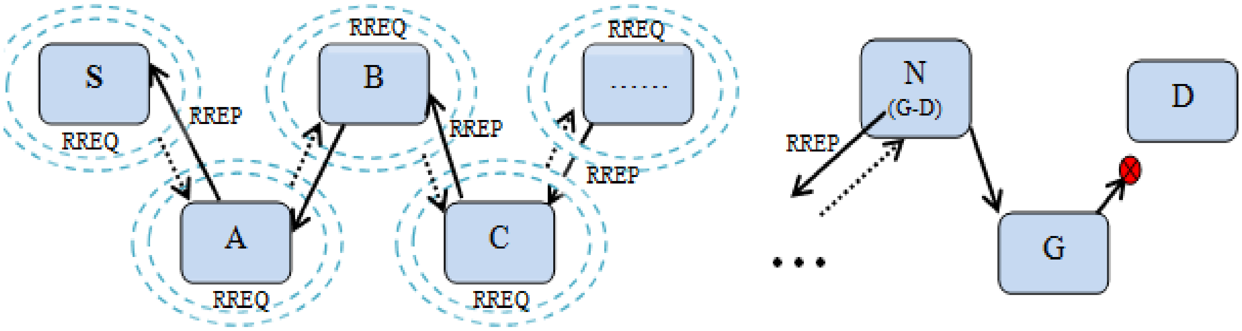


Figure 4.1. A case of problem statement

Source node (S) has data to send to destination node (D). At first, since S doesn’t have any route to D, it will broadcast RREQ packet containing (D, [S], id=1) to its neighbors. Each node receives this RREQ (A) adds its IP address to the route record and then broadcast it again, since it is not the destination (D). Then, node (B) and node (C) will receive the RREQ respectively, since none of them is the destination node, their IP addresses will be added to the route record, then re-broadcast RREQ. After six hops (for example), RREQ reaches node (N). Node (N) has a cached route to (D). This cached route contains [G-D]. Node (N) will return RREP to S containing its own cached route accumulated with that in the received route record (S,A,B,C,….,N,G,D). When source node receives this reply, it will begin to send its own data along route received in RREP packet. At reaching node (G), node (G) may discover that the link [G-D] is broken!!. G sends RERR to S containing this broken link. If node (G) has a cached route to node (D), then it will try to salvage packet. Otherwise, it discards the packet.

In this case, node (N) returned RREP of eight nodes route length containing two nodes from its cache. About three-quarters of the path length (six hops) have been passed before taking route from the cache of node (N). The aim of using cache in this case is reducing quarter of route discovery process cost (broadcast two RREQ from node(N) to node (D)). However, the result does not satisfy this aim, because the route discovery cost is up to double the expected cost without using cached route. New route discovery cost will be added instead of reducing quarter of existing route discovery process because of using stale cache entries. In this case, the path length (number of passed hops) until reaching a broken link is too long compared with cached route length. In this case, without using cached route, the chance of obtaining a fresh route is high.

As declared above, using stale cached routes can degrade performance instead of improving it.

4.2. Research questions

Two research questions are discussed in this work:

Question one:

During route discovery, when do we use the cache, and when do we stop using it? In other words, what is the suitable threshold value that using cache at it improve EDSR performance?

Question two:

Which route should be cached, and which route should be removed?. In other words, what is the criterion that is used to specify which route should be cached?.

To answer these two research questions, we carried out an empirical study. The implementation of basic DSR is used to implement our contribution.

4.3. Enhanced DSR

The aim of this thesis is to improve the performance of Basic DSR protocol in terms of using route cache in route discovery mechanism. EDSR protocol is developed to provide answers to the research questions. The main objectives of this work are:

- To suggest an efficient Route Discovery mechanism that uses a threshold based technique. This technique is used to determine when to reply the cached route and when to continue broadcasting of RREQ packet. The suggested Route Discovery mechanism is based on distance metric.
- To reduce data packet lost, Control Overhead, and packets delay, that results from using stale cached routes.
- To investigate limited node's cache size, by avoiding the ability of caching more than two routes per destination (the main route and the alternative route, which could be used in case of main route broken). These two routes are selected based on some caching criteria, which will be discussed in section (4.4.2.a).
- To make caches as fresh as possible. Caching just fresh routes that came from full RREQ broadcasting instead of caching any route that might be cached in previous time.

4.4. Design of EDSR

Since the proposed EDSR is extended and developed from Basic DSR, the basic operations are the same as in DSR with some modifications of using cache in Route Discovery mechanism.

The proposed EDSR is composed, generally, from two modules: EDSR Route Discovery module, and EDSR caching module (as shown in figure 4.2). The first module (route discovery) accepts three inputs: source node, desired destination node, and data packet that the source node attempts to send it to the destination node. The unique output of this module is the route between source and destination. This output acts as input for the second module (caching module). The caching module will decide to caching this route or not, depending on specific caching criteria that will be discussed in section (4.4.2.a).

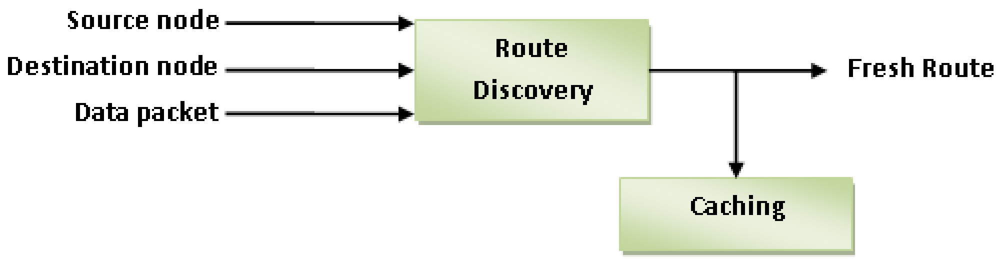


Figure 4.2. General module for EDSR

4.4.1. Module One: EDSR Route Discovery Mechanism

As in Basic DSR, The process of EDSR Route Discovery is fully on demand. It has the same operation of Basic DSR, except the term of replying from intermediate route cache.

In EDSR, route is discovered based on the source route length that RREQ packet passed and the length of cached route. In other words, when RREQ packet is far away from source node, and it become near destination, we prefer to continue broadcasting RREQ instead of reply the intermediate cached route (if exist). Figure (4.3) represents data flow diagram (DFD) for route discovery process.

Before reply from route cache, our system ensures that the fraction of remaining distance to the destination node to the passed distance from the source node is greater than the empirical threshold value (see equation 4.1).

$$\frac{\text{distance from intermediate node to destination}}{\text{distance from source node to intermediate node}} > threshold$$

………(*Equation 4.1)*

An empirical study is carried out to determine the suitable threshold value that will improve the network performance. We will select an empirical threshold value depending on the network diameter (as will be discussed in chapter six).

A) Design of Packet Structure for EDSR Route Discovery Process:

The Route Request option header included in EDSR header option format is the same as in Basic DSR. For more details see appendix (A). It is used without any modifications. While the Route Reply option header included in EDSR header option format is modified. New flag (cached flag) is added to the Basic DSR, (See table(4.1)). This flag is set to one (if any part of route that reply to source node are taken from cache). Adding this flag may avoid caching any route that was cached in previous time in another node. Since this route may be stale quickly

Table 4. 1. RREP header format

Option Type	Opt Data Len	Last Hop External (L)	Reserved	***Cached flag***
Address[1]				
Address[2]				
. . .				
Address[n]				

B) Processing EDSR Route Request

As in Basic DSR, when source node wishing to send data to a desired destination node, it searches initially in its cache on route(s) to the desired destination. If route exist, it uses it to transfer data packets. Otherwise, it broadcast RREQ packet to all neighbors. If any of these neighbors is a destination, it returns a RREP packet to source node. Otherwise, if any neighbor has route to destination in its cache, EDSR has different technique from that in DSR.

EDSR uses threshold based technique before reply the cached route from intermediate node. In threshold based technique, node will calculate the length (hop count) of the coming RREQ's route record and the length of that found in intermediate node's cache. The fraction of cached route hop count and RREQ's route record hop count must be greater than threshold value to return RREP from intermediate node's cache. Otherwise, discard cached route and continue of broadcasting RREQ in the same way in Basic DSR until reach the desired destination (see figures (4.3) and (4.4)) .

C) Processing EDSR Route Reply

The Route Reply packet (RREP) is processed by three types of nodes; source node, destination node and intermediate nodes of the request. The operation that performed in both source and destination is the same as in Basic DSR. The modification is performed in case of intermediate node. In EDSR, when intermediate node has cached route to destination, it calculates the fraction between the distance to destination node and the distance from source node. If the fraction is greater than threshold value, then, set cached flag which in RREP header option to one, and then, send RREP to source node (see figure (4.3) and (4.4)).

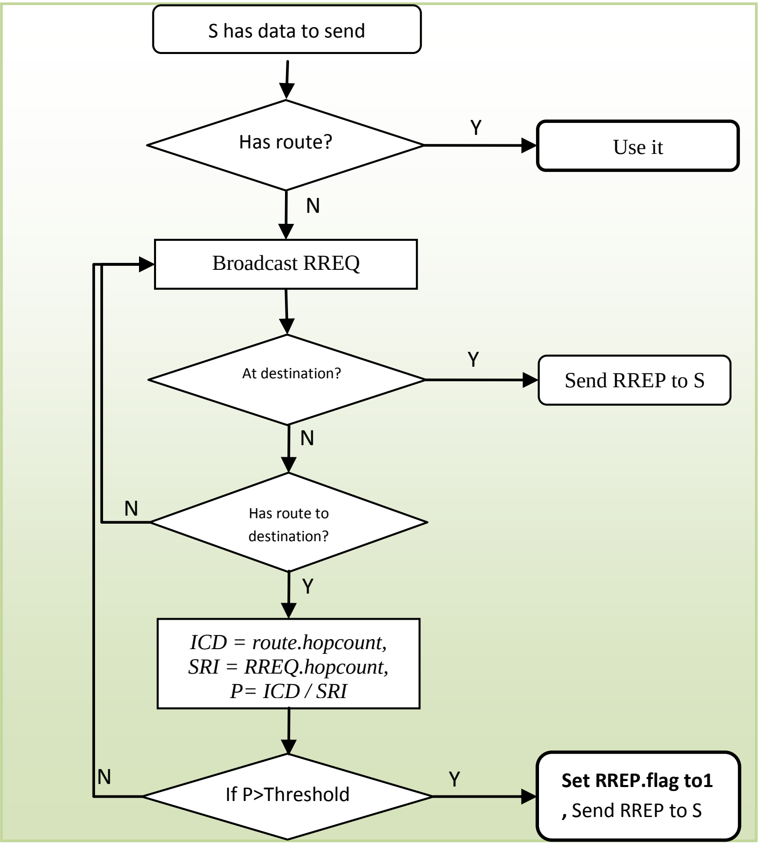


Figure 4.3. Module one (EDSR Route Discovery (RREQ & RREP))

```
1. Let ICD be the route taken from intermediate node cache to destination.
2. Let SRI be the route from source node to intermediate node that has cached route to destination.
3. Calculate P, where $P = {ICD.HopCount}/{SRI.HopCount}$
4. If $P < threshold$, then discard the route from intermediate cache (node must add itself to source route record of RREQ and continue of RREQ propagation)
```

Figure 4.4 . Pseudo-code for route discovery process

4.4.2. Module Two: Design of EDSR Caching Module

In this module, we answer the question of "which route should be cached and which route should be removed?"

The input of this module is the route discovered in route discovery mechanism. The output of this module may be two routes, as maximum, or zero, as minimum (see figure (4.5)). When the node hear or receive a route, it does not directly cached this route. It ensure that this route satisfies some caching criteria (freshness and hop count). If the node's route cache contains less than two routes per destination and if the route satisfies freshness criteria, then node will cache this route. Otherwise, if cache contains two routes for specific destination, it will performs a comparison between the three routes based on hop count criteria (route length). The two least routes lengths will be added to cache, third route must be discarded or dropped from cache if it has already cached.

a. Caching Criteria

We cached only two routes for specific destination. These two routes is selected based on two caching criteria:

1. **Freshness**: this criterion is achieved by selecting the route that is generated using complete broadcasting of route request. The flag, that was added to RREP header option,

is used to check freshness criterion. If the route is picked (or part of it) from node cache, the flag will be set to one. In case of setting this flag, the route, that contained in RREP packet, is forbidden from caching in another node cache. Otherwise, if flag is set to zero, the route is allowed to be cached in any other node's cache.

2. **Hop count:** this criterion represents number of hops of each route. It is equal to number of nodes in the route minus one. The least hop count, the best route.

When node receive or hear a route, it should ensure that this route satisfies freshness criterion at first, then the comparison is done based on hop count criterion. Least two routes for specific destination is selected to be cached.

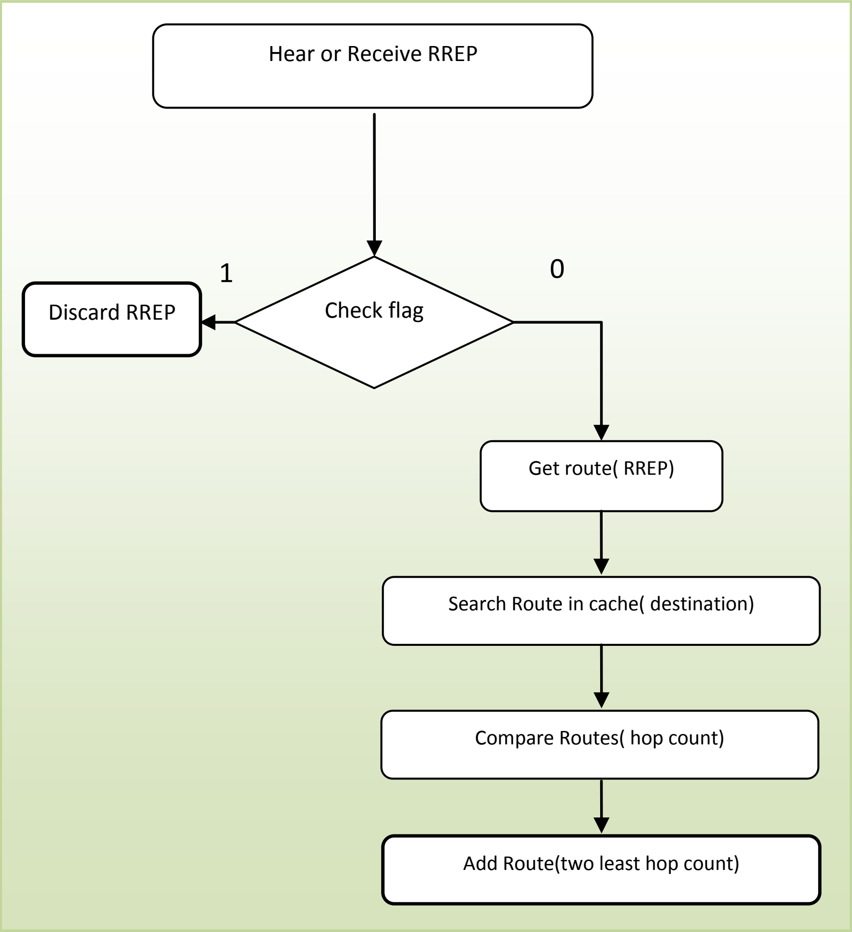


Figure 4.5. Module two (EDSR caching management)

Chapter 5

Simulation

Simulation methodology is used to evaluate Enhanced DSR (EDSR) protocol (suggested in this work). The suggested protocol experimented with 900 seconds of simulation time over a rectangular area. Rectangular area is chosen to ensure high probability of long routes between any two nodes (D. Jörg, 2003). We experiment this simulation with 900 second of simulation time (15 minutes) over (1500×300 m^2) of terrain area.

EDSR simulation is performed throughout three different scenarios. Each of them has three states that differ in number of CBR (Constant Bit Rate) sources, packet transmission rate (which repeated for each number of CBR sources), and the level of mobility. All scenarios have equal values of terrain area size, number of mobile nodes, and radio transmission range:1500×300m^2, 50 nodes, 250m respectively. All mobile nodes were randomly distributed within terrain area.

5.1. Simulation Environment

The simulation results evaluation is implemented using a simulator of GLOMOSIM-2.03 (Global Mobile Information Systems Simulation) (Zeng et al, 1998). GLOMOSIM is a scalable simulator for wired and wireless network systems. It is used for parallel discrete-event simulation capabilities. It is a C-based compiler that designed as a layered approach. Application, transport, network, MAC, and physical (radio propagation) layers are defined in GLOMOSIM simulator.

In this simulation, we used CBR, UDP, DSR, 802.11, and TWO-RAY as GLOMOSIM models for each layer respectively. We compared the performance of the suggested EDSR with the Basic DSR that was implemented by Johnson et al in 1996.

5.2. Simulation Tools and Parameters

GLOMOSIM simulator has several input files and one output file. In this simulation, we use two input files and one output file: CONFIG.IN, APP.CONFIG, and GLOMO.STATE. At the start of simulation, the simulator read parameters from input files. During simulation, simulator saves simulation information in a specific data structure. At the end of simulation, it collect information from the data structure, and view it in the output file (GLOMO.STATE). Major parameters are defined in the input file (CONFIG.in). We specify the number of sources, destinations, and the number of data packets to send to destinations in input file (APP.CONFIG). Table (5.1) shows the general parameters that we used in our simulation:

Table 5. 1. Simulation parameters

Parameter	Value
1. SIMULATION TIME	15M
2. NUMBER OF NODES	50 NODES
3. SIMULATION AREA	(1500×300) m^2
4. NODE-PLACEMENT	RANDOM
5. MOBILITY MODEL	RANDOM WAY-POINT (RWP)
6. SPEED UNIFORM	(0, 20) m/s
7. PAUSE TIME	0s, 300s, 600s, 900s
8. RADIO PROPAGATION	TWO-RAY
9. TRANSMISSION RANGE	250 m
10. CHANNEL CAPACITY	2000000 bit/s
11. MAC PROTOCOL	IEEE 802.11
12. PROMISCUOUS-MODE	YES
13. ROUTING PROTOCOL	DSR
14. NETWORK DIAMETER	9

The selected mobility model, which describes the movement behavior of mobile nodes, is the RWP model. For RWP, a node randomly selects a destination point in the area, and then it moves in the direction of this destination in a speed uniformly chosen between the minimum mobility speed and the maximum mobility speed. When the node reaches its destination, it stays there for a fixed pause time, then it chooses a new destination point, and so on (Bettstetter et al, 2003). We choose the minimum mobility speed equal to zero m/s, and the maximum mobility speed equal 20 m/s. Several mobility pause times are chosen (0s, 300s, 600s, 900s).

Traffics between source and destination are generated using CBR model. The standard CBR traffic format is as follows:

CBR <src> <dest> <items to send> <item size> <interval> <start time> <end time>

Where:

<src> represents the source node.

<dest> represents the destination node.

<items to send> represents how many items to send.

<item size> represents the size of each item.

<interval> is the interdeparture time between the items.

<start time> represents when to start CBR during the simulation.

<end time> represents when to terminate CBR during the simulation.

To understand CBR model, consider the following command line examples:

a) CBR 1 2 20 1500 1S 0S 600S

This line means that Node 1 sends to Node 2 twenty items of 1500B each at the start of the simulation up to 600 seconds into the simulation. The interdeparture time for each item is 1 second (one packet per second). If the twenty items are sent before 600 seconds elapsed, no other items are sent.

b) CBR 1 2 0 1500 1S 0S 600S

0 in <item to send> means that number of packets in infinity. This line means that Node 1 continuously sends to Node 2 items of 1500B, each at the start of the simulation up to 600 seconds into the simulation. The interdeparture time for each item is 1 second.

c) CBR 1 2 0 1500 1S 0S 0S

This line means that node 1 continuously sends to node 2 items of 1500B each at the start of the simulation up to the end of the simulation. The interdeparture time for each item is 1 second.

d) CBR 1 2 100 1500 0.5S 0S 0S

This line means that node 1 continuously sends to node 2 items of 1500B each at the start of the simulation up to the end of the simulation. The interdeparture time for each item is 0.5 second (two packets per second).

In APP.CONFIG file, We specified three different number of CBR sources (5,10, and 15) with fixed payload size (512 bytes), at a transmission rate of (1,2, and 4) packets per second respectively as considered in (D. Jörg, 2003), transmitted from start to end of the simulation time.

5.3. Evaluation Metrics

Several evaluation metrics are used to evaluate network performance, when using specific protocol in comparison of another one. We used four metrics to evaluate the suggested EDSR protocol in comparison with Basic DSR. The metrics that we used are Packet Delivery Ratio, Control Overhead, Average End-to-End delay, and Throughput.

- ***Packet Delivery Ratio(PDR):***

  PDR represents the fraction of total number of data packets successfully received by destination node(s) divided by total number of data packets sent by source node(s) (see equation (5.1)). This metric can describe the network throughput and lose rate (Conrad, 2003)(Adane, 2003).

$$\mathbf{PDR} = \frac{\sum packets\ successfully\ received\ by\ destination\ node\,(s)}{\sum packets\ sent\ by\ source\ node\,(s)} \ldots\ldots\ldots\ldots\ldots \mathit{(equation\ (5.1))}$$

  For instance, if the total number of data packets sent by source node(s) is (2000) packets, and the total number of data packets successfully delivered by destination nodes is (1500) packets, then PDR equal to (75%). This means that 3/4 of all data packets sent by source are successfully delivered by destination. The aim is to increase PDR to improve network performance. The higher PDR, the better network performance.

- ***Control Overhead***

  Control Overhead is the percentage between the total number of control packets sent to the total number of data packet received by the destinations (Hassan, 2008).

$$\textbf{\textit{Control overhead}} = \frac{\sum control\ packets\ sent}{\sum data\ packet\ received\ by\ the\ destinations} \ldots\ldots\ldots\ldots \mathit{(equation(5.2))}$$

  For instance, if the total number of control packets sent within simulation time equals to (1000) control packets. If the total number of data packets received by destination within the same simulation time equal to (500) data packets, then Control Overhead equal to two. This means that we need two control packets to send one data packets. The aim is to keep overhead as small as possible to improve network performance. The lower control overhead the better network performance.

- ***Average End-to-End delay:***

Average End-to-end-delay is the average time taken to send data packet from source node until received by destination node. This time includes processing time, propagation time, MAC transmission time, etc. End-to-end delay for each data packet is the difference between start time and end time for sending this data packet. Average End-to-End delay is the division between total end-to-end duration and the total number of data packets received (Gani et al, 2009). The aim is to decrease average End-to-End delay in order to improve network performance. The lower Average End-to-End delay, the better network performance.

- ***Throughput:***

Network Throughput is the average rate of successful data packets delivered to destination. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second (*Gani et al,2011).*

5.4. Scenarios Design

A number of different scenarios with different simulation parameters are experienced. We have designed three different simulation scenarios to evaluate the EDSR work. Each scenario is configured with fifty mobile nodes placed over a 1500×300m^2 of terrain area with 250m of radio transmission range.

a) Scenario one:

Scenario one represents the simulation of five CBR sources. Each of them sends (10000) application items (packets) of 512B each at the start of the simulation up to the end of the simulation. The interdeparture time for each item may be one of three cases for each of the three scenarios; one second, half second, and quarter second. Each case is repeated for four different mobility pause times: 0S, 300S, 600S and 900S (simulation time). The following are the three cases of scenario one:

Scenario one- case one: the interdeparture time for each item is one second. This means that the transmission rate is one packet per second.

Scenario one- case two: the interdeparture time for each item is half second. This means that the transmission rate is two packets per second.

Scenario one- case three: the interdeparture time for each item is quarter second. This means that the transmission rate is four packets per second.

b) Scenario two:

Scenario two represents the simulation of ten CBR sources, each of them sends (10000) packets of (512B) each at the start of the simulation up to the end of the simulation. The interdeparture time for each item is also one of three cases mentioned in scenario one each of them repeated for the four mobility levels. The following are the three cases of scenario two:

Scenario two- case one: the interdeparture time for each item is one second. This means that the transmission rate is one packet per second.

Scenario two- case two: the interdeparture time for each item is half second. This means that the transmission rate is two packets per second.

Scenario two- case three: the interdeparture time for each item is quarter second. This means that the transmission rate is four packets per second.

c) Scenario three:

Scenario three represents the simulation of fifteen CBR sources each of them also sends (10000) packets of (512B) each at the start of the simulation up to the end of the simulation. The interdeparture time for each item is also one of the three cases mentioned in scenario one, each of them repeated for the four mobility levels. The following are the three cases of scenario three:

Scenario three- case one: the interdeparture time for each item is one second. This means that the transmission rate is one packet per second.

Scenario three- case two: the interdeparture time for each item is half second. This means that the transmission rate is two packets per second.

Scenario three- case three: the interdeparture time for each item is quarter second. This means that the transmission rate is four packets per second.

At the end of simulation scenario design, we gained three scenarios with nine different cases for evaluating the Enhancement DSR protocol in comparison with the basic DSR.

Chapter six

Results Assessment

This chapter discusses the simulation results of the proposed EDSR protocol compared with the results of Basic DSR. The evaluation and comparison is performed according to four metrics mentioned in chapter five (PDR, Control Overhead, Average End-to-End delay, and Throughput). Three scenarios are considered, three different states are simulated for each of the three different scenarios (defined in chapter five). Each data value in the figures below is achieved by run the simulation ten times, and then get the average of these ten results. This gives relative errors that don’t exceed 5% with 95% confidence. We have calculated the percent improvement by division the difference between original value and new value on the original value.

6.1. Threshold Selection

In this section, we described how to choose the suitable threshold value, which used in EDSR route discovery. We concerned to select the threshold value that satisfies equation (4.1) to achieve our thesis objectives.

In this simulation, we specified the value of the network diameter equal to nine as maximum diameter value that used in DSR, (see Table 5.1). Nine candidate values are selected to choose threshold value from. The nine candidate values are: (0, 1, 2, 3, 4, 5, 6, 7, 8). The selection of these candidate values are done depending on the network diameter. For instance, if the route length equals to nine nodes (maximum route length), the pairs of (distance from source to intermediate, distance from intermediate to destination) may be one of the permutation [(0,9), (1,8), (2,7), (3,6), (4,5), (9,0), (8,1), (7,2), (6,3), (5,4)]. Many experiments were performed on these nine values. We evaluated EDSR using these candidate values throughout two simulation scenarios. The first one is, five CBR sources each sends one packet per second (scenario one/ case one). The second one is, fifteen CBR sources each sends four packets per second (scenario three/case three). These two scenarios represent the lower and upper part of simulation scenarios. We also compare the performance results of EDSR with each nine candidate threshold values depending on two metrics; Packet Delivery Ratio and Control Overhead. We named the results for EDSR of each threshold values by $\boldsymbol{enh}_j$, where $_j$ represent the candidate threshold value. To determine threshold value, we carried out the following experiments.

(Figure 6.1) show the results of PDR of EDSR with different threshold values simulating throughout five CBR sources each sends one packet per second in different mobility levels. From this figure, EDSR outcomes Basic DSR in all mobility levels and with all threshold values except threshold value (8). We note that threshold values (**2,3, and 4**) has the best results of PDR and they have convergent results.

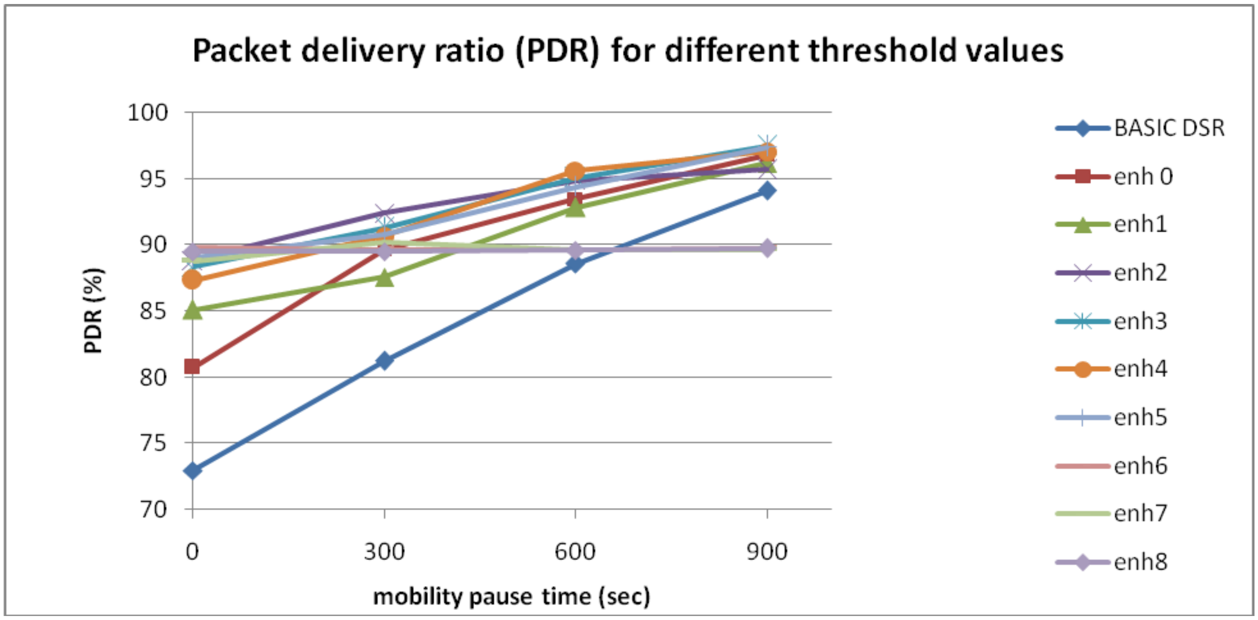


Figure 6.3. PDR of 5 sources each sends 1PPS

The relationship between Control Overhead and mobility level for five CBR sources each sends one packet per second that simulating for EDSR with different threshold values are shown in Figure (6.2). This figure shows that Basic DSR outcomes EDSR protocol for all threshold values at high mobility. However, in the other mobility levels both of DSR and EDSR have convergent results of Control Overhead except in threshold value (4) that have the worst results in this scenario. In this case we can't get any evidence to determine threshold value since Basic DSR actually outcomes EDSR with all threshold values.

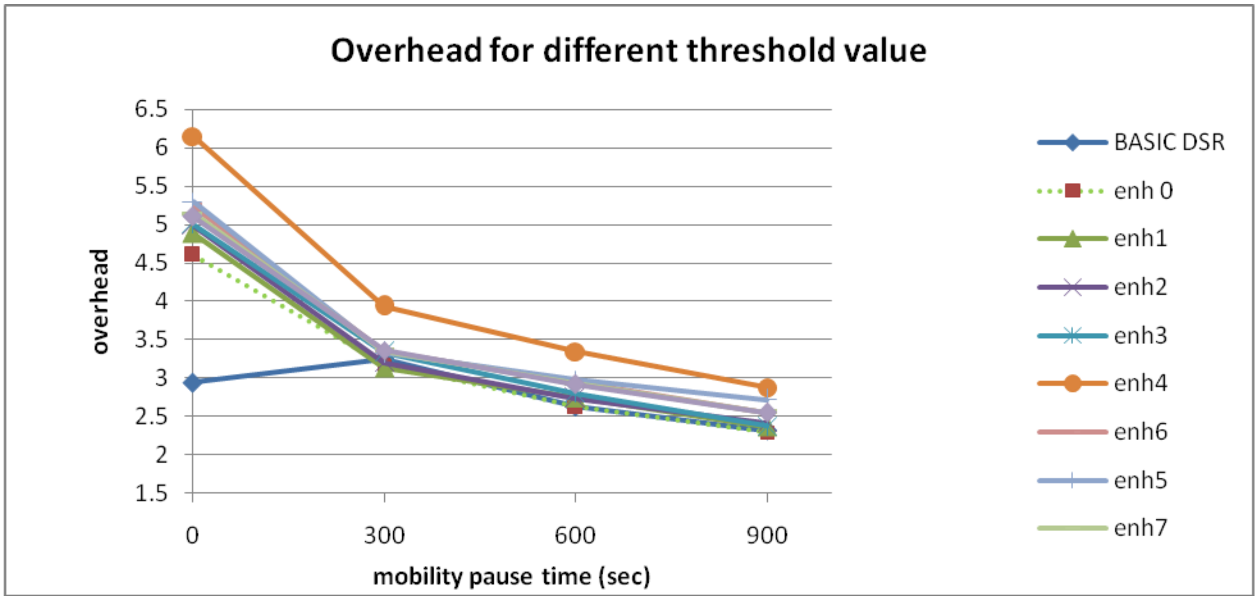


Figure 6.4. Overhead of 5 sources each sends 1PPS

The relationship between PDR and mobility level for fifteen CBR sources each sends four packets per second that simulating for EDSR with different threshold values are shown in Figure (6.3). In this figure, it is obvious that EDSR with threshold value **(4)** outcomes DSR and EDSR with all other threshold values in the most of mobility levels and it gives the best results of PDR than others.

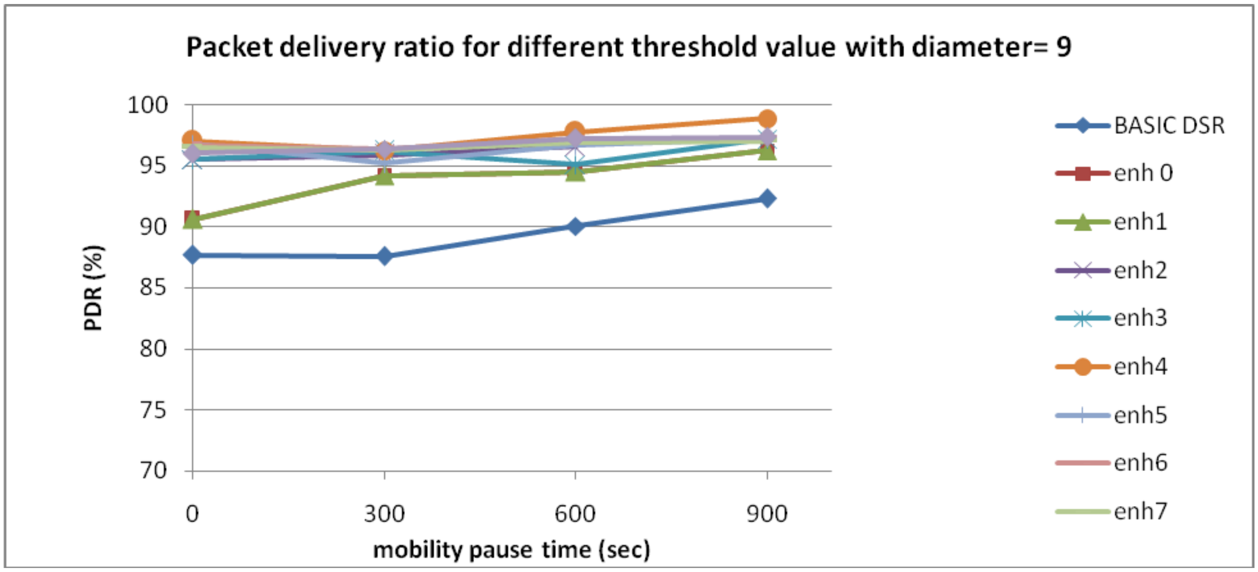


Figure 6. 5. PDR of 15 sources each sends 4PPS

The relationship between Control Overhead and mobility level for fifteen CBR sources each sends four packets per second that simulating for EDSR with different threshold values are shown in Figure (6.4). In this figure, we note that EDSR outperforms Basic DSR in at most all threshold values. However, threshold values **(4, 7, and 8)** give the best Control Overhead results convergent results with least values of overhead than other threshold values.

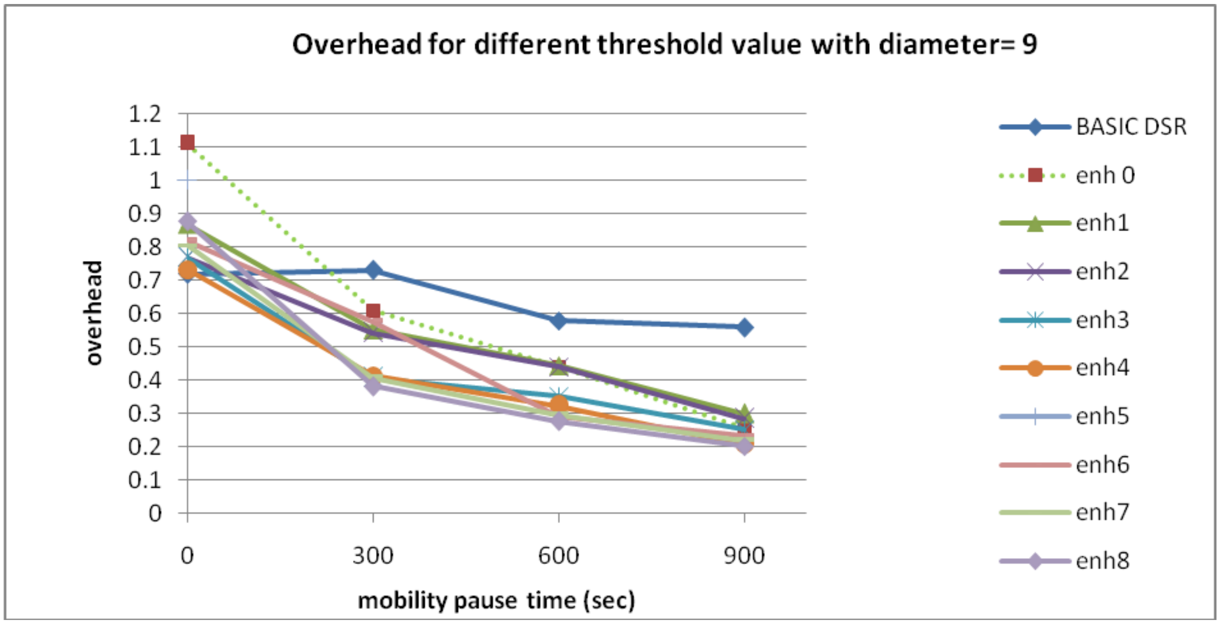


Figure 6.6. Control Overhead of 15 sources each sends 4PPS

Based on the results of PDR and Control Overhead metrics that shown in Figure (6.1), Figure (6.3), and Figure (6.4), we can concludes that the value (4) can be chosen to be the threshold value. At overall, we have chosen the following threshold value since it has given good results than other threshold values:

Threshold value = $\lfloor$ ***½ network diameter*** $\rfloor$

When network diameter= 9 → suitable threshold value= 4 (see Table (6.1)).

By testing this value in Equation (4.1), We can deduced that after one hop away from source node we stop replying from cache. This means that each source node accept a cached route just from its neighbor.

6.2. Evaluation Results: Packet Delivery Ratio

In this section we consider the relationship between mobility levels and Packet Delivery Ratio metric of both Basic DSR and the EDSR protocols for the nine cases illustrated in chapter five. Result analysis of PDR for each of the nine cases will show if there is any improvement in the EDSR over Basic DSR. The results of PDR for the nine cases are illustrated below:

- ***Scenario One- Case One:*** Five CBR sources are chosen with transmission rate of one packet per second. Figure (6.5) shows that EDSR outperforms the Basic DSR in all mobility levels. In this case, when node mobility is high (pause time= 0s), both of the two protocols show low delivery ratio, but EDSR gained about 20% ((87-72) / 72) of delivery ratio over Basic DSR. At pause time=300s, the performance is improved for both protocols. The delivery ratio of EDSR outperformed Basic DSR by about 12%. When pause time=600s, the delivery ratio also improved; EDSR gained about 8% of delivery ratio over Basic DSR. At pause time= 900s, EDSR outperforms The delivery ratio of Basic DSR by 3%. We get the best performance of both protocols when mobility is low. This reason because the node change its position once at the simulation time, therefore, route failure reduced.

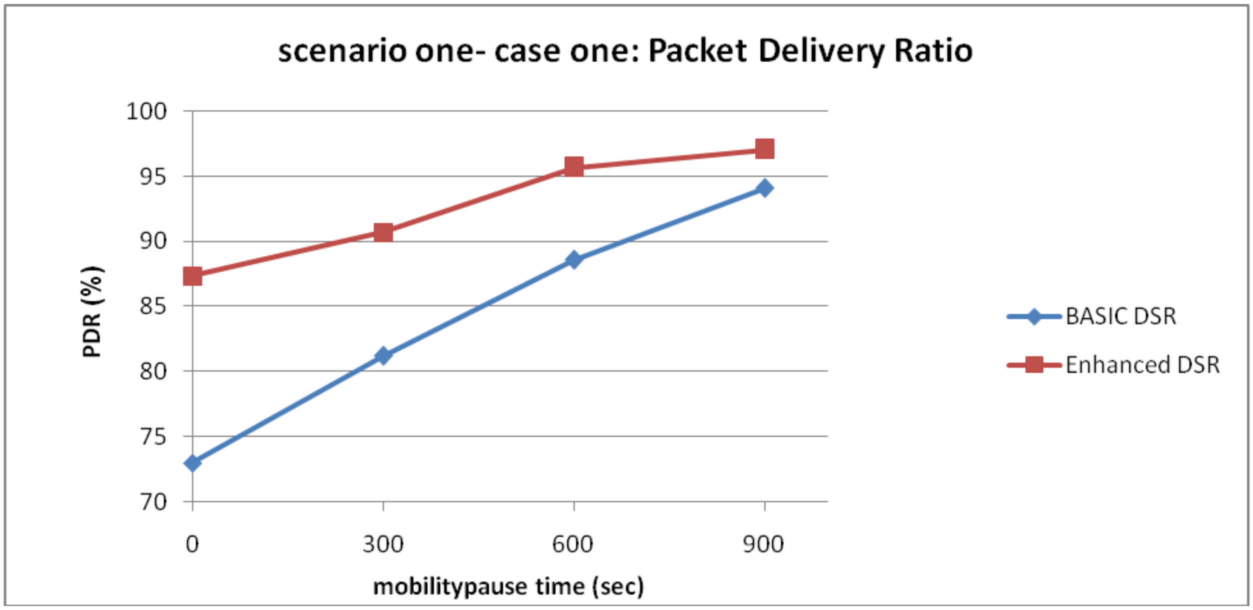


Figure 6. 5. PDR of 5 sources each sends 1PPS

Generally, in scenario one- case one, the higher mobility level, the higher improvement of EDSR over Basic DSR. This reason since as mobility increases, more routes will become stale; the advantage of our thesis become more significant.

- ***Scenario One- Case Two:*** Five CBR sources are chosen with transmission rate of two packets per second. Figure (6.6) shows that the delivery ratio of EDSR outperforms Basic DSR in all mobility levels. In this case PDR of both of the two protocols increase as pause time increase. When mobility is high, EDSR outperforms Basic DSR by about 14% of delivery ratio. At pause time=300s, EDSR outperforms Basic DSR by about 9% of delivery ratio. When the pause time=600s, EDSR overtakes Basic DSR by about 6% of delivery ratio**.** At low mobility, EDSR outperforms DSR by about 2% of delivery ratio. In general, we can also deduce that, the higher mobility level, the higher improvements of EDSR over Basic DSR.

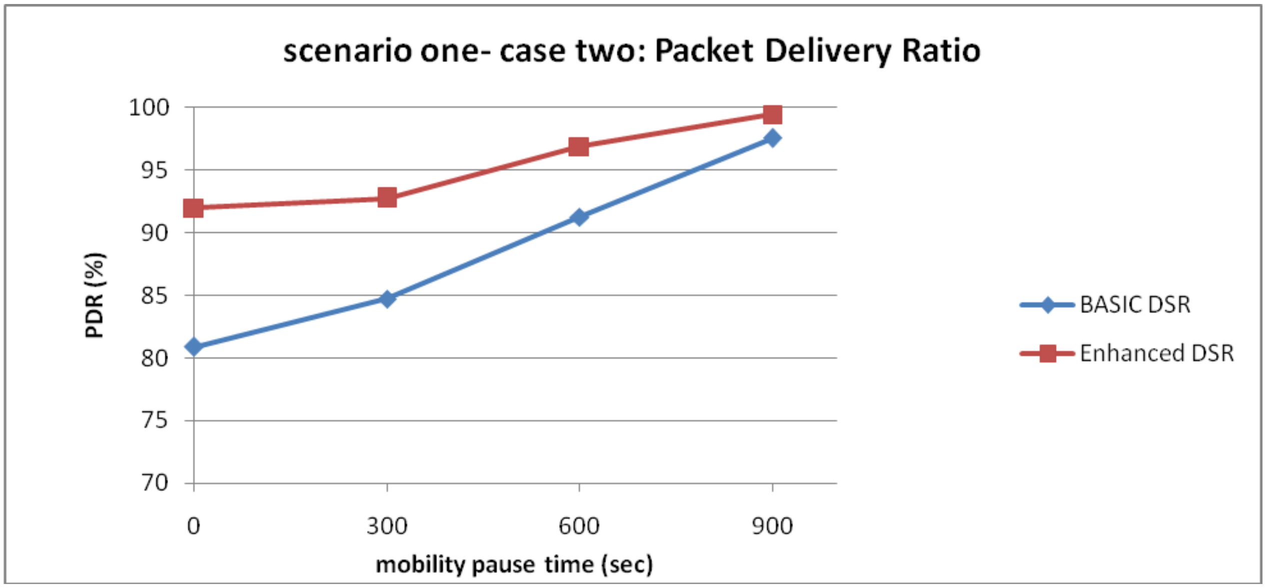


Figure 6.6. PDR of 5 sources each sends 2PPS

- ***Scenario One- Case Three***: Figure (6.7) illustrates the relationship between PDR and mobility levels of scenario one- case three, where five CBR sources are chosen with transmission rate of four packets per second. From the figure, it is clearly seen that EDSR outperforms Basic DSR in all mobility levels. Both two protocols raise their PDR along all mobility levels. When mobility is high, EDSR overtakes Basic DSR by about 9% of delivery ratio. At pause time=300s, EDSR outperforms Basic DSR by about 9% of delivery ratio too. The improvement of delivery ratio of EDSR when pause time= 600s is about 5%. When the mobility is low, EDSR outperforms Basic DSR by about 2%. In this case also, it is obvious that the higher mobility level the higher improvement of delivery ratio of EDSR. As pause time increases, the network topology becomes stable, and the trend of packet delivery ratio is increasing. The spots of EDSR and DSR are almost superposition at 900s, which means that network topology trends to a stable situation, so the performance of EDSR and DSR become almost similar.

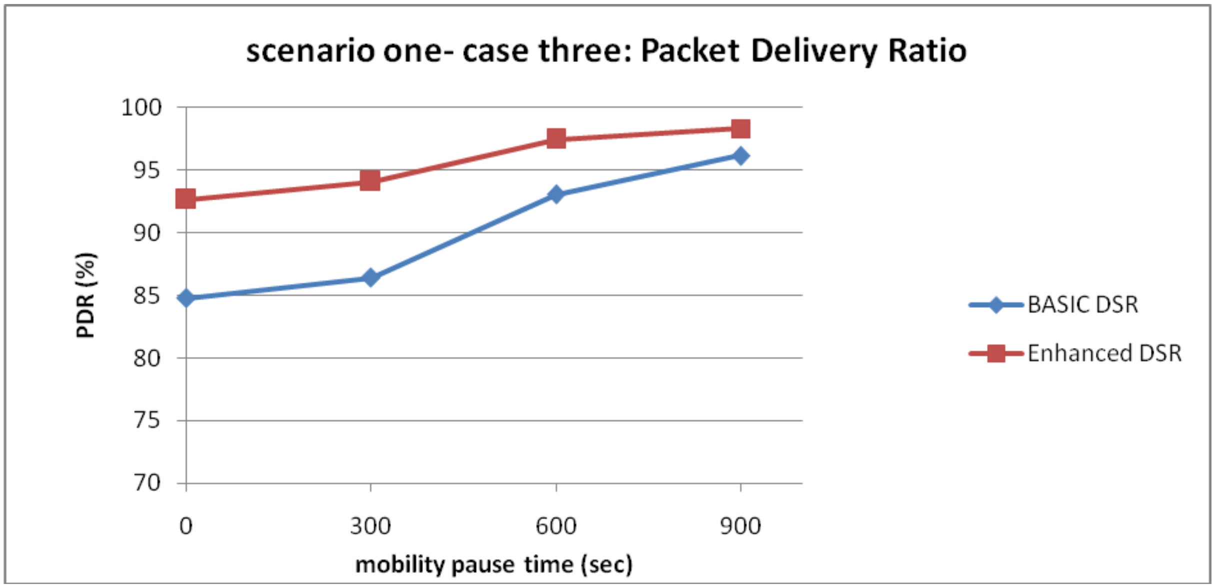


Figure 6.7. PDR of 5 sources each sends 4PPS

- ***Scenario Two- Case One:*** Figure (6.8) shows the relationship between PDR and mobility pause time of scenario two- case one, where ten CBR sources are chosen with transmission rate of one packet per second. Both of the two protocols are ascendant along all mobility pause time. EDSR also outperforms Basic DSR for all mobility levels. When mobility is high, EDSR outperforms Basic DSR by about 20%. At pause time= 300s, EDSR overtakes Basic DSR by about 10%. When pause time=600s, EDSR outperforms Basic DSR by about 7%. With low mobility, EDSR outperforms Basic DSR by about

4%. In this case, we can deduce also that, the higher the mobility levels, the higher improvement of EDSR over Basic DSR.

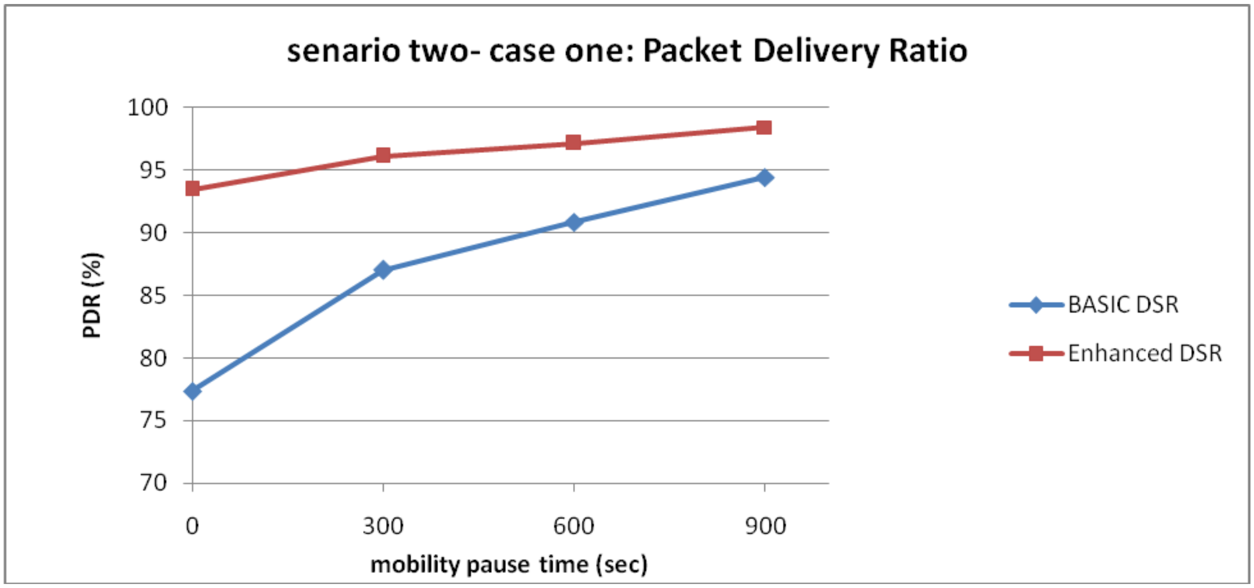


Figure 6.8. PDR of 10 sources each sends 1PPS

- *Scenario Two- Case Two: Ten CBR sources are chosen with transmission rate of two packets per second. From Figure (6.9), it is clearly seen that, EDSR outperforms Basic DSR in all mobility levels. When mobility is high, EDSR overtakes Basic DSR by about 12%. At pause time = 300s, EDSR outperforms Basic DSR by about 6%. When pause time=600s, EDSR outperforms Basic DSR by about 6%. When mobility is low, EDSR outperforms DSR by about 3%. Generally,* it is obvious also that, the higher mobility level, the higher improvement of delivery ratio of EDSR over Basic DSR.

- ***Scenario Two- Case Three:*** Ten CBR sources are chosen with transmission rate of four packets per second. Figure (6.10) shows that EDSR outperforms Basic DSR in all mobility levels. When mobility is high, EDSR outperforms Basic DSR by about 8%. When mobility is low EDSR outperforms Basic DSR by 10%. At pause time={300S and 600S}, EDSR overtakes Basic DSR by 10% too. The performance improvement of EDSR over Basic DSR is approximately fixed along all mobility levels.

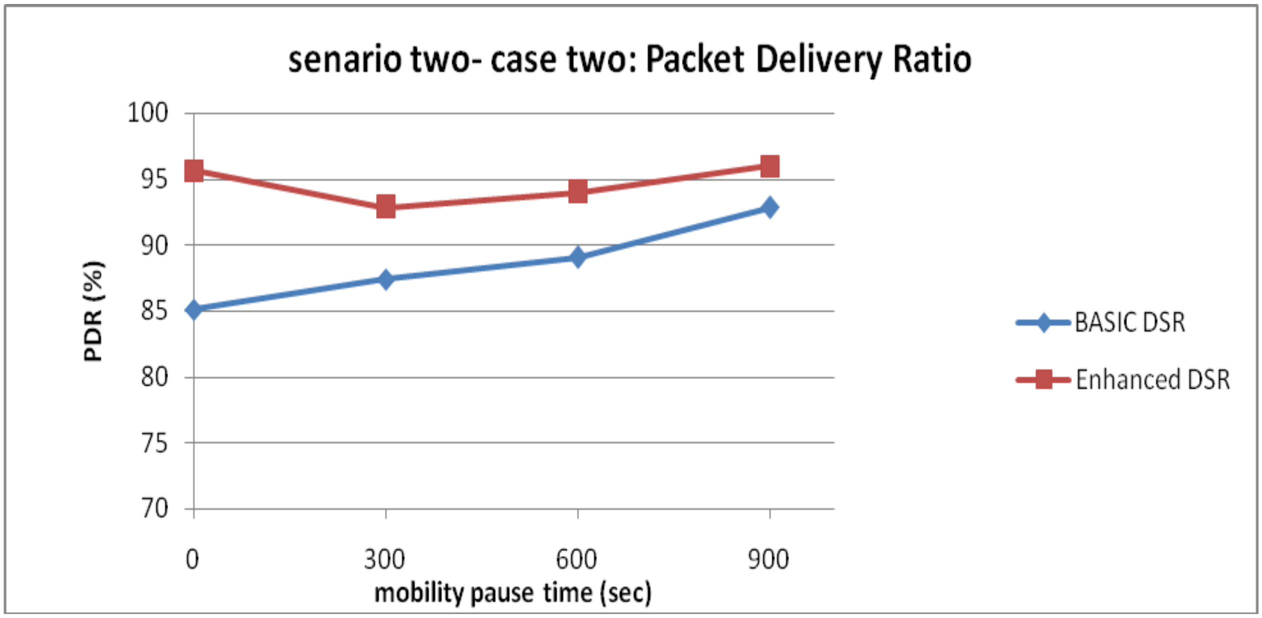


Figure 6.9. PDR of 10 sources each sends 2PPS

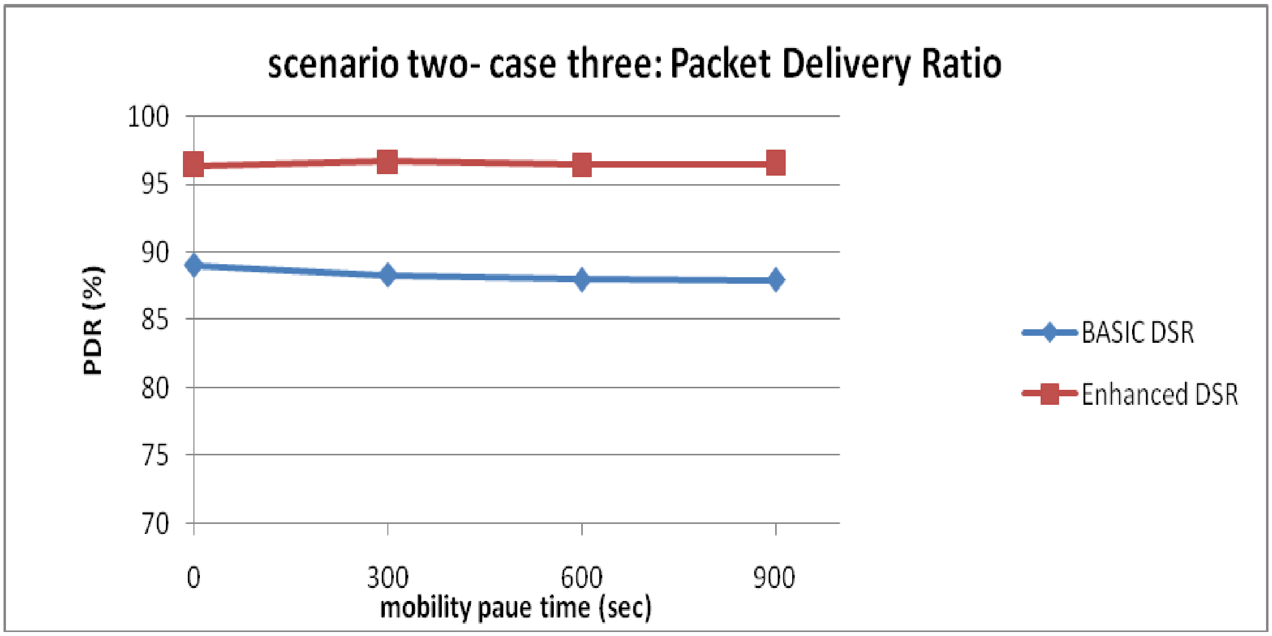


Figure 6.10. PDR of 10 sources each sends 4PPS

- ***Scenario Three- Case One*:** In this case, fifteen CBR sources are selected with transmission rate of one packet per second. The higher the mobility level, the better performance for both protocols. Figure (6.11) shows the relationship between PDR and mobility pause time. It is clearly seen that EDSR outperforms Basic DSR in all mobility pause times. When mobility is high, EDSR outperforms Basic DSR by about 22%. At pause time=300S, EDSR overtakes Basic DSR by about 12%. When pause time=600S, EDSR outperforms Basic DSR by about 8%. When mobility is low, EDSR outperforms Basic DSR by about 7%. The delivery ratio of EDSR over Basic DSR highly improved as long as mobility level increases for the same reason mentioned in above scenarios discussion.

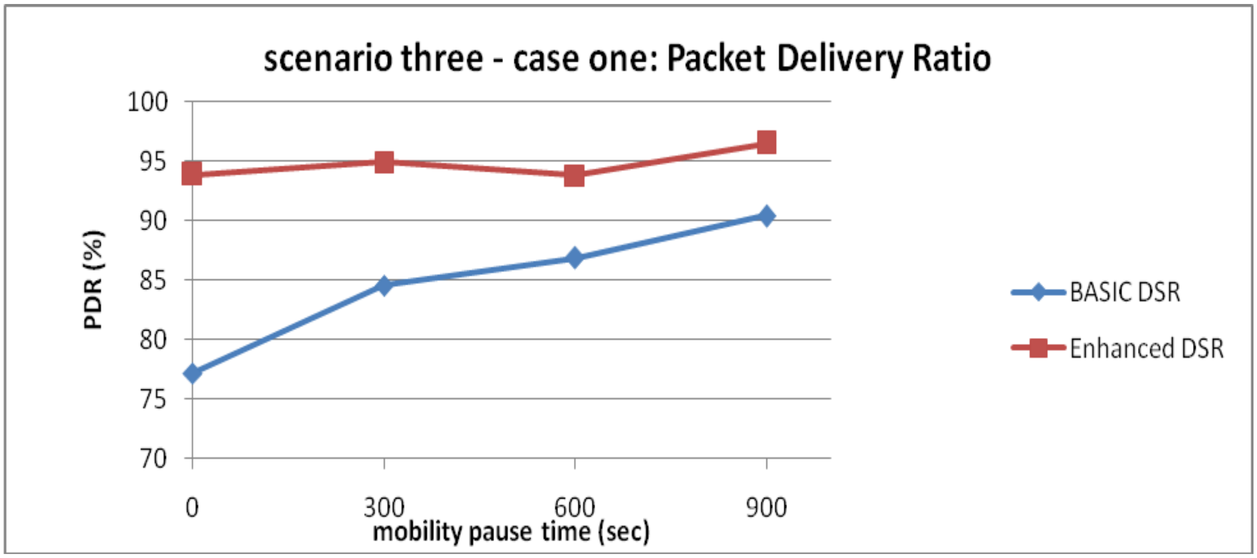


Figure 6.11. PDR of 15 sources each sends 1PPS

- ***Scenario Three- Case Two***: Fifteen CBR sources are selected with transmission rate of two packets per second. The behavior of EDSR line has fixed values along all mobility levels. EDSR outperforms Basic DSR in all mobility levels. Figure (6.12) shows that EDSR outperforms Basic DSR by about 17% at pause time=0s. When pause time=300s, EDSR outperforms Basic DSR by about 8%. At pause time=600s, EDSR outperforms basic DSR by about 6%. When mobility is low (pause time=900s), EDSR outperforms Basic DSR by about 4%. The best performance of EDSR over Basic DSR is when mobility is high also.

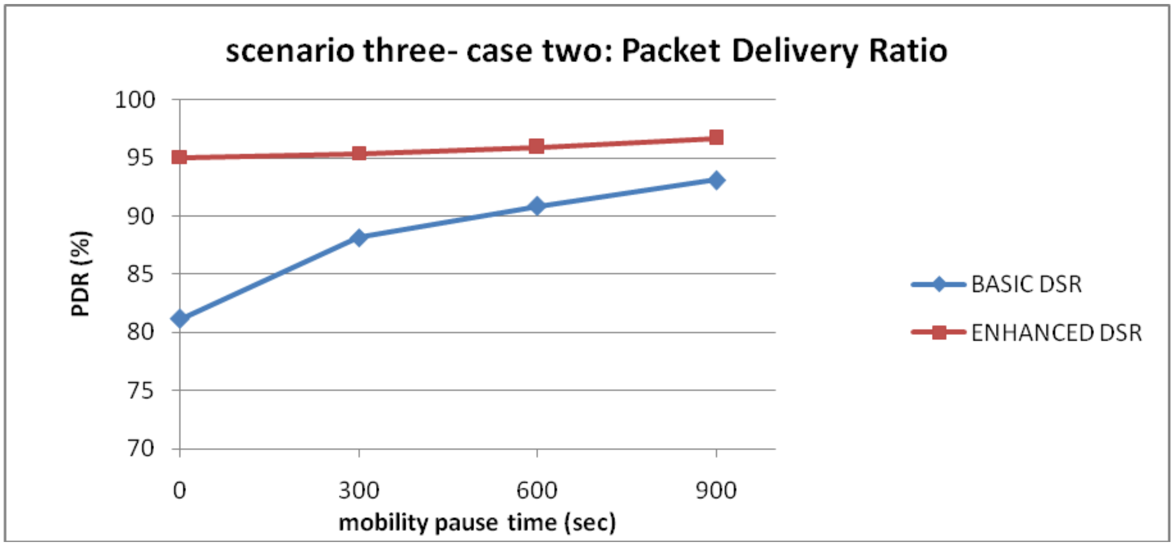


Figure 6.12. PDR of 15 sources each sends 2PPS

Scenario Three-Case Three: Figure(6.13) shows the relationship between PDR and mobility pause time of scenario three- case three. In this case, fifteen CBR sources are selected with transmission rate of four packets per second. EDSR outperforms Basic DSR in all mobility levels. For pause times=(0s, 300s, 600s, and 900s), EDSR outperforms Basic DSR by about (11%, 10%, 9%, and 7%) respectively. The performance improvement of EDSR over Basic DSR is approximately fixed along all mobility levels.

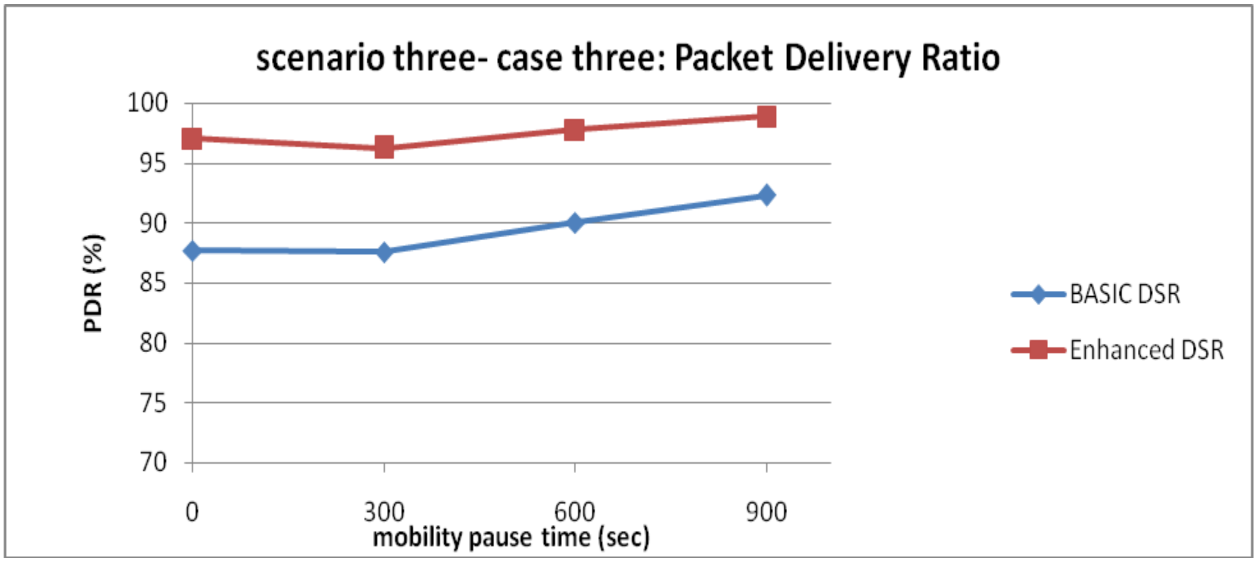


Figure 6.13. PDR of 15 sources each sends 4PPS

6.3. Evaluation Results: Control Overhead

In this section, we consider the relationship between mobility levels and Control Overhead metric of both Basic DSR and EDSR protocols for the nine cases illustrated in chapter five. The results of this relationship are analyzed to study the behavior of EDSR compared with Basic DSR in each of the nine cases. The following are the Control Overhead results of the nine cases:

- ***Scenario One- Case One***: Five CBR sources are chosen with transmission rate of one packet per second. In this case, Basic DSR outperforms EDSR in all mobility levels. EDSR shows degradation in Control Overhead metric, see Figure (6.14). Note that, Control Overhead of EDSR decreases along all mobility pause times. In this case there is no improvement of EDSR over Basic DSR. The worst performance of Control Overhead for EDSR is at high mobility. This reason due to frequent route failure because mobility, fewer cached route and fewer using of route cache.

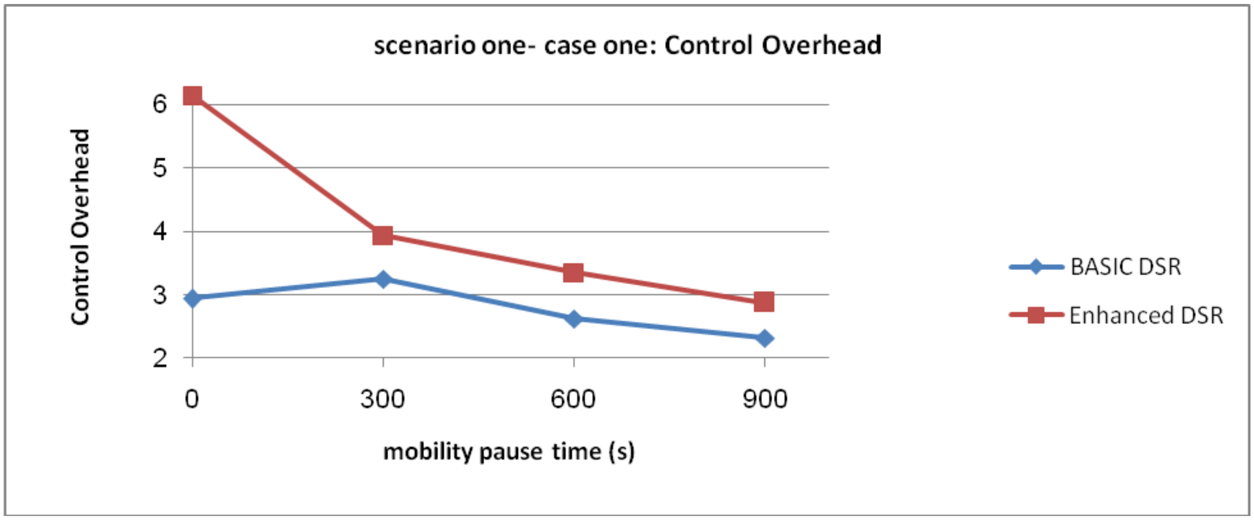


Figure 6.14. Control Overhead of 5 sources each sends 1PPS

- ***Scenario One- Case Two***: Five CBR sources are chosen each sends two packets per second. In this case, EDSR outperforms Basic DSR in two mobility levels by about (50%). While Basic DSR outperforms EDSR just in a high mobility level. This is due to frequent topology changes, which increase links failure. At low mobility, two protocols have convergent value. In this case, we can say that EDSR improves the Control Overhead of Basic DSR (see Figure 6.15).

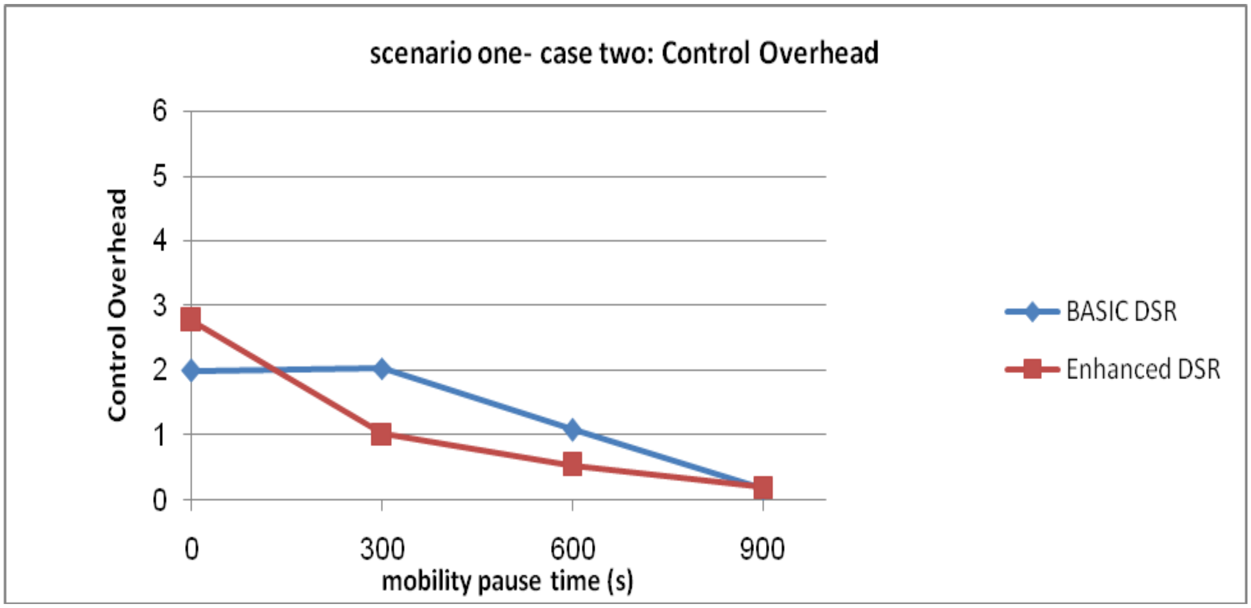


Figure 6.15. Control Overhead of 5 sources each sends 2PPS

- ***Scenario One- Case Three***: Five CBR sources are selected each sends four packets per second. As in the two cases of scenario one (scenario one- case one and case two), the Control Overhead of DSR and EDSR are decreased along all mobility pause time and the worst EDSR Control Overhead is at high mobility. In this case, both two protocols have convergent values of Control Overhead except in a high mobility level, where Basic DSR outperforms EDSR. So that we can say that EDSR does not improve the Control Overhead of Basic DSR and it does not worse than the Control Overhead of Basic DSR. They have an equal performance of Control Overhead except in a high mobility as shown in Figure (6.16).

  Generally, In scenario one, since number of CBR sources is few, and since fresher route are just able to be cached, the probability of found entries in cache may be low. In addition, EDSR has fewer cached route ,which leads to fewer packet salvaging. So that Control Overhead is increased because of increasing the broadcast of RREQ.

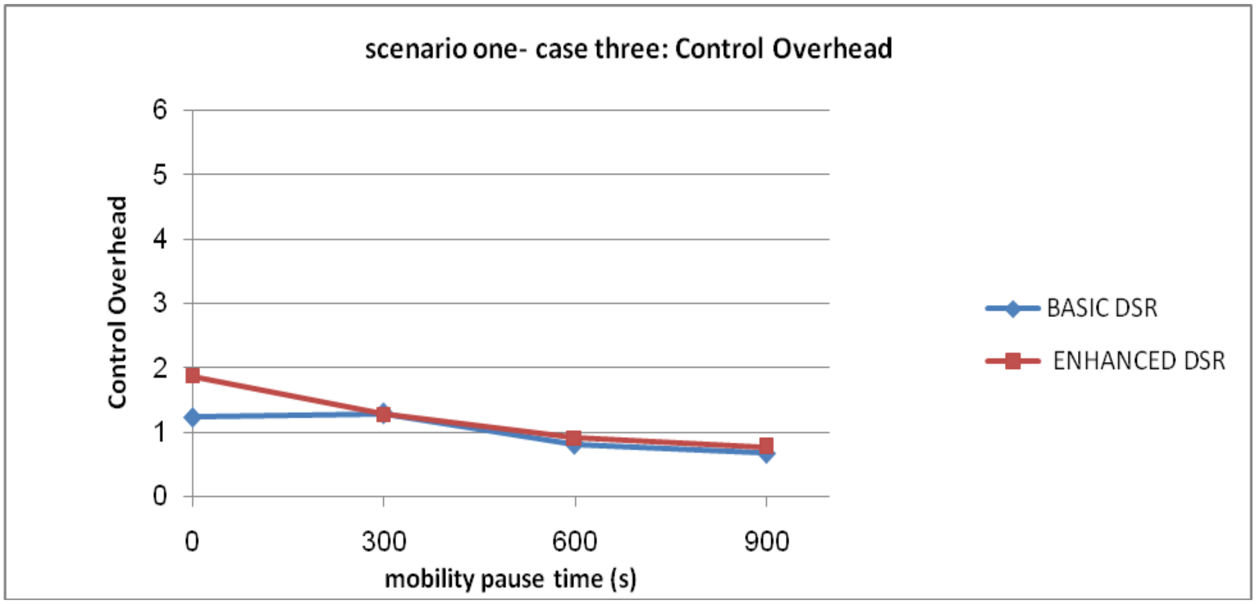


Figure 6.16. Control Overhead of 5 sources each sends 4PPS

- ***Scenario Two- Case One***: Ten CBR sources are selected with transmission rate of one packet per second. In this case, both two protocols decrease Control Overhead along all mobility pause times. They have convergent values of Control Overhead except in a high mobility (as shown in Figure (6.17)). We can say that, in this case, EDSR and Basic DSR have a convergence Control Overhead except in a high mobility as discussed above.

- ***Scenario Two- Case Two***: Ten CBR sources are chosen, each sends two packets per second. In this case, the Control Overhead of both Basic DSR and EDSR decreases for all pause time values. Basic DSR outperforms EDSR in all mobility levels but they have convergent values except in a high mobility for the same reasons discussed above, see Figure(6.18).

- ***Scenario Two- Case Three:*** Ten CBR sources are chosen with transmission rate of four packets per second. In this case, EDSR decreases its Control Overhead for all mobility levels. While Basic DSR increases its Control Overhead for all pause time values. In this case, EDSR outperforms Basic DSR in all mobility levels, except at high mobility, where Basic DSR outperforms EDSR by about (28%). The worst performance of EDSR is at high mobility for the same reasons mentioned above. When pause time= {300s, 600s, and 900s}, EDSR outperforms Basic DSR by about (63%, 77%, 85% respectively), as clearly seen in Figure (6.19). In this case, we can say that EDSR has good improvement of Control Overhead over Basic DSR.

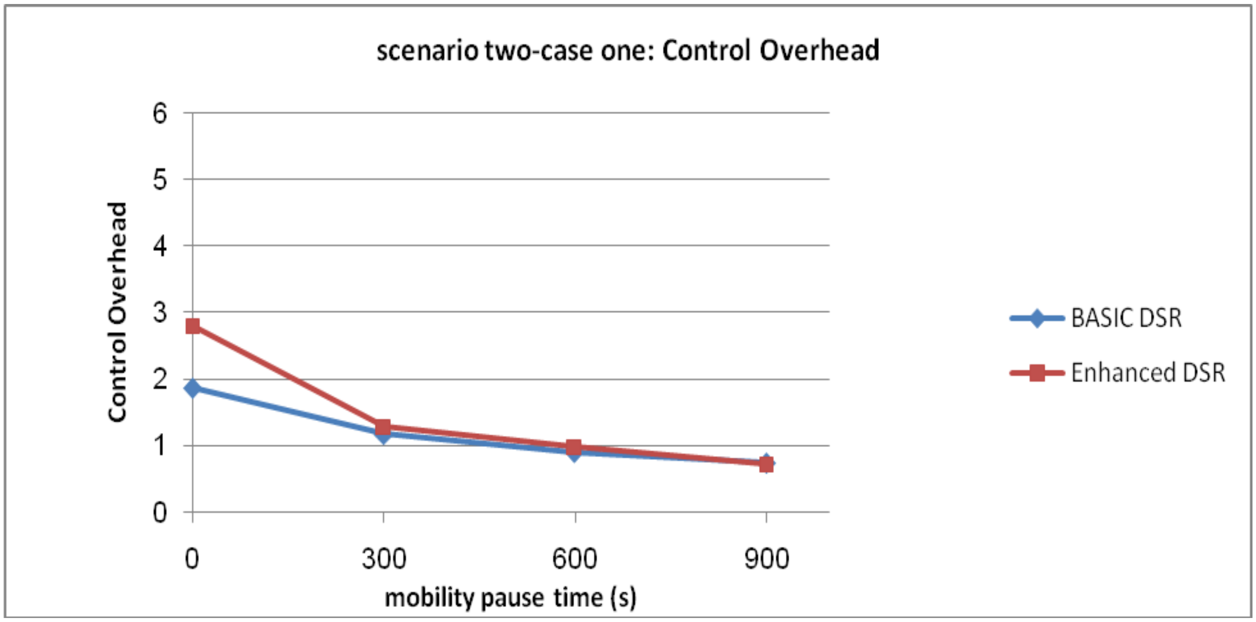


Figure 6.17. Control Overhead of 10 sources each sends 1PPS

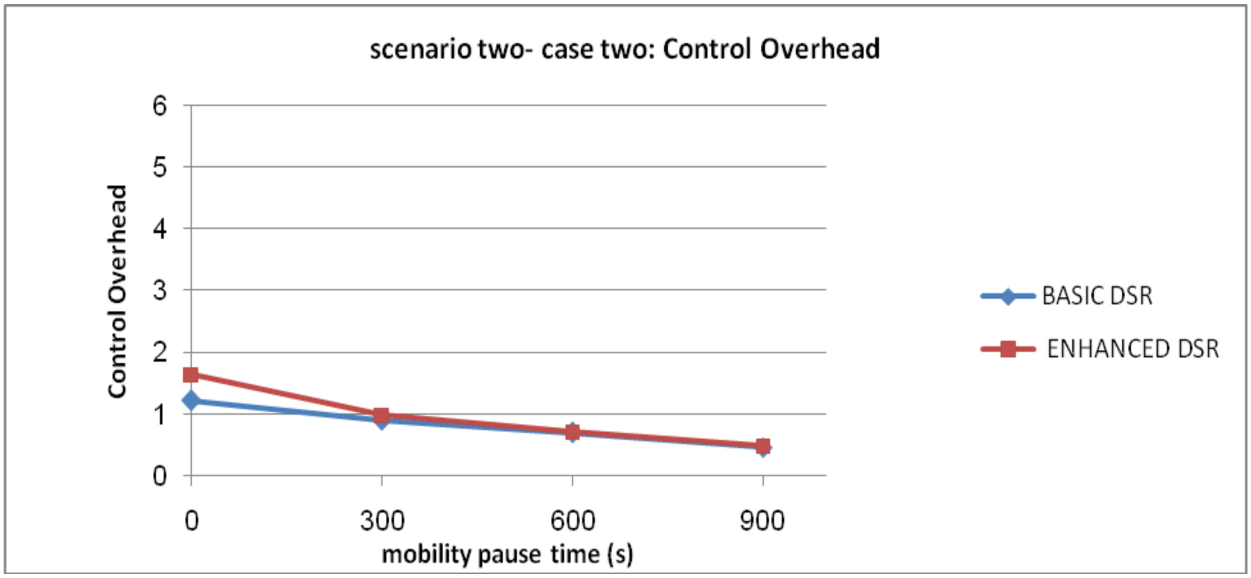


Figure 6.18. Control Overhead of 10 sources each sends 2PPS

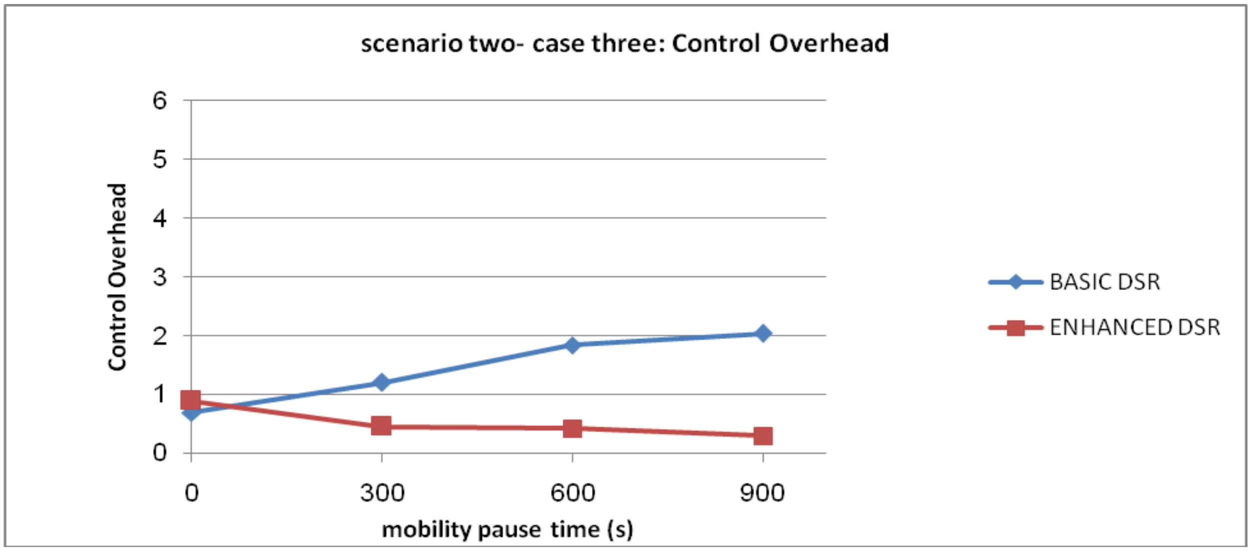


Figure 6.19. Control Overhead of 10 sources each sends 4PPS

- ***Scenario Three- Case One:*** Fifteen CBR sources are chosen with transmission rate of one packet per second. In this case, both of the two protocols decrease their Control Overhead along all mobility pause time values. EDSR outperforms Basic DSR in all mobility levels, except in a high mobility level, where it has the worst performance for the same reason discussed above, see Figure(6.20). We can say that EDSR improves Control Overhead of Basic DSR.

- ***Scenario Three- Case Two:*** Fifteen CBR sources are selected with transmission rate of two packets per second. In this case, both protocols decrease Control Overhead for all pause times, see Figure(6.21). The Control Overhead of both of the two protocols have convergent Control |Overhead in all mobility levels. However, EDSR outperforms Basic DSR in all mobility levels by about (9%, 29%, 24%, 42% respectively). Note that the highest mobility level the worst EDSR performance for the same reasons mentioned above. We can say that, EDSR improves the Control Overhead over Basic DSR.

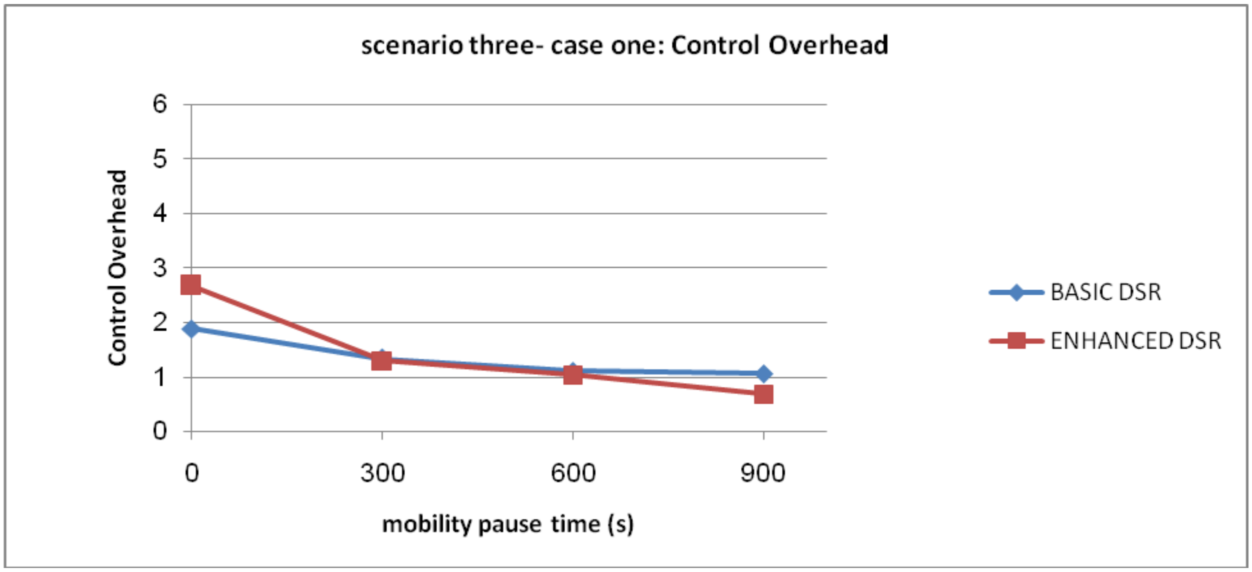


Figure 6.20. Control Overhead of 15 sources each sends 1PPS

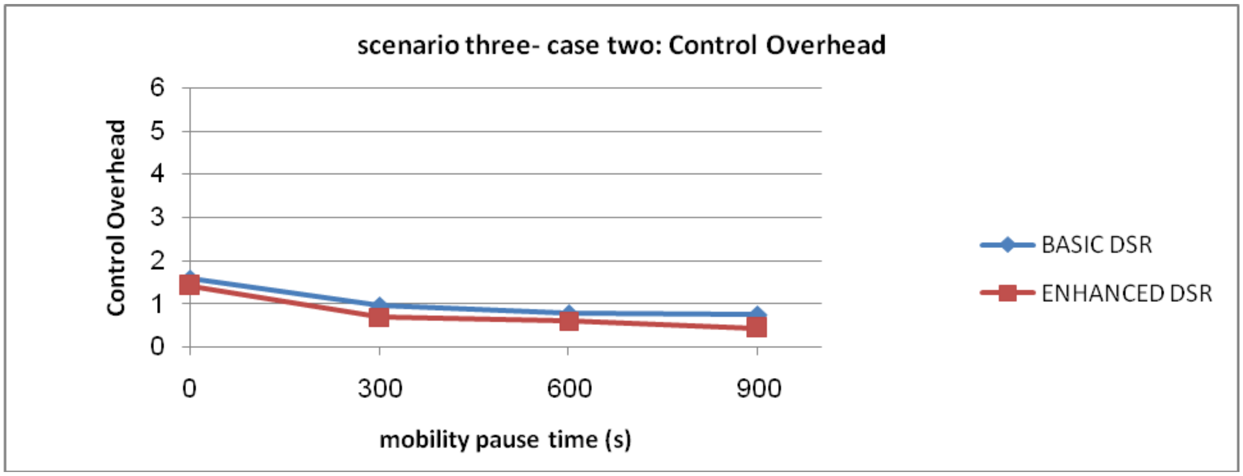


Figure 6.21. Control Overhead of 15 sources each sends 2PPS

- ***Scenario Three- Case Three:*** In this case, fifteen CBR sources are selected with transmission rate of four packets per second. As shown in Figure (6.22), the Control Overhead for both of the two protocols decrease at all pause time values. Control Overhead for both of the two protocols is convergent. However, EDSR outperforms Basic DSR in all mobility levels by about (1%, 43%, 44%, 63% respectively). Note that, EDSR has the worst performance at high mobility for the same reasons discussed above. In this case we can say that EDSR improves Control Overhead over Basic DSR.

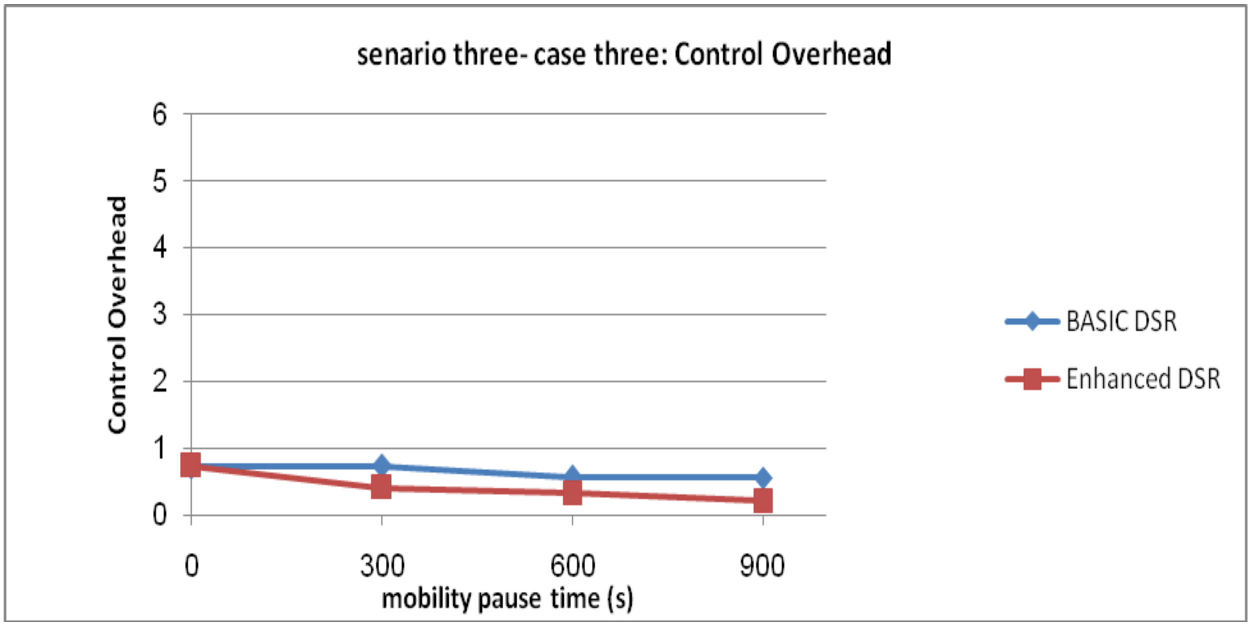


Figure 6.22. Control Overhead of 15 sources each sends 4PPS

6.4. Evaluation Results: Average End-to-End

In this section, we consider the relationship between mobility levels and Average End-to-End delay metric of both Basic DSR and EDSR protocols for the nine cases. The simulation results for the nine cases will be analyzed to study the performance of EDSR compared with Basic DSR from End-to-End delay point of view. The analyses of the nine cases are illustrated below:

- ***Scenario One- Case One***: Five CBR sources are selected with transmission rate of one packet per second. In this case, EDSR outperforms Basic DSR at two points (pause time= {600S and 900S}) by about (40% and 24% respectively), as shown in Figure (6.23). Basic DSR outperforms EDSR at the other two points (pause time= {0S and 300S}) by about (14 % and 8 % respectively). We can say that, EDSR upgrades Average End-to-End delay at a moderate mobility levels specially, at pause time=600S. However, EDSR degrades the Average End-to-End delay in high mobility levels. This result because cache entries change frequently so that any cache's check may give negative results, since links is broken always. So that, cache checking may degrade End-to-End delay.

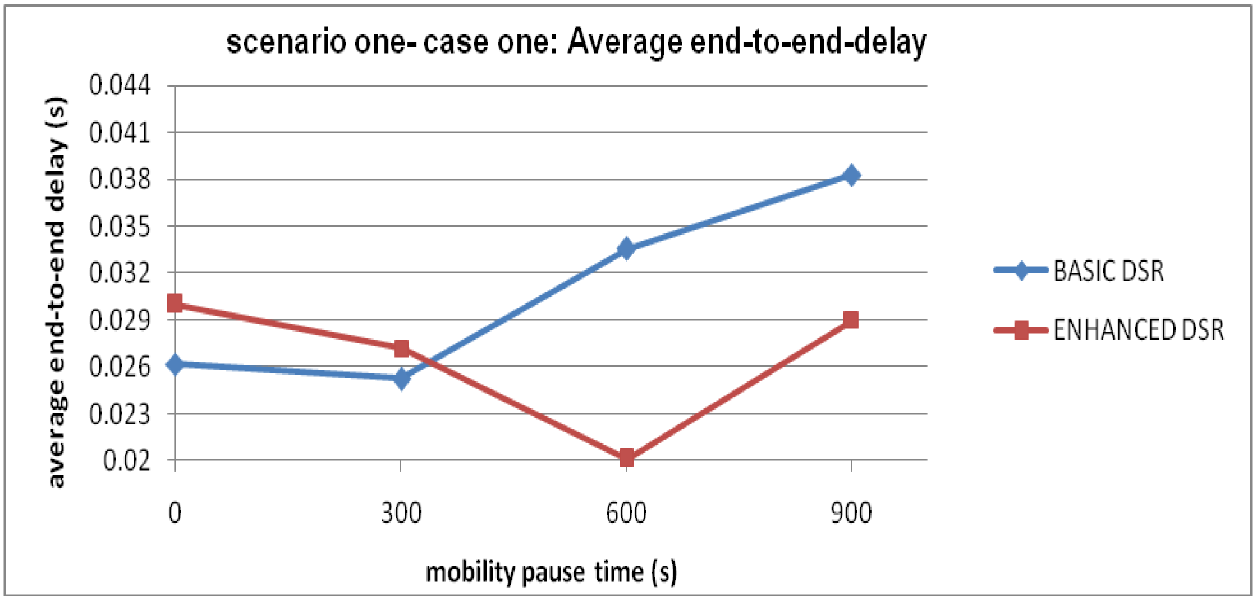


Figure 6.23. Average End-to-End delay of 5 sources each sends 1PPS

- ***Scenario One- Case Two:*** Five CBR sources are selected with transmission rate of two packets per second. In this case, EDSR outperforms Basic DSR in all mobility levels, as shown in Figure(6.24). EDSR outperforms Basic DSR by about (30%, 43%, 41%, 30% at all pause time respectively). The highest improvement is at moderate mobility level, specially, at pause time={300S and 600S}. As in previous case, EDSR has worst performance at high mobility for the same reason mentioned in previous cases. In this case, we can say that there is a good improvement of EDSR over Basic DSR.

- ***Scenario One- Case Three:*** Five CBR sources are selected with transmission rate of four packets per second. In this case, EDSR outperforms Basic DSR in three mobility levels (pause time= 300S, 600S, and 900S) by small percents (6% as maximum). However, Basic DSR outperforms EDSR only at a high mobility, as illustrated in Figure (6.25). Although the improvement of EDSR is low, but we are concerned with the improvement over Basic DSR. The highest improvement in Average End-to-End delay is at moderate mobility level as in all cases of scenario one, specially at pause time={300S and 600S}. In this case, we can say that EDSR improves Average End-to-End delay of Basic DSR.

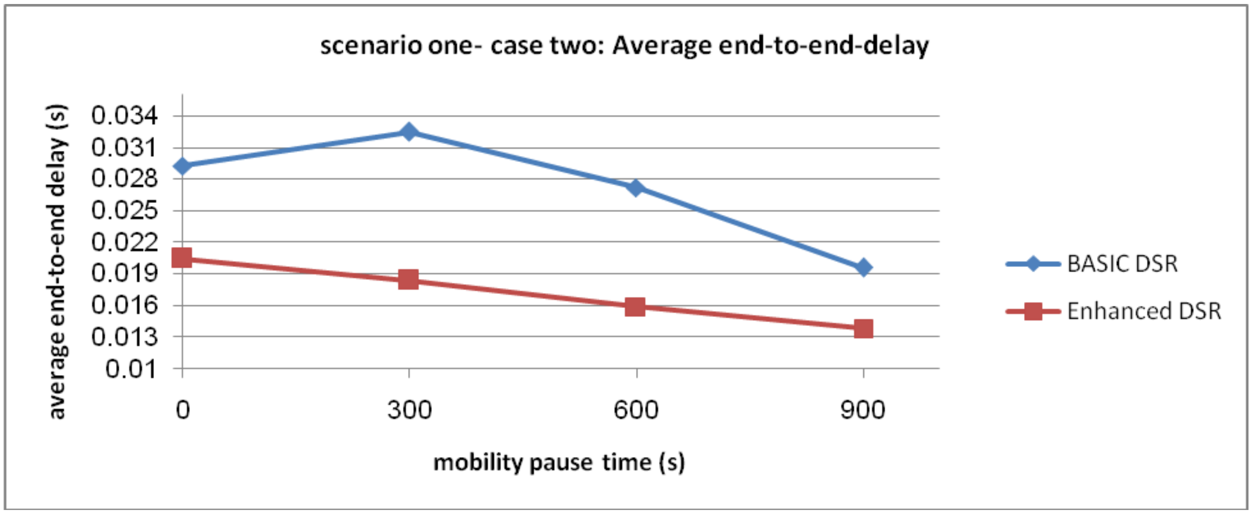


Figure 6.24. Average End-to-End delay of 5 sources each sends 2 PPS

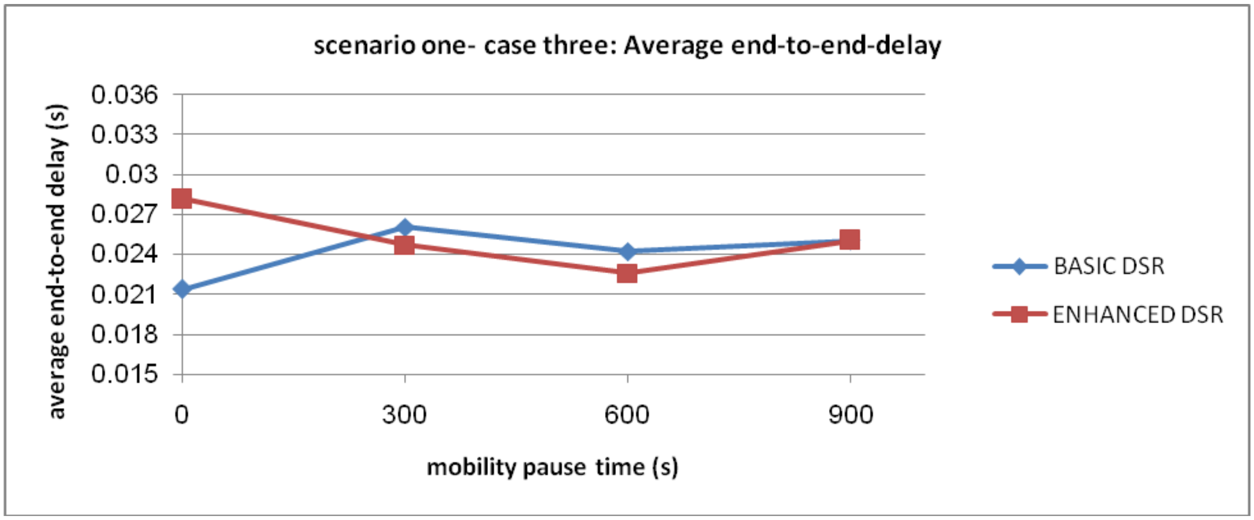


Figure 6.25. Average End-to-End delay of 5 sources each sends 4PPS

- ***Scenario Two- Case One:*** Ten CBR sources are selected with transmission rate of one packet per second. In this case, EDSR has approximately constant behavior for all mobility levels, while Basic DSR increases its Average End-to-End delay as pause times increases. Basic DSR outperforms EDSR in two mobility levels (pause time= {0S and 300S}) by about {35% and 17% respectively). However EDSR outperforms Basic DSR in two mobility levels (pause time= {600S and 900S}) by about (29% and 60% respectively), as shown in Figure(6.26). In this case, we can say that there is a good improvement of Average End-to-End delay of EDSR over Basic DSR at a low mobility levels.

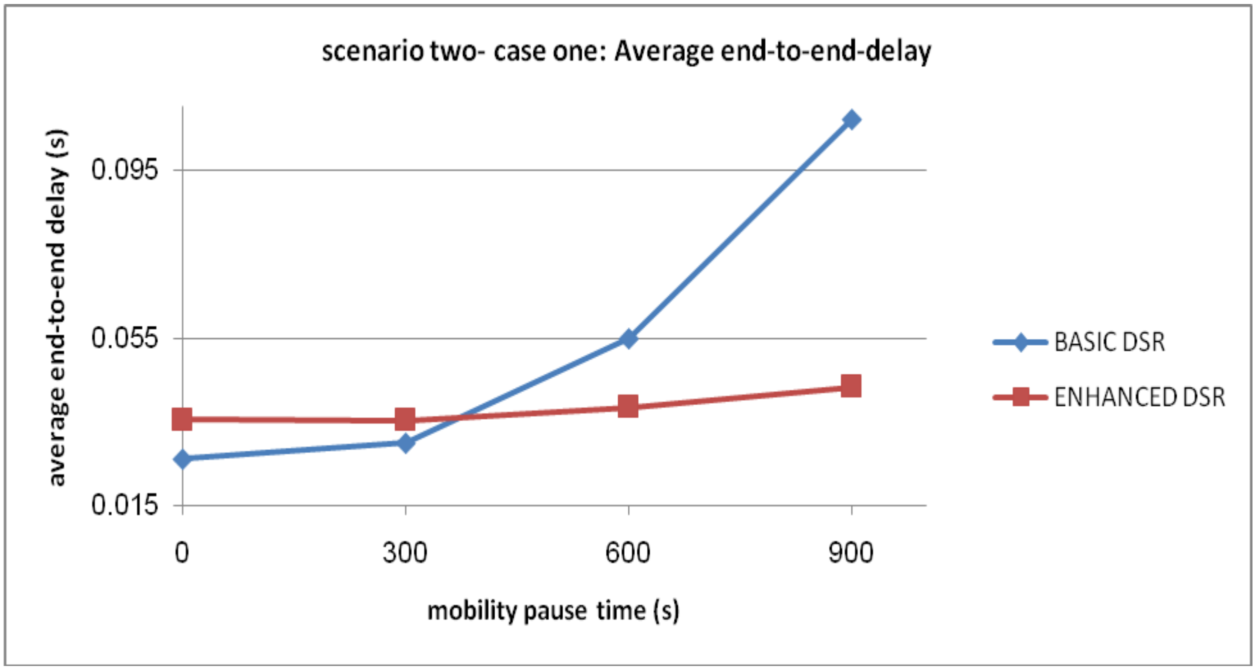


Figure 6.26. Average End-to-End delay of 10 sources each sends 1PPS

- ***Scenario Two- Case Two:*** Ten CBR sources are selected with transmission rate of two packets per second. In this case, Basic DSR outperforms EDSR in all mobility levels by about (14%) (as a maximum value), as shown in Figure(6.27). The best Average End-to-End delay value of both of the two protocols is at high mobility. However, the worst value is at low mobility. In other word, the highest mobility level, the best Average End-to-End delay. In this case, EDSR degrades the average End-to-End delay performance compared to that of Basic DSR.

- ***Scenario Two- Case Three:*** Ten CBR sources are selected with transmission rate of four packets per second. In this case, Basic DSR outperforms EDSR in approximately all mobility levels (as shown in Figure (6.28)). So we can deduce that there is no improvement of EDSR over Basic DSR in this case and in scenario two at overall.

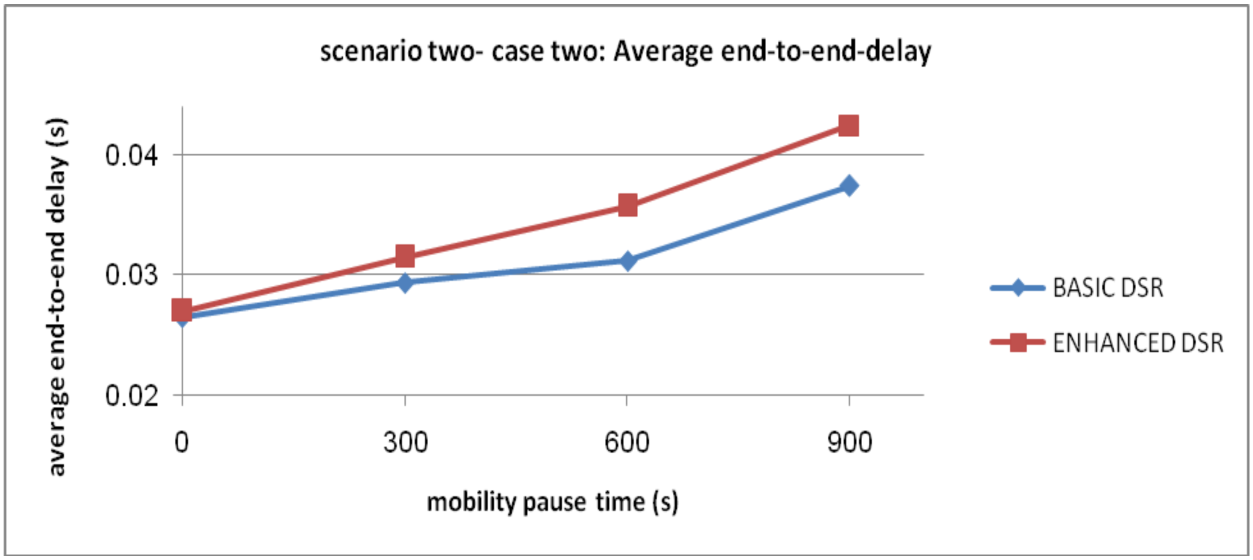


Figure 6.27. Average End-to-End delay of 10 sources each sends 2PPS

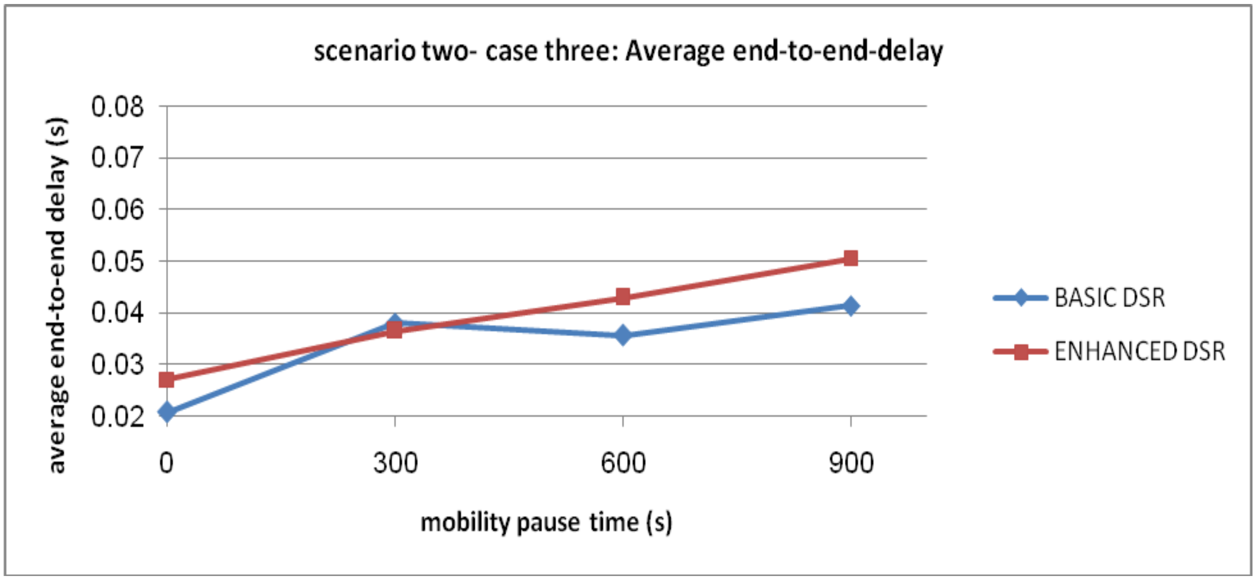


Figure 6.28. Average End-to-End delay of 10 sources each sends 4PPS

- ***Scenario Three- Case One:*** Fifteen CBR sources are selected with transmission rate of one packet per second. In this case, Basic DSR outperforms EDSR only when mobility is high by about (8%) as shown in Figure (6.29). However, EDSR outperforms Basic DSR in three mobility levels (pause time= {300s, 600s and 900s}) by about (3%, 8%, 5% respectively). In this case, we can say that EDSR has little improvement of Average End-to-End delay over Basic DSR.

- ***Scenario Three- Case Two:*** Fifteen CBR sources are selected with transmission rate of two packets per second. In this case, EDSR outperforms Basic DSR at high mobility (pause time= 900s)by about (7%). However, Basic DSR overtakes EDSR in the other mobility levels by about (6%, 7%, and 17% respectively), as shown in figure (6.30). In this case, we can’t say that there is any improvement in the Average End-to-End delay of EDSR over Basic DSR.

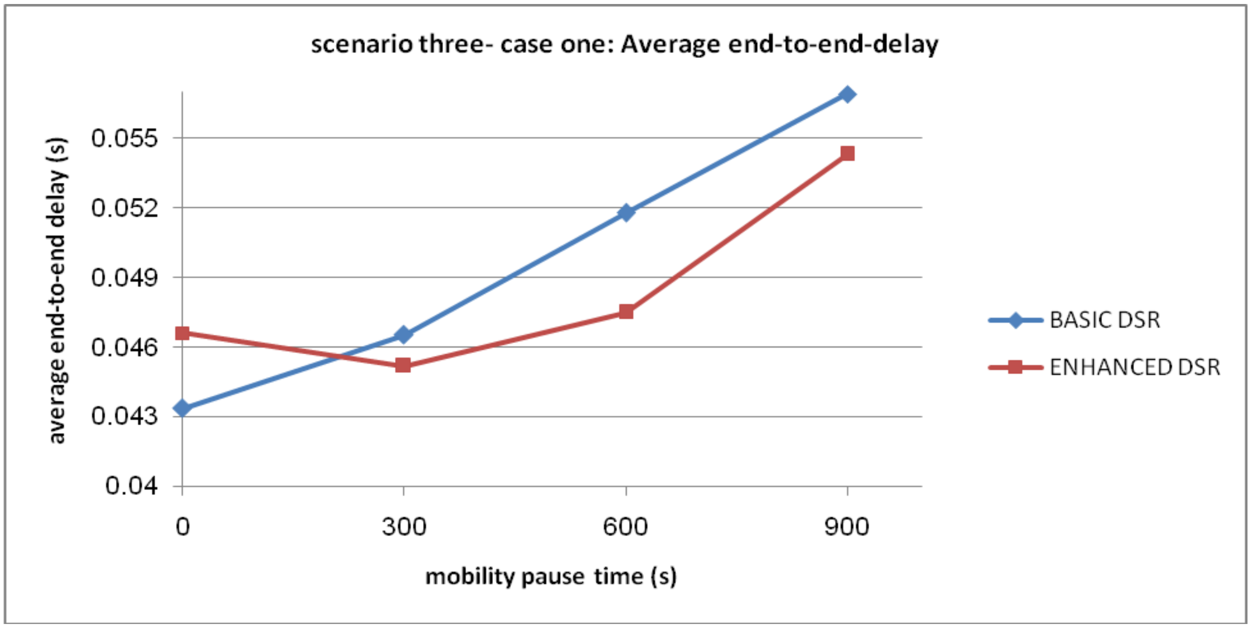


Figure 6.29. Average End-to-End delay of 15 sources each sends 1PPS

- ***Scenario Three- Case Three:*** Fifteen CBR sources are selected with transmission rate of four packets per second. In this case, EDSR outperforms Basic DSR in three mobility levels (pause time= {0S, 300S and 600S}) by about (27%, 20% and 20% respectively), as shown in figure (6.31). However, Basic DSR overtakes EDSR only at low mobility by small percent (6%). In this case, EDSR has good improvement in the Average End-to-End delay over Basic DSR. At overall, there is admissible improvement of Average End-to-End delay of EDSR in scenario three, specially, at moderate mobility level.

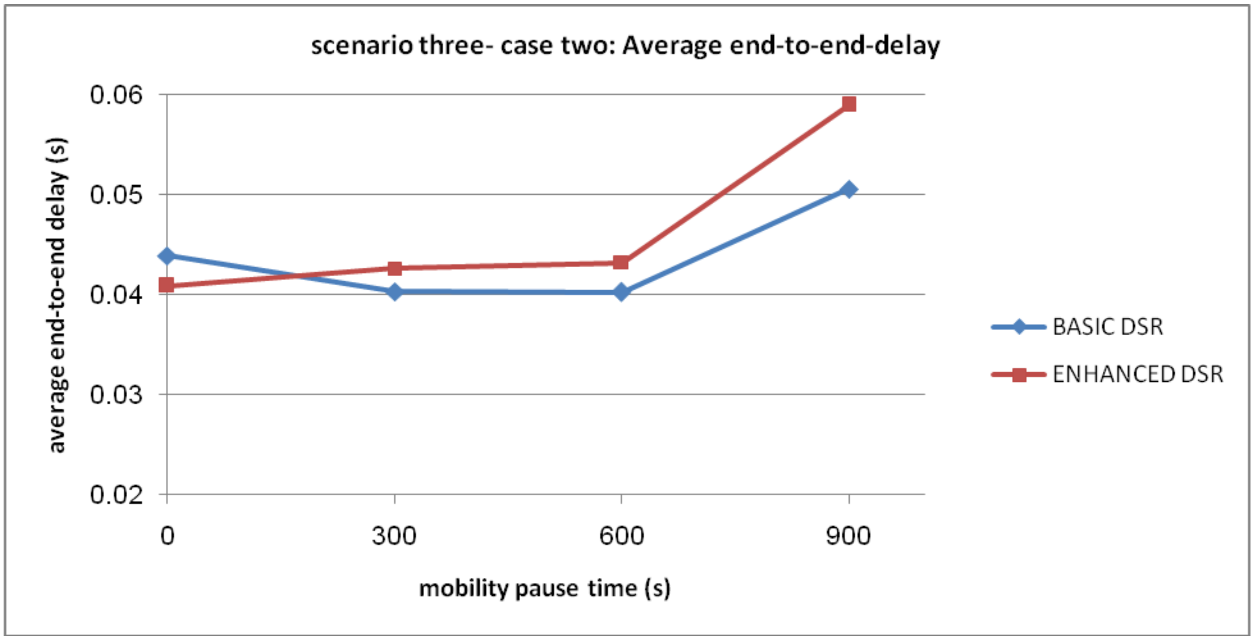


Figure 6.30. Average End-to-End delay of 15 sources each sends 2PPS

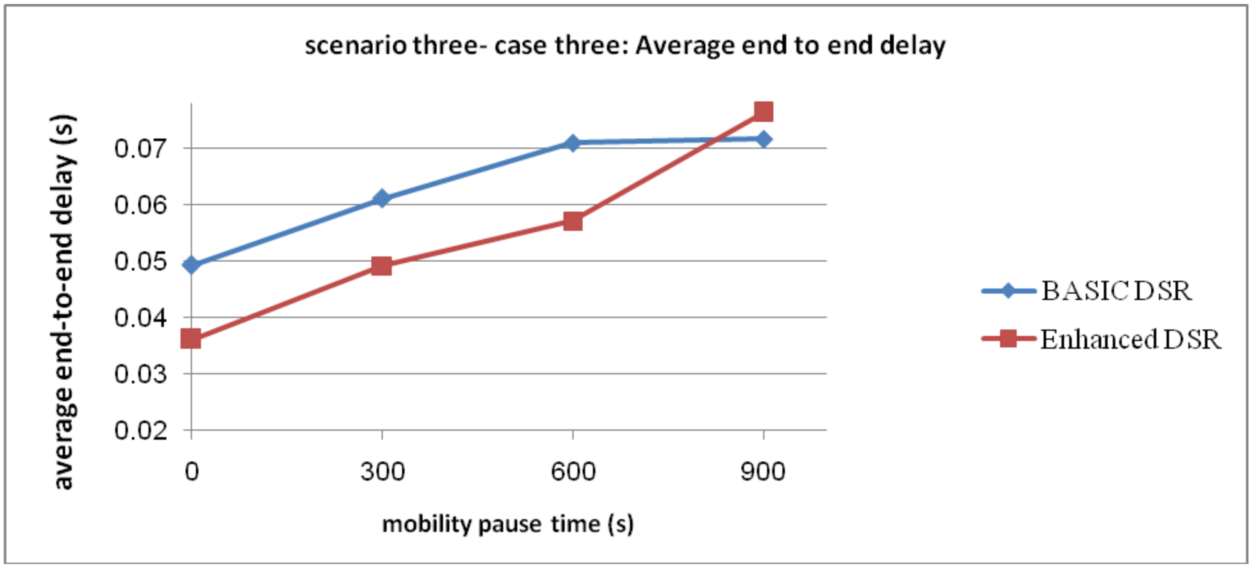


Figure 6.31. Average End-to-End delay of 15 sources each sends 4PPS

6.5. Evaluation Results: Throughput

In this section, we consider the relationship between mobility levels and throughput metric of both Basic DSR and EDSR protocols for one case: scenario one- case one. The results of this relationship are analyzed to study the behavior of EDSR compared with Basic DSR in this case.

Figure(6.32) show the Throughput result of scenario one- case one ,where five CBR sources are selected with transmission rate of one packet per second. In this case, EDSR and Basic DSR approximately have the same throughput results.

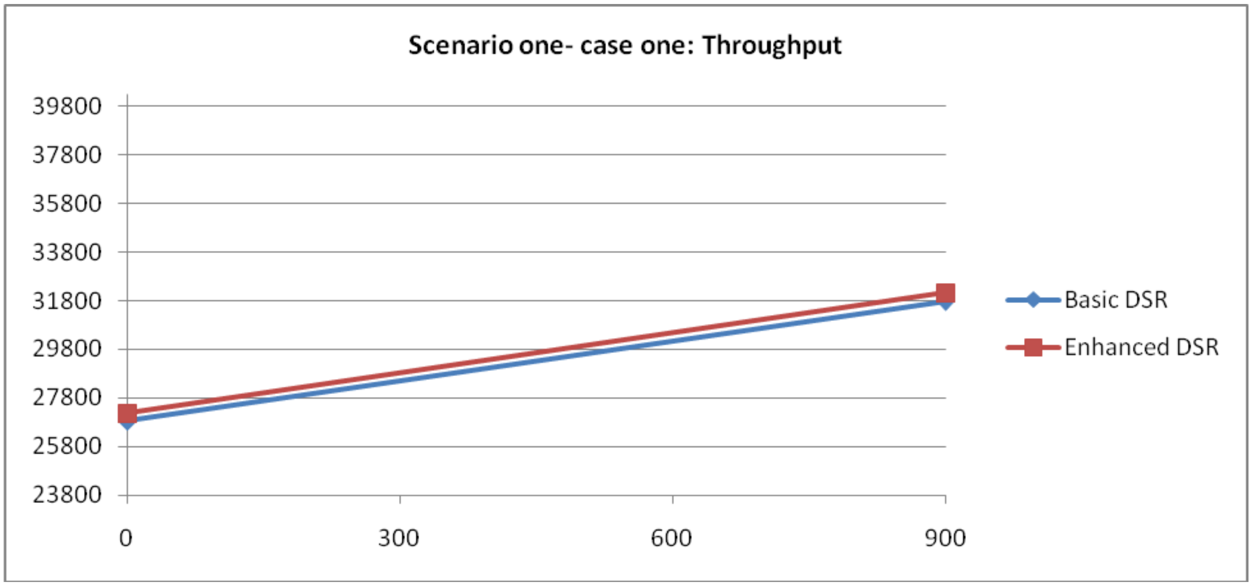


Figure 6. 32. Throughput of 5 sources each sends 1PPS

6.6. Summary of Evaluation Metrics Results:

The results section shows the network performance of EDSR and DSR with line graphs. In this section we summarize the results of EDSR and DSR based on the three evaluation metrics.

6.6.1. Packet Delivery Ratio:

The enhanced protocol provides better performance than the original. EDSR has considerable improvements of the delivery ratio over Basic DSR in almost all scenarios. The results of PDR of DSR and EDSR are shown on the figures (6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13). Its percent improvement is between 2% up to 22%. It is also obvious that the best improvement of PDR is when mobility is high. This is due to the following reasons: EDSR never reply the cached routes immediately. Instead of directly reply the cached route, EDSR has to decide to reply the cached route based on the selected threshold value. This reduces the opportunity of utilization of the staled routes which add to the packet drop rates. From the figures of PDR, we can see that, in a high mobility, a link may break soon, so that using cache based on threshold is better than using cache immediately. However, in a low mobility, the node doesn't move during simulation time. Each source node discover route to destination just once because there is no mobility. The chance of link failure is low. In this case, using cache is better because it may have a fresh entries. As conclusion, we can say that we achieve our aim of this thesis, which is improve PDR metric.

6.6.2. Control Overhead:

Figures (6.14, 6.15, 6.16, 6.17, 6.18, 6.19, 6.20, 6.21, 6.22) show that the results of Control Overhead of both DSR and EDSR. We can see that EDSR has admissible improvements in Control Overhead over Basic DSR specifically in scenario three, where the traffic is high. The two protocols have convergent values of Control Overhead in the other two scenarios. This reason because source nodes send either one packet per second in scenario one, two packets per second in scenario two, and four packets per second in scenario three. As instance, after two second, a link may be failed. In scenario one, source node sends just two packets (1PPS) before link failure. In scenario two, source node sends four packets (2PPS) before link failure. However, in scenario three, source node sends eight packets (4PPS) before link failure. We have to send 1000 packets by each source node during simulation time. So that node need to discover new route 500 times in scenario one, 250 times in scenario two, and 125 times in scenario three. This route discovery requires extra control packets, which increase the Control Overhead. The Control Overhead's percent improvement of scenario three is between 2% up to 63%. Note that the highest improvement of Control Overhead is at low mobility. However, at high mobility, there is degradation of Control Overhead. This reason is because nodes move continually at high mobility. So that, the probability of link to fail is high. Extra control packets is needed to discover new route, which also increase Control Overhead.

In our work, Control Overhead is affected by two factors: positive factor and negative factor. The positive factor yields due to applying freshness criteria in caching routes. Ensure satisfying freshness criteria reduce link failure. As a reason, broadcasting of both RREQ and RREP control packets are reduced. This factor improves Control Overhead. The negative factor yields due to reduction of using route cache. Instead of replying the route cache immediately, EDSR continues of broadcasting RREQ based on threshold value. This factor increases Control Overhead due to increase the number of control packets. In simulation we cannot predict which factor is stronger than the other. So that, Control Overhead results sometimes go up and sometimes go down.

6.6.3. Average End-to-End delay:

Figures (6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31) show the results of Average End-to-End delay of both DSR and EDSR. From these figures, we can see that, EDSR improves Average End-to-End delay in both of scenario one and scenario three. However, in scenario two, Basic DSR shows better Average End-to-End delay than EDSR. The Average End-to-End delay's percent improvement of scenario one and three is between 0.1% up to 27%. EDSR has high improvement of average End-to-End delay at moderate mobility level. This is because, at high mobility, cache entries may be stale quickly. At low mobility, nodes does not move during simulation time, so that the route to specific destination is discovered once. At this case, cache entries may absolutely fresh. Checking cached route, when cache entries absolutely fresh or absolutely stale, may degrade delay metric. When cache entries is absolutely fresh, then always reply the cached route is effective without checking cache. When cache entries is absolutely stale, then always reply the cached route is ineffective without checking cache.

In general, there is trade-off between PDR and both Control Overhead and Average End-to-End delay. When PDR is improved, Control Overhead and Average End-to-End delay may be degraded. PDR is more important metric than Control Overhead and Average End-to-End-delay, since we are concerned with delivering data packet rather than how many control packets needed to send this data packet or how long it delayed until it has been delivered. EDSR improves PDR of Basic DSR, while the Control Overhead and Average End-to-End delay still in admissible values and they are improved in most cases.

Chapter Seven

Conclusion and Future Works

This chapter illustrates the main points investigated from developing EDSR protocol and simulating it in comparison with Basic DSR.

7.1. Conclusion

In this work we study the behavior of Basic DSR as an Ad-hoc routing protocol. We criticism it in case of using cache in route discovery process. In route discovery process. We proposed EDSR protocol, which deals with this case, to improve the performance of Basic DSR. Threshold based technique is implemented in the proposed EDSR to decide when do we use the cache and when do we stop using it.

We employed two caching criteria that any route must satisfy to be cached: freshness and hop count of incoming route. These criteria used to investigate limited cache size and to obtain fresher routes.

To determine suitable threshold value, we carried out an empirical study using several experiments throughout two different simulation scenarios. Based on these experiments, we conclude that the suitable threshold value is half the network diameter. We chose this threshold value because it has given good results in comparison with Basic DSR in terms of PDR and Control Overhead.

Based on this threshold value, we evaluated the proposed EDSR throughout three different scenarios each has three different cases that selected by different CBR sources and packet transmission rate. The simulation results of EDSR show high improvements in packet delivery ratio over Basic DSR in all scenarios. Highest improvement is achieved specifically at a high mobility. The percent improvement of PDR is up to 22%.

We also achieved an improvement of Control Overhead up to 63% in scenario three. When mobility is high, EDSR degrades Control Overhead performance. However, it upgrade it at low mobility. In our work, Control Overhead is affected by two factors: negative and positive. It may be affected by negative factor due to little use of route cache and broadcasting more RREQ. However, positive factor is activate due to reduction of link failure since freshness criteria employed in caching routes process. This factor reduce broadcasting of RREQ in indirect way. We did not determine which factor is stronger than the other. We use simulation to show which factor is strongly affects the Control Overhead. Evaluation results show that sometimes control overhead affects by negative factor and sometimes it affects by positive factor.

Average End-to-End delay is also improved, at scenario one and scenario three. High improvement of Average End-to-End delay is at moderate mobility speed. The improvement in Average End-to-End delay is up to 27%. Throughput metric is the same in both of Basic DSR and EDSR.

We deduced that there is trade-off between PDR, and both control overhead and Average End-to-End delay. The higher the PDR improved, the lower the Control Overhead and Average End-to-End delay. PDR is more important evaluation metric than the other two evaluation metrics, because we concerned with delivering data packet rather than concerning of how many control packets and how long it take to send this data packet. Although both of these two metrics are important to improve network performance.

7.2 Future Works

During our study, we take a future research directions, some are short terms and others are long term. In this section, We propose number of perspectives which aim to provide more investigations in the area of route caching. These perspectives are as follow:

- Evaluate EDSR using another ad hoc routing protocol that uses route cache.
- In our work we use two cached route per destination. We are going to study the effects of implementing EDSR using different number of cached routes per destination.
- Studying the effects of using different network density by varying some simulation parameters such as, terrain area and number of nodes.

References

1. A. Adane, "***Active communication energy efficient routing protocol of mobile ad hoc networks (manets)",*** M.Sc. Thesis, Engineering Department, Addisababa university, ethiopia, april, 2008, available at:*"http://etd.aau.edu.et/dspace/handle/123456789/1531?mode=full".*
2. A. Gani, M. Hassan, A. Zaidan and B. Zaidan, **"Intelligent routing information protocol using full triggered update mechanism",** International Journal of the Physical Sciences, Vol. 6, pp. 2750-2761, ISSN 1992 – 1950, June, 2011.
3. A. Gani, Q. Han, N. BadrulAnuar, O. Zakaria, "***Enhancing DSR Protocol Performance in Mobile Ad Hoc Network Using ACK Reply***", WSEAS TRANSACTIONS ON COMMUNICATIONS, ISSN: 1109-2742, Issue 2, Vol. 8, pp. 227-236, Feb. 2009, available at: "*http://airccse.org/journal/jcsit/0202csit10.pdf"*.
4. C. Bettstetter, H. Hartenstein, X. P´erez–Costa, "**Stochastic Properties of the Random Waypoint Mobility Model**", Kluwer Academic Publishers. Printed in the Netherlands, 2003, available at: *"http://dl.acm.org/citation.cfm?id=1027342&CFID=125406386&CFTOKEN=22839604".*
5. C. perkins, E. Royer, S. Das, M. Marina, "***Performance comparison of two on demand routing protocols for ad hoc networks***", IEEE personal comm., vol.8, no.1, pp.16-28 Feb. 2001.
6. D. Johnson and D. Maltz, "***Dynamic Source Routing in Ad Hoc Wireless Networks***", Mobile Computing, Edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, Chapter 5, pp. 153–181, 1996.
7. D. Johnson, D. Maltz, J. Broch, "***The Dynamic Source Routing protocol for Mobile Ad Hoc Networks in IPv4***", in Network Working Group, Request for Comments (RFC):4728, Microsoft Research in feb.2007. Copyright (C) The IETF Trust (2007). available at: http://tools.ietf.org/html/rfc4728#section-8.6.7. Visited on 11/December/2011.
8. D. Jörg, "***Performance Comparison Of MANET Routing Protocols In Different Network Sizes"***, Computer Science Project, University of Berne, Switzerland, 2003.
9. I. Chlamtac, M. Conti, and J. Liu, "***Mobile ad hoc networking: imperatives and challenges*",** Elsevier, Ad Hoc Networks 1, pp 13–64, 2003.

10. J. Boyer, D. Falconer, and H. Yanikomeroglu, "**A Theoretical Characterization of the Multihop Wireless Communications Channel",** Global Telecommunications Conference, GLOBECOM '01, IEEE, vol. 2, pp. 841-845, 2001.
11. J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, ***A performance comparison of multi-hop wireless ad hoc network routing protocols***, In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pages 85–97, Oct. 1998.
12. K. Levchenko, G. Voelker, R. Paturi, and S. Savage, **"Xl: An Efficient Network Routing Algorithm**", ACM, SIGCOMM08, USA, pp. 15-26, 2008.
13. L. Conrad , "***Secure Routing in Mobile Ad Hoc Networks***", M.Sc. Thesis ,Technical University of Denmark (DTU), 2003.
14. L. Zhou Z. Haas, "**Securing Ad Hoc Networks**", IEEE NETWORK MAGAZINE, DEC. 1999.
15. M. Neelakantappa, B. Satyanarayana, A. Damodharam, "***A Stable Adaptive Optimization for DSR Protocol in Ad hoc Networks***", Int. J. of Advanced Networking and Applications vol. 1 Issue: 03 pp. 170-174, 2009.

16. S. Biswal, S. Mohanty, and D. Seth, "***Study of DSR Routing Protocol in Mobile Adhoc Network",*** International Conference on Information and Network Technology, IPCSIT, vol.4, 2011.

17. S. Rizvi, M. Jafri, and K. Elleithy, "***A Novel Optimization of the Distance Source Routing (DSR) Protocol for the Mobile Ad Hoc Networks (MANET),***" International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 08), December 5 - 13, 2008.

18. S. Sultana, S. Begum, N. Tara, A. Chowdhury, "***Enhanced-DSR: A New Approach to Improve Performance of DSR Algorithm***", International Journal of Computer Science and Information Technology, vol. 2, no. 2, pp. 113-123, April, 2010.
19. W. Lou and Y. Fang, "***Predicative Caching Strategy for On- Demand Routing Protocols in Wireless Ad Hoc Networks***", Wireless Networks, vol. 8, no. 6, pp. 671-679, 2002.

20. X. Yu, "Distributed ***Cache Updating for the Dynamic Source Routing Protocol***", IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 609-626, Jun., 2006.

21. Y. Hu, and D. Johnson, ***"Ensuring Cache Freshness in On-Demand Ad Hoc Network Routing Protocols***", Published by ACM 2002 Article, pp. 25-30.

22. Zeng, X, Bagrodia, R and M, Geria, "**GloMoSim: A library for parallel simulation of large-scale wireless networks**", proceeding of the 12th Workshop on Parallel and Distributed Simulations (PADS), Banff, Alberta, Canada, May 1998.

Appendix (A)

Packet Header format

In this appendix, we present a brief description of the format of all packets header that used in both route discovery and route maintenance mechanisms: RREQ,RREP, and RERR.

Route Request packet Header:

The Route Request option header included in DSR header option format is encoded as follows:

Table A. 1. Route Request header format

Option Type	Opt Data Len	Identification
Target Address		
Address[1]		
Address[2]		
...		
Address[n]		

Option Type: represents type of this option within DSR option header

Opt Data Len: it must be set equal to (4×n) + 6, where n represents the number of addresses in the Route Request Option.

Identification: is a unique number generated by initiator of RREQ to determine the precedence of RREQ received by any node.

Target address: represents the address of node that is a target of RREQ.

Address[1..n]: represents all nodes recorded as source route to destination.

Route Reply packet Header:

The Route Reply option header included in DSR header option format is encoded as follows :

Table A. 2. Route Reply header format

Option Type	Opt Data Len	Last Hop External (L)	Reserved
Address[1]			
Address[2]			
...			
Address[n]			

Option type: represents type of this option within DSR option header

Opt Data Len: it must be set equal to (4 * n) + 1, where n represents the number of addresses in the Route Reply Option.

Last Hop External(L): a flag set to indicate that last hop of path stored in route cache is not an internal to DSR networks**.**

Reserved: must set to 0 and receiver discard it.

Route Error packet Header:

The Route Error option header included in DSR header option format is encoded as follows:

Table A.3. Route Error header format

Option Type	Opt Data Len	Error Type	Reserved	Salvage
Error Source Address: represents the address of node triggering RERR.				
Error Destination Address: represents the address of node to which RERR must be delivered.				
Type-Specific Information: represents information that specifies error type of RERR.				

Option Type: represents type of this option within DSR option header.

Opt Data Len: it must be set to 10 plus the size of any Type-Specific Information in route error option.

Error Type: represent the type of triggered error, it may be one of the following values:

1 = NODE_UNREACHABLE

2 = FLOW_STATE_NOT_SUPPORTED

3 = OPTION_NOT_SUPPORTED

Reserved: must set to 0 and receiver must discard it.

Salvage: determines how many times this packet has been salvaged *(RFC4728).*

الملخص

DSR هو احد البروتوكولات المستخدمة في إيجاد المسارات في الشبكات الخاصة المتحركة. يستخدم DSR الذاكرة السريعة لتسريع عملية إيجاد المسار. خلال عملية البحث عن مسار ، قد تحوي ذاكرة العقد الوسطية على مسار للمستقبل. عندها تقوم العقد الوسطية بإرسال رد للمرسل. قد تحدث مشكلة عندما تكون العقدة التي أرسلت قريبة من المستقبل, حيث من الممكن أن يكون المسار الموجود قديم ومن المحتمل إيجاد مسار أحدث بدون استخدام الذاكرة السريعة.

السؤال, متى نستخدم الذاكرة السريعة ومتى نتوقف عن استخدامها؟ وما هي المعايير التي نستخدمها عند تخزين المسارات؟

للإجابة عن السؤال، قمنا بإجراء دراسة تجريبية لتحديد عتبة نستخدم عندها الذاكرة السريعة. وتبين لنا من خلال التجارب أن أفضل عتبة هي نصف محيط الشبكة حيث أظهرت نتائج جيدة. كما تم السماح بتخزين مسارين فقط لكل مستقبل. وتم اقتراح معيارين لتخزين هذين المسارين: معيار الحداثة و طول المسار.

لتقييم ما قمنا به, تم استخدام محاكي GLOMOSIM. حيث تم التقييم اعتمادا على ثلاث مقاييس: Packet Delivery Ratio (PDR)، Control Overhead and Average End-to-End delay. أظهرت نتائج المحاكاة تحسن في PDR بنسبة تصل إلى 22%. كما أظهرت تحسن على Control Overhead بنسبة تصل إلى 63%. أيضاً تحسن Average End-to-End delay بنسبة تصل إلى 27%.