Faculty of Computer Science & Information Technology

ENHANCED DYNAMIC SOURCE ROUTING (EN-DSR) WITH QOS BASED ROUTING FOR IPV6 IN AD HOC NETWORKS

TAI WAI YEE

Master of Science
2013

ENHANCED DYNAMIC SOURCE ROUTING (EN-DSR) WITH QOS BASED ROUTING FOR IPV6 IN AD HOC NETWORKS

TAI WAI YEE

A thesis submitted
in fulfillment of the requirements for the degree of
Master of Science

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
2013

DECLARATION

No portion of the work referred to in this thesis has been submitted in support of an application for another postgraduate or qualification of this or any other university or institution of higher learning.

(Tai Wai Yee) June 2013

ACKNOWLEDGEMENTS

A great heartfelt gratitude is dedicated to two supervisors of mine for their guidance and advices along the way of this research. They are Dr. Tan Chong Eng and Mr. Lau Sei Ping. These two great men gave me guidance and encouragements throughout the whole process of completing this thesis. Thank you very much to both of you.

Besides, my sincere appreciation and love are given to my parents, both sides of my grandma, sister and brother and of course him who shows me unconditional supports during the years. The comfort and encouragement are precious that made me feel loved and motivated to work harder which lead to today's achievement.

More importantly, thank you very much for the government's support on providing Mini Budget Scholarship that supported my research school fees. Universiti Malaysia Sarawak (UNIMAS) for their Zamalah scholarship that helped me went through the whole research with financial aids and as well realizing my dream of achieving this in my life.

Last but not least, to my friends who are there constantly in telling me to graduate as soon as possible which have motivated me even more. Elise, Ling Sun and Wei Jan, I might not be able to make it without you guys. I would like to express my sincere appreciations to everyone who are with me all these while. To all those I have mentioned above and anyone who I might have left out here, thank you very much.

ABSTRAK

Rangkaian ad hoc bergerak merupakan satu koleksi alat-alatan mudah alih di mana alatan tersebut boleh berhubung sesama sendiri tanpa wayar dan membuat kiriam daripada satu nod yang memulakan transaksi data kepada destinati nod yang dikehendakinya. Tanpa sebarang sistem pengurusan, nod-nod menggunakan laluan algoritma untuk mendapatkan hala yang sesuai bagi membuat data transaksi kepada nod penerima. Rangkaian ini biasanya digunakan di kawasan malapetaka,mesyuarat atau perjumpaan yang memerlukan sambungan antara alat-alatan komunikasi dengan segera di mana infrastruktur tidak disediakan sebelumn ini. Walaupun rangkaian ad hoc mudah diperuntukkan dan mobiliti amat berfaedah, rangkaian ini terdedah kepada gangguan udara dan bunyi serta perubahan lokasi nod mengubahkan topologi rangkaian tersebut pada bila-bila masa sahaja.

Dengan perubahan algoritma En-DSR yang baru, laluan penghalaan dapat dilaksanakan dengan pertimbangan Quality-of-Service (QoS) yang sewajarnya.Algoritma ini memilih laluan yang lebih stabil dengan membuat perhitungan kestabilan nod di setiap laluan yang ada sebelum membuat data pengiriman di dalam rangkaian topologi yang dinamik ini. Ini dapat membantu meningkatkan pretasi rangkaian di mana topologi yang dinamik ini boleh menjejaskan data transaksi yang sedia ada. Dengan pemilihan laluan yang lebih stabil, pengiriman semula dapat dielakkan dan sumber rangkaian dapat digunakan dengan lebih cekap. Penggunnaan informasi IPv6 "Flow Label" , sumber rangkaian dapat disimpan dan digunakan bagi memenuhi syarat-syarat yang diberi oleh pelbagai applikasi.

Dalam perbandingan Dynamic Source Routing (DSR), En-DSR memberi keputusan yang memuaskan. Walaupun saiz data paket informasi bertambah, satu kombinasi baru di

paket IP akan digunakan seperti satu aliran transaksi yang dapat mengurangkan beban dan mempermudahkan data informai pemprosessan. Oleh demikian, QoS dapat diperuntukkan kepada setiap permintaan applikasi and justeru itu, meningkatkan pretasi rangkaian tersebut.

ABSTRACT

Mobile ad hoc network comprises a collection of mobile devices such as laptops that connect to each other wirelessly and they mutually agreed to forward data packets on behalf of the mobile node that initiates transmission within the network. Without any centralized management system, mobile nodes use routing algorithm to perform the routing process from finding a route to sending data to their recipients. They are usually deployed in the area of catastrophe, meeting or conferences where impromptu connections are required and no centralized infrastructure is readily available. Although ad hoc connections and mobility are advantageous, mobile ad hoc network is exposed with wireless disruptions. Moreover, movement in within the topology changes the network topology unpredictably at unpredictable time.

As a newly enhanced algorithm, routing process in En-DSR will be carried out with Quality-of-Service (QoS) in the transmission. Over a highly dynamic ad hoc network, this enhanced algorithm selects route that is more stable by measuring the stability of each intervening nodes in the route. This effort made to improve the network performance where highly dynamic mobile nodes behave unpredictably over the time which will affect on-going transmission. With stable routes for transmission, retransmission is minimized and network resources can be efficiently used. By utilizing IPv6 Flow Label in the IP packet header, resource reservation can be done on different application requests. In comparison with conventional Dynamic Source Routing, En-DSR performed relatively well. Despite the increasing length of the source route header, a new combination of fields in IP header is used to label the stream of data transmission as flow that reduces the per packet overhead as well

as simplifying the header processing. Then, better QoS can be provided to the transmission and improve the network performance entirely.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FORMULAS

LIST OF ABBREVIATIONS

ABR……..	Associativity Based Routing
ABR……..	Available Bit Rate
AODV……..	Ad-Hoc On Demand Distance Vector Routing
AQOR……..	Ad Hoc QoS On-Demand Routing
ARP……..	Address Resolution Protocol
ARS……..	Automatic Route Shortening
AS…….	Adaptive Services
ASAP……	Adaptive Reservation and Pre-Allocation Protocol
ATM…….	Asynchronous Transfer Mode
BA……	Behavior Aggregates
BB…….	Bandwidth Broker
BE…….	Best-effort
BECN……..	Backward-explicit Congestion Notification
BL…….	Base Layer
BT…….	Burst Tolerance
BW……...	Bandwidth
CBR……..	Constant Bit Rate
CLP……...	Cell Los Priority
CLR……..	Clear packet
CONSER…..	Collaborative Simulation for Education and Research
DAG…….	Directed Acyclic Graph
DBCLH……	Delay and Bandwidth Constrained Least Hop
DiffServ.....	Differentiated Services
DSCP……	Differentiated Services Code Point
DSDV…….....	Destination Sequence Distance Vector
DSR …….	Dynamic Source Routing
ECN……..	Explicit Congestion Notification
EL…….	Enhancement Layer
ext-QOLSR..	Extension of QoS extended Optimized Link State Routing
FECN……	Forward-explicit Congestion Notification
HR……	Hard Reservation
ICMP……	Internet Control Message Protocol
IETF…….	Internet Engineering Task Force
IntServ….....	Integrated Services
IPDV……	IP Delay Variation
IPv4……..	Internet Protocol version 6
IPv6..….....	Internet Protocol version 4
LAN……..	Local Area Network
LSP……...	Label-Switched Path
LSR……...	Label-Switched Router
MANET……	Mobile Ad Hoc Network
MBS…….	Maximum Burst Size
MCR…….	Minimum Cell Rate
MIT……..	Massachusettes Institute of Technology
MPLS……..	Multiprotocol Label Switching
MPR…….	MultiPoint Relay

MRL……. Message Retransmission List
NL……. Node Lifetime
NPDU……... Network Protocol Data Unit
NRL…….. Normalized Routing Load
NS2……... Network Simulator 2
NSF…….. National Science Foundation
OLDW…..... On-Demand Link Weight
OLSR ……... Optimized Link State Routing
OSI……... Open Systems Interconnection
OWD…… One Way Delay
OWPL……... One Way Packet Loss
PARC…….. Xerox Palo Alto Research Center
PCR…….. Peak Cell Rate
PDF …….. Packet Delivery Fraction
PFA…….. Path-finding Algorithm
PHB…….. Per-hop Behaviors
QoS ..…..... Quality-of-Service
QRY……. Query packet
RREP…… Route Reply packet
RREQ……... Route Request packet
RRER…… Route Error packet
RSVP…..... Resource Reservation Protocol
RTI……... Real Time Intolerant
RTT…….. Real Time Tolerant
SCR…….. Sustainable Cell Rate
SMTP……... Simple Mail Transfer Protocol
SR……. Soft Reservation
TC …… Topology Control
TCP/IP..... Transmission Control Protocol / Internet Protocol
TOA…….. Type of Applications
TORA……... Temporally-Ordered Routing Algorithm
ToS …….. Type of Service
TTL……... Time To Live
UBR…….. Unspecified Bit Rate
UDP…….. User Datagram Protocol
UPD…….. Update packet
USC/ISI…… Information Sciences Institute
VBR…….. Real-time Variable Bit Rate
VC…… Virtual Circuit
VINT…… Virtual InterNetwork Testbed
VOD……. Video-On-Demand
VoIP……. Voice-over-IP
WAN…… Wide Area Network
WRP …..... Wireless Routing Protocol
WRR……. Weighted Round Robin
ZRP……... Zone Routing Protocol

PUBLICATIONS

W.Y.Tai, C.E.Tan and S.P.Lau, Enhance QoS Based Routing in IPv6 Ad Hoc Networks: Current Initiatives and Challenges, A Comparative Study, in Proceeding of Young ICT Researchers Colloquium, FCSIT, Universiti Malaysia Sarawak, Kota Samarahan, 12-13th May 2010

W.Y.Tai, C.E.Tan and S.P.Lau, Enhanced QoS Based Routing in Ad Hoc Network in Ad Hoc Networks, in Proceeding of 8th Asia- Pasific Symposium on Information and Telecommunication Technologies (APSITT 2010), Kuching, Sarawak, 15-18th June 2010.

W.Y.Tai, C.E.Tan and S.P.Lau, Improving IPv6 Wireless Ad Hoc Networks QoS via Enhanced Flow Label with Stability Based Dynamic Source Routing Scheme, in Proceeding of the 17th Asia-Pacific Conference on Communications (APCC 2011), Kota Kinabalu, Sabah, 2-5th October 2011

W.Y.Tai, C.E.Tan and S.P.Lau, Toward Utilizing Flow Label IPv6 in Implicit Source Routing for Dynamic Source Routing (DSR) in Wireless Ad Hoc Network, in Proceeding of 2012 IEEE Symposium on Computers & Informatics (ISCI 2012), Penang Malaysia 18-20th March 2012

CHAPTER 1

OVERVIEW

1.1 Introduction

Temporary connections are sometimes needed during emergency or conferences where preexisting infrastructure such as the wired connection is unavailable. In some isolated places, wired connections can hardly be available. With the absence of connection infrastructure, communication can be frustrating. Thus with mobile devices usage nowadays, ad hoc network can be created spontaneously without much technical skills required. Mobile devices can be connected to form an ad hoc network via the wireless connection. Transmissions can be carried out wirelessly and eventually ended without much hassle in comparison to wired connection. However, wireless ad hoc network is exposed with potential disruptions while transmission in progress due to the nature of wireless connection. Wireless connection encounters interruptions such that they may be halted due to the changes of topology. Transmissions that are on-going in within the network will be affected and subsequently lower down the ad hoc network performance.

Therefore, an adaptive routing scheme is presented here to provide a better network performance for the highly dynamic mobile ad hoc network. Mobile devices in the network will transmit data packets using a more reliable routing scheme where the transmission will have end-to-end delay, packet loss and so forth minimized in order to provide a better network performance for end users. Albeit the unpredictable movement among the devices that change the topology, this routing scheme will allocate a reliable link for the transmissions in order to render a good quality route for the transmission.

1.2 Background

Communications between people can be carried out ubiquitously. Internet connects to a great number of computers allowing almost all kinds of communication to be taken place anywhere and anytime. Introduction to numerous state-of-the-art technologies facilitates the communication needed by the people. In comparison to the early usage of Internet, typical email or data transfer is no longer the only traffic in the network. As propelled by the development of Internet and computer technologies, multimedia applications such as the Video-On-Demand (VOD) or Voice-over-IP (VoIP) make information sharing and other applications to be even more sophisticated. They are no longer bounded by distance with the introduction of different ways in communication. Besides, users are no longer limited to static way of data presentation in order to stay in touch or to interact with different applications.

Emergence of wireless mobile technologies has contributed to the higher demand of connections whereby such communications are conventionally supported by fixed infrastructure at the end. People are enjoying the wireless connection especially the mobility feature where they can move freely while staying connected. Mobile devices such as Smartphone are equipped with wireless capability to connect to the Internet. With only one single hop to the base station, mobile devices connect to the base station which is eventually linked to the wired infrastructure. They use the fixed infrastructure for connections whereas ad hoc network works differently. Without any preexisting infrastructure, ad hoc networks can be established instantaneously for the end users at many kinds of application environments and typically are deployed at conferences, catastrophic recovery areas and battle fields. Mobile devices in ad hoc network (henceforth referred to as node) can be statically located or dynamically moving at different directions as well as their speeds. Having

such a dynamic and unpredictable nature, changes in the network topology is inevitable. Even though mobile ad hoc networks provide temporary connections and it has to allow mobility to facilitate the needs of the end users to move from one place to another while staying connected. Therefore, maintaining contemporary connections become a challenge in wireless ad hoc network.

Wireless connections are exposed to numerous disruptions in the air that cause jitters, end-to-end delay, packet drops and so forth. Occurrence of these disruptions often brings down the whole network performance which reflects a not reliable network. Packet drops along the data transmission will deter a good experience for end users and this is not acceptable. Therefore, various efforts are made in order to create a good user experience and better Quality-of-Service (QoS) in mobile ad hoc network. QoS is defined generally as "totality of characteristic of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service" in ITU-T Recommendation (2008).It includes the capability to provide sufficient resources to different type of services in a variety of applications within the network. It will be able to guarantee services and acceptable performances to a certain level of acceptance by the end users rather than relying on the conventional "best-of-efforts" service.

Depletion of Internet Protocol version 4 (IPv4) addresses in the modern network has been addressed by the introduction of Internet Protocol version 6 (IPv6) where it can support up to 2^{128} addresses usage in the near future. The structure of IPv6 protocol has been reconstructed to supersede the previous version. It has not only provided more IP addresses for the future usage, it does include other important aspects such as QoS into the

consideration during the reconstruction of the IP addresses. In this era, many time sensitive and bandwidth hunger applications are introduced. They require good connections and acceptable QoS level in the network in order to carry out their transmission. Therefore, QoS requirement is emphasized. Owing to the dynamic nature of the network, QoS becomes a very crucial aspect.

1.3 Problem Statement

Acceleration and change of positions on random nodes during the transmission contribute to low network performance. Network topology changes at unpredictable time affects on-going transmissions which eventually cause retransmission due to the route failure. Eventually, the transmission suffers delays and low packet delivery. As there is no centralized management from any fixed infrastructure such as router or access points, the data transmissions are unmanaged and easily interrupted when disruptions are encountered. Therefore, routing in the highly dynamic mobile ad hoc network is not easy. While maintaining good network performance, routing across the ad hoc network requires data to route over a reliable path. In selecting the best path to route among those vigorously moving nodes in the wireless ad hoc network, QoS is crucial for the end users.

QoS is important for real time applications where they require delays and certain routing metrics to be at their minimum. “Best-of-service” effort has no guarantee for the data to be delivered or reach their destination with a certain level of assurance. In contrast, IPv6 has promised to facilitate the special handling requests for data using Flow Label in the IP packet header where it has yet to be standardized. With such a feature, an approach is proposed here to utilize the IPv6 Flow Label together with a route selection that prefers

reliable path in the highly dynamic mobile ad hoc network in order to provide a better QoS in the network.

1.4 Significance

With IPv6 QoS Provision, mobile ad hoc network will be provided with better resources reservation for all data transmission across a dynamic ad hoc network. Besides, route selection that prefers more stable route will minimize the retransmission in the ad hoc network as well as minimizing the impact on network performance whenever route failure is encountered. Despite the dynamic nature of the ad hoc network, QoS can be applied to different services on the data transmissions by satisfying different needs that are requested by the applications respectively. Network performance will be improved despite the wireless disruptions. Packet drop will be reduced while packet delivery will be increased.

1.5 Objectives

Objectives for this research are as follow

- To propose an adaptive route selection scheme that would route data over a more reliable path in the dynamic nature ad hoc network.
    - To reduce packet drop probability with the new adaptive route selection scheme over the transmission in the ad hoc network
    - To allocate sufficient resources to maximize percentage of packet delivery
- To provide Quality-of-Service (QoS) on different application requests and improve network performance in the ever-changing ad hoc network
- To evaluate the network performance by comparisons over the other routing schemes with the proposed routing scheme

1.6 Methodology

Research methodology is done by reviewing literature of other similar cases where it relates to achieve better QoS in high mobility mobile ad hoc network. Performance evaluation on previous proposed schemes is done to identify the pros and cons in the schemes proposed. After analyzing the pros and cons, new scheme is proposed to address the shortcoming and problems arise from the ad hoc network data transmission. Simulation is done to model the proposed scheme with random mobility and different traffic patterns in comparison to previous proposed scheme.

1.7 Scope

The aim of this research is to analyze the various routing schemes that route across the dynamic mobile ad hoc network and propose a new scheme that would provide better QoS in transmission and better network performance. IPv6 ad hoc networks are simulated in this research. This research will run the proposed scheme and other routing schemes to be compared under four different total mobile nodes in high mobility ad hoc network where the wireless ad hoc networks consists of static and mobile nodes at different rate and different speeds of moving mobile nodes. Connections are established at random time and data are transmitted accordingly. IPv6 Flow label will be used to convey the information for resource reservation based on requests on the source node. Route selection in the proposed scheme will use stability values will be applied with assumptions on all node will update their timestamps whenever movement is detected. Besides, a comparison will be carried on a network where all nodes are mobile at different time and speeds. There will be no static node located in the wireless ad hoc network transmissions in order to simulate a random stability value for dynamic ad hoc network.

1.8 Thesis Overview

1.8.1 Chapter 1: Overview

In this chapter, the problem statement and significance of the research are presented. In addition, objectives, methodology and the scope are also addressed accordingly.

1.8.2 Chapter 2: Literature Review

This chapter examined ad hoc routing protocols that are proposed by researchers and architectures that include Quality-of-Service (QoS). Besides, IPv6 packet headers are presented with proposals of Flow Label utilization suggested by researchers over the time. Furthermore, ad hoc routing protocols that include QoS into routing mechanism are addressed as well.

1.8.3 Chapter 3:Enhanced Dynamic Source Routing (En-DSR) Implementation

In this chapter an adaptive routing protocol is proposed. Together with an analysis of the proposed enhanced Dynamic Source Routing (En-DSR) protocol, it is presented with Flow Label field utilization in IPv6 packet header. Route selection over the routing mechanism is introduced.

1.8.4 Chapter 4:Implementation Results and Analysis

This chapter covers an introduction of Network Simulator 2 along with parameters for the simulations to be carried out. Simulations are described and results are discussed as well analyzed with comparisons made.

1.8.5 Chapter 5: Conclusion and Future Work

Final chapter presents a conclusion of the research and summarizes the simulation results. Future area of work and areas of improvement are also discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Changes in topology often disrupt undergoing transmission in the highly dynamic mobile ad hoc network. Addition of new nodes allow more possible routes to perform better in time while the departure of existing nodes might break discovered routes for transmission. All these transitions happen at unpredictable time which eventually allows more transmission to be reestablished. Without a centralized management, wireless ad hoc routing depends on the routing protocol on forwarding data. A number of routing protocols are introduced to bring in QoS into ad hoc networks for better transmission. Different categories of routing protocols such as proactive and reactive routing protocols that are commonly used in wireless ad hoc networks are studied in this chapter to understand their natures and mechanisms. Besides, QoS architectures such as Multiprotocol Label Switching (MPLS), Frame Relay, Asynchronous Transfer Mode (ATM), Internet Protocol (IP), Differentiated Services (DiffServ) and Integrated Services (IntServ) are proposed to improve the service conditions in their respective areas. In this study, it is to analyze all advantages and disadvantages of routing protocol which is useful in helping to create a solution to address the current shortcomings in the wireless ad hoc networks with QoS requirement implied.

2.2 Mobile Ad Hoc Networks (MANETs) Routing

Mobile ad hoc network is defined as a network that is composed of devices where they connect to each other in within the wireless transmission range(Sarangapani, 2007). Mobile devices in ad hoc network may communicate in a single hop or multi-hop paths in order to communicate with each other. They do not need any existing infrastructure as they can self-configure the network wirelessly. Since they are mobile devices, mobility has to be taken into consideration where it changes the topology dynamically. Each of the mobile devices (henceforth referred to as nodes) respectively acts as the host and router altogether as they mutually agreed to forward data packets among themselves wirelessly. Node can be statically located at one point or moves at different direction where speed is as well varied. Consequently changes in the ad hoc network topology are inevitable. Without pre-existing infrastructure they can be created spontaneously unlike wired networks, therefore MANET is not a type of centralized wireless network (Toh, 2002)where it can be a standalone network or externally connected to other networks.

MANET deployment is commonly used in different environments where fixed infrastructure is not supported. Temporary connections are deployed without requiring much specialized technical skill. It allows nodes to be connected rapidly unlike wired connections which require time to be invested in establishing and dismantling the connections (Sarangapani, 2007). Catastrophic recovery areas, battle fields or even conferences are often in absence of proper wired connections where the problem can be solved by substituting it with the ad hoc networks. Nodes can be connected among each other wirelessly and in order to initiate, a transmission routing path must first be identified.

2.2.1 Routing Protocols

Routing protocol specifies on how nodes in the network communicate among each other based on the routing algorithm. Routing algorithm selects the best route that allows communication to be constructed between sources to their destination with the knowledge of the network topology. In gathering information of connecting nodes in route within the wireless ad hoc network, Layer 3 of the Open Systems Interconnections (OSI) model is made used of. There are generally two major categories of routing protocol in ad hoc network which are proactive and reactive routing protocols.

2.2.1.1 Proactive (Table-Driven) Routing Protocols

Proactive also known as Table-Driven routing protocols keep tables of information regarding their neighbors' details which resembles the wired network (Liu & Kaiser, 2005). Proactive protocols learn about the changes in topology along the time by sending out topological information to their neighbors. Each node in the network will have their respective tables containing the latest information of the routes to nearby nodes. Information about the network topology is updated frequently and therefore, immediate establishment of transmission can be carried out without much delay. However, in order to maintain an up-to-date table it will be required a compensation of overhead in the routing table. Higher bandwidth consumption will happen if the changes of the topology are frequent. Among the proactive routing protocols are Destination Sequence Distance Vector (DSDV), Optimized Link State Routing (OLSR) and many more.

2.2.1.1.1 Destination Sequence Distance Vector (DSDV)

DSDV is one of the table-driven routing protocols(Perkins & Bhagwat, 1994)based on Bellman-Ford routing mechanism where it computes shortest paths from the source in a weighted digraph where the main contribution of DSDV is said to solve the routing loop problem. In this DSDV study, it improved the distance vector approach in solving the aforementioned problem. It has a routing table for each respective mobile node where each of them will keep a list of all available destinations and the number of hops. Besides, each entry in the table will be tagged with a sequence number originated by the destination node. Information of updates will be disseminated by all nodes periodically with increment on the sequence number. Latest sequence number in a route is preferred in selecting the best path whereas with the same sequence number the smallest metric will be selected. In addition, even sequence numbers are used generally to indicate the presence of a link while odd is used otherwise. In order to alleviate the possibility of large amount of updates to be done over a highly dynamic network, there are two types of packets introduced. Firstly, full dump carries all available routing information that requires multiple network protocol data units (NPDUs) whereas smaller incremental packets is for updating purposes when the network topology changes slightly over a period of time. Nodes keep track of the installed time of each route in the table to identify the stale routes. Besides, a settling time table will be used to prevent fluctuations of all route entries into the table by advertisement (Perkins & Bhagwat, 1994). Fluctuations occur whenever two routes with identical sequence numbers are received in a wrong order and therefore, by delaying the broadcast of routing updates traffic in the network will be minimized.

In DSDV, the updates are relatively small where it is not sending entire routing table over the network while maintaining loop-free routes for transmissions. However in this simple routing protocol, it is inefficient as it requires periodically updates on the transmission regardless how frequent topology changes. It increases the usage of bandwidth in promoting the updates across the network especially when full dump occurs.

2.2.1.1.2 Wireless Routing Protocol (WRP)

WRP is one of the well-known proactive routing protocols that allows node to communicate the distance as well as the second-to-last hop for each respective destination. In WRP, all nodes are required to have four tables in storing all information about their neighbors in the mobile ad hoc network and they will use the path-finding algorithm (PFA) where it avoids routing loops and counting-to-infinity problems (Murthy & Garcia-Luna-Aceves, 1996). Node will exchange updates among themselves periodically especially when link changes. An update message contains a list of updates that included the destination, distance as well as a list of which nodes should acknowledge the update. During the link breakage, neighbors will modify their distance table entries and check for alternatives. For all new paths, it is relayed back for updates. Each node has:-

- Distance Table – stores a matrix that holds distance of each destination node of their neighbor, e.g. node *b* via each neighbor *c* of the respective node and the predecessor of *c*.
- Routing Table – stores the destination node address, distance to the destination, and predecessor of the chosen shortest path to the destination as well as the successor of the shortest path to destination. Besides, a tag marker is used to label the entry in the

table to mark them if they are simple path, loop or a destination that has not been marked.

- Link Cost Table – contains link's cost between each neighbor of the node where they keep track of the number of update periods from the time last since they receive a successful update from the particular neighbor.
- Message Retransmission List (MRL) – keep track of which neighbors that have yet acknowledge the update message and if needed update message will be retransmitted.

With all the information in the network, consistency check is often carried out to check their predecessor information as reported by the neighbors. It helps to achieve loop-free routing and perform faster convergence in the dynamic ad hoc networks. However, sufficient memory requirement is needed to maintain the four routing tables where periodic updates are required in WRP.

2.2.1.1.3 Optimized Link State Routing (OLSR)

OLSR is an optimization of the classical link state algorithm where it uses Hello and Topology Control (TC) messages to retrieve information from the network. Being a proactive routing protocol, it includes multipoint relays (MPRs) concept by using extraction algorithm (Liu & Kaiser, 2005)where MPRs are selected among the node in the ad hoc network (Clausen & Jacquet, 2003). It performs a hop-by-hop routing where each node in the network uses the most recent topology information to compute the next hop destination using shortest hop in paths. Hello messages are used to find one and two hops neighbor through their responses. With that, MPRs can be selected where they are the only selected nodes that can flood the network with TC messages that contain the neighbor information. It reduces the

overhead as compared to the traditional flooding process where every single node is broadcasting message. MPRs will declare the links among the MPR selectors and by connecting to the nodes a subset of links are created among the MPRs and the selectors. Hence, it is a simple routing protocol where routes are constantly available with updates in the ad hoc network. However, OLSR is not entirely loop-free and periodic updates often increase the bandwidth usage.

2.2.1.2 Reactive (On-Demand) Routing Protocols

Reactive also known as on-demand routing protocols will only initiate the task to search for possible routes from the source node to the destination node when there is a demand. This approach is also known as source-initiated on-demand routing where it creates route only desired by the originator (Royer & Toh, 1999). Transmissions in these protocols often require more time in the route finding. Initially, route discovery will be initiated to search for potential routes for the particular transmission and upon successful establishment the route will be maintained by a procedure until the end of transmission or when transmission is terminated due to link failures. There are a few protocols that are categorized as the reactive routing which are Ad-hoc On Demand Distance Vector Routing (AODV), Dynamic Source Routing (DSR) and Temporally-Ordered Routing Algorithm (TORA).

2.2.1.2.1 Ad-hoc On Demand Distance Vector Routing (AODV)

AODV algorithm is one pure on-demand routing protocol in route acquisition that invokes route discovery when only there's a request for transmission in the mobile ad hoc network(Perkins & Royer, 1999). It allows nodes in the network to obtain routes via dissemination of route request (RREQ) messages. RREQ will be rebroadcasted by the nodes between the source and destination if the receiving node is not the destination requested,

otherwise Route Reply (RREP) will be sent back to the source node. Intervening nodes will keep track of the information in within RREQ for the reverse path of RREP which eventually setting up the route to destination from the source. Besides, AODV uses routing table to store information about the next hop towards destination and a sequence number to indicate the freshness of the information. Destination sequence numbers are used to ensure all routes are loop-free and contain the most recent information on the route itself.

Movements of nodes often cause link breakages. Hence, whenever movements are detected a special RREP will be sent to the affected source nodes(Perkins & Royer, 1999). The special RREP uses fresh sequence number with hop count of ∞ will be propagated towards upstream of the break and by relaying the intervening nodes will be updated about the link breakages. Source node can start the route discovery once again for alternative route to the destination. However, AODV may lead to high overhead in RREQ packets transmission within the network and usage of unnecessary bandwidth for the information updating. Besides, the route between source and destination nodes is expected to be symmetrical(Perkins, Belding-Royer, & Das, 2003).

2.2.1.2.2 Dynamic Source Routing (DSR)

DSR is one of the source-initiated routing protocols that use source routing in seeking potential routes for transmission. It sends out Route Request (RREQ) as shown in Figure 2.1 only when a transmission is initiated by a node in the mobile ad hoc network.

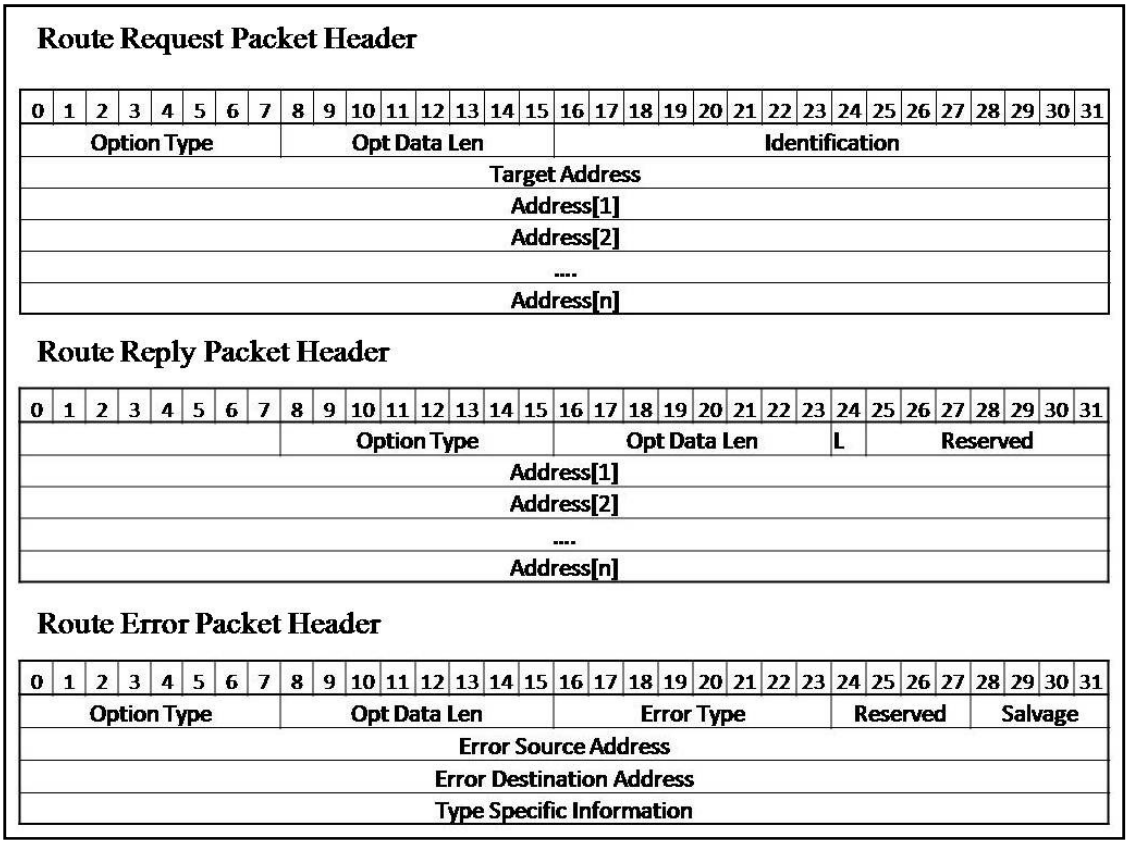

Figure 2.1: Conventional DSR Route Request, Route Reply and Route Error packets (Hu, Johnson, & Maltz, 2001)

Unlike AODV, RREQ in DSR is used to obtain a list of nodes in between the source and destination nodes. Upon receiving RREQ, destination node will send a RREP otherwise if it's not the destination node will add its own address by piggybacking onto the RREQ before forwarding it to its neighbor nodes. Eventually, the source will have a list of nodes as route. Possible routes are stored in a route cache. Whenever there's a transmission initiated, the route cache will be consulted first before triggering route discovery. Figure 2.2 shows the DSR protocol flow chart for both the route discovery and route maintenance mechanism. Being an on-demand routing protocol, it avoids the need for the network to get updated information on all the occasions for the whole duration (Johnson & Maltz, 1996) saving bandwidth as well as power consumption.

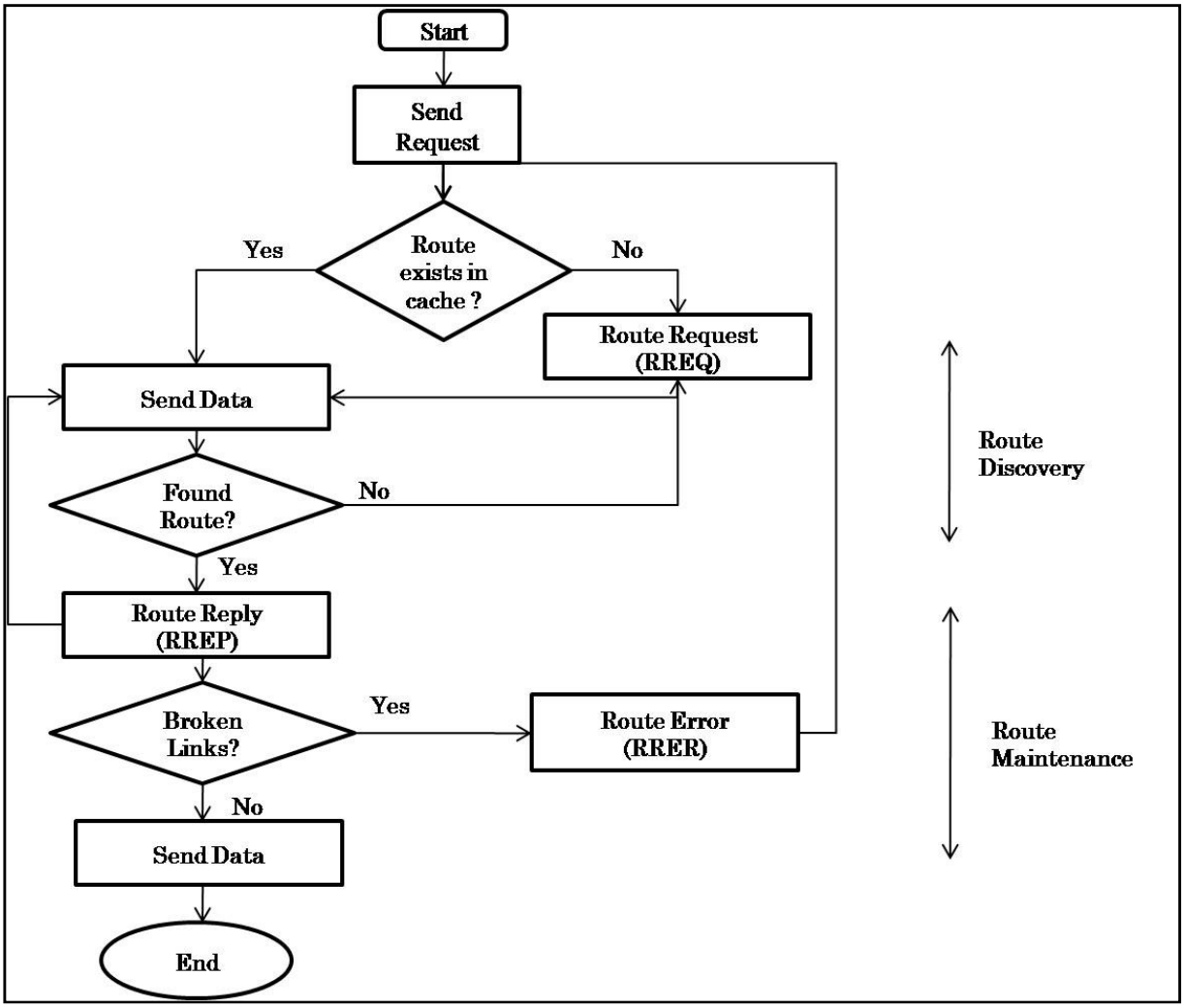

Figure 2.2: Flow Chart of DSR Protocol

DSR has a few optimizations where it helps to provide a better performance in the mobile ad hoc network. In the route discovery, nodes can overhear any packets and add the information in the RREQ into their respective route cache. A promiscuous mode is introduced in avoiding the RREP storm where all the nodes would have replied to the same RREQ from their cache. With this promiscuous mode, node will wait for a period of time for any source route that is better or equally good route and may then cancel its RREP to the source. While in the route maintenance phase, packet salvaging will be carried out whenever a transmission is interrupted. Node that encounters link break will send Route Error message back to the source and attempt to send the data using another route found in the route cache.

Besides, Automatic Route Shortening (ARS) shorten route if the one or more nodes in between is no longer needed. Node will examine the unexpended portion of the source route where the node is not the actual next-hop node but is named in the later portion of the source route can infer that the unexpended portion of the source route can be taken off. Participating node that can be reduced should return a "gratuitious" RREP packet to the original sender of the packet and concatenation of the portion of the original route can be done whenever the node may.

However, main disadvantage of the DSR is the increased per-packet overhead due to the piggybacked source route header. Implicit Source Routing for DSR is proposed by Johnson and Hu (2001) where in most of its simulations it showed that it preserves the DSR advantages while avoiding the lengthy overhead in each of the packet. Conceptually, Implicit Source Routing uses a flow identifier to identify the transmission per flow with a Flow Table which is also known as the Flow-State extensions to DSR. It minimizes the length of the source route header once the transmission is established. Flow Table consists of the next hop address to which the packet is to be forwarded to together with the combination of source address, destination address and flow identifier. Besides, it holds the previous-hop node's address along the flow, a timeout after which the record in the table must be deleted. Moreover, it consists of an expected value of the Hop Count field in the header for packets received for forwarding along this field (Maltz, Johnson, & Hu, 2004) .Initially, the source establishes a flow by sending out flow establishment packet with two headers which one is for the flow identifier whereas the other is the source route and a timeout for the flow. Subsequent data will be routed with the implicit source routing by adding a flow identifier header in lieu of a source route for the rest of the transmission. For all intermediate nodes, if the Flow Table consists of a route it will forward the data accordingly otherwise it will send a

FLOW UNKNOWN error back to the source. Upon receiving the FLOW UNKNOWN error, the flow will have to be reestablished.

2.2.1.2.3 Temporally-Ordered Routing Algorithm (TORA)

Temporally-Ordered Routing Algorithm (TORA) uses a "physical or logical clock" approach to create "temporal order" of topological change events that is used to construct the algorithm's reaction during the changes(Park & Corson, 1997). TORA's required an assumption of synchronized clock which is accomplished by external source such as Global Positioning System (Cordeiro & Agrawal, 2006). TORA is also a source-initiated and loop-free distributed routing protocol that based on link reversal concept. It includes a formation of directed acyclic graph (DAG) rooted at the destination node during the route creation and maintenance phase. During a route discovery, Query (QRY) packets and Update (UPD) packets are required where both consists of destination id (did) and UPD stores the height of the nodes. Height is a term introduced by TORA that defines the distance of the responding node's distance to their respective destination. Using the height, source node will select the best route toward destination and not always the shortest path. It requires all nodes to be synchronized with the same timing as the height is depended on the logical time of link failure. TORA has a route erasure phase where it will flood with Clear (CLR) packets throughout the network in erasing the invalid routes when nodes detect network partitions. Links are assigned with direction of upstream or downstream based on the relative height metric of the neighboring nodes. If the topology changes causing the routes break, the node will take the opposite streams to route the data as in link reversed to adapt to the changes. In addition, TORA supports multiple routes in the transmission.

2.2.1.3 Others Routing Protocols

Other than the two major categories routing protocol that are discussed as above, there are a few more categories routing protocols that are introduced along the years. There is Link Stability Based routing protocol that find routes which consider the links' stability in the networks (Liu & Kaiser, 2005). Besides there is this Signal Stability Routing Protocol (SSR)(R.Dube, C.D.Rais, K.Wang, & S.K.Tripathi, 1997) which uses signal strength and location stability as the metric of evaluating a link. Routes with stronger signal strength are favored but required to maintain two routing tables for signal strength records. Signals are obtained by the periodic beacons sent in the network. In other hand, Associativity Based Routing (ABR) Protocol uses the degree of association stability to predict the lifetime of the link itself (Toh, 1997). Periodic beacons are used to notify their neighboring nodes of their connection stability which is known as associativity tick. Higher associativity tick count reflects more stable connections in the network.

Apart from stability, Zone Routing Protocol (ZRP) is a hybrid routing protocol where it divides the network into routing zones according to the distance between the nodes(Zygmunt, 1997). Different approaches are used to different zones in the network where they proposed to reduce the overhead in this proactive approach but as the zones overlap heavily, it incurs more overhead.

2.3 Quality-of-Service (QoS) Initiatives

Internet with Internet Protocol version 4 (IPv4) adopted the "best-of-effort" services where all data transmission will be treated equally and sends with best effort. No guarantees are made on the packet delivery. Multimedia and real time application require higher priority in transmission as compared to the non-real time applications request where traffic in the network is mixed with different types of data packets. High figure of jitters, end-to-end delay, packet delivery, packet loss and others are encountered in the traditional way of transmission. Demands over a better quality or treatment on the packets are overwhelming especially in real time applications such as video conferencing, VoIP, and so forth where they have low tolerance over delays in the network. Therefore, QoS services are required by these applications in assuring certain level of services provided in their transmission from one end to the other.

QoS has been used in many different fields in the network in effort to provide a better service to the end users hence, QoS is broadly defined. There is a definition where QoS is defined as a measurement on how a network will provide preferential treatments to certain different types of traffic (Ferguson & Huston, 1998). However, QoS is defined as a set of service requirements to be met by the network when transmitting data from the source to the respective destination in the (Crawley, Nair, Rajagopalan, & Sandick, 1998) standard. Regardless, QoS requirements are much needed to enhance the performance of the network.

2.3.1 Current architectures that support QoS over the Internet

QoS provides certain level of assurance to the transmission in the network especially to those much needed real time applications. Large amounts of digitized voice as well as video data have been populating the Internet daily. End users demand to have acceptable performance in their transmissions where delay and packet loss are minimized. Some of the QoS architectures are discussed as followed.

2.3.1.1 Frame Relay

Over the years, Frame Relay is introduced for the Wide Area Networks (WANs) in the attempts to provide a better performance in the packet technology. It has better integrations among the LAN applications with the same physical network and more reliable networks at a lower cost (Buckwalter, 1999). It grows popularity as the cost of connection is low over long distances. Working at the physical and data link layer, Frame Relay requires end users to be connected to the local area networks (LANs) as well as the public WANs to a frame relay node where they perform encapsulations on the data packets before transporting over to the end destination. Virtual links are created in this network where the size of data packets varies and data packets are commonly known as “frames”. Error corrections are to be done at each of the end points allowing a faster transmission over the networks. End users can select a level of service quality with this network where some packets are being prioritized over the rest. In order to achieve simplicity and reduce network overhead, Frame Relay omits link-by-link flow control and implements Forward-explicit congestion notification (FECN) and Backward-explicit congestion notification (BECN). However, during network congestions where some of the services are overwhelmed with requests and it leads to difficulty in ensuring QoS for all

transmissions. In Frame relay, the inconsistent size of data packet causes difficulty in providing guarantee QoS.

2.3.1.2 Asynchronous Transfer Mode (ATM)

Asynchronous Transfer Mode is a technique used in network technology which uses the asynchronous time-division multiplexing to encode different types of data into fixed-size-packets known as cells. It has been defined as a transfer mode that organized video, voice and data into 53 octet cells (Dunn & Martin, 2000). Virtual circuits (VCs) have to be constructed from two ends before the transmission is to take place similarly to Frame Relay architecture. In ATM, quality assurance is provided by involving traffic contract in the connection. In order to measure traffic characteristic, a range of traffic parameters such as Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) , Maximum Burst Size (MBS) , Burst Tolerance (BT) and Minimum Cell Rate (MCR)(Kasera, 2006). Besides, there are a number of service categories defined in ATM for different applications' requests in the connection but four commonly known are listed as below:-

- Constant Bit Rate (CBR) – a Peak Cell Rate (PCR) is specified, a constant number
- Real-time Variable Bit Rate (VBR) –an average cell rate that is being specified
- Available Bit Rate (ABR) – a guaranteed rate that is specified at minimum for congesting traffic that has real and non-real time variants
- Unspecified Bit Rate (UBR) – remaining transmission capacity that has been allocated to the traffic

ATM involves traffic management where it manages the traffic within the network in order to provide certain level of satisfactory services as well as optimizing the network

resources. Traffic shaping is implemented at the entry point of the ATM network while traffic policing is deployed to maintain the network performance. Actions such as dropping the cells or mark the cells as discardable known as Cell Loss Priority (CLP) bit will be taken if the VC exceeds the traffic contract. Besides, another four more traffic management elements introduced are Traffic Contract Management for admission control, Priority Control that utilizes the cell header bits, Flow Control and Congestion Control.

2.3.1.3 Transmission Control Protocol/ Internet Protocol (TCP/IP)

TCP/IP has been introduced as a set of communication protocols that allow computers to communicate with each other over the Internet. Comprised of two significant layers, Transmission Control Protocol (TCP) is the primary transport layer in the OSI model while Internet Protocol (IP) is the primary network layer. This communication protocols are made up of a four-layer architecture that provides a framework for various protocols. They are Link Layer, Internet Layer, Transport Layer and the Application Layer. Figure 2.3(Chaudhuri, 2009) shows the Internet Protocol Suites with specific layers labeled. Link Layer deals with the technologies communicated in the local network in the link where Internet layer plays its role in facilitating the interconnections between the nodes as routing of data packets take place in this Internet Layer. Transport layer provides an end-to-end communication service to all applications (R.Branden, 1989)using the TCP with a reliable flow of data or User Datagram Protocol (UDP) that uses a simpler service that guarantees nothing on the packets delivery to their respective destination. Lastly, Application layer handles all type of applications that reach the specific layer in the protocol. It has all implementations on vastly defined of TCP/IP applications in the data communication services.

Internet Protocol Suite
Application Layers
E.g. – Simple Mail Transfer Protocol (SMTP) , File Transfer Protocol (FTP), Telnet, etc
Transport Layer
E.g. – Transmission Control Protocol (TCP), User Datagram Protocol (UDP), etc
Network Layer
E.g. – Internet Protocol (IPv4, IPv6), Internet Control Message Protocol (ICMP), etc
Data Link Layer
E.g. – Address Resolution Protocol (ARP), Ethernet, Frame Relay etc

Figure 2.3: Internet Protocol Suite

2.3.1.3.1 Internet Protocol version 4 (IPv4)

Initially, IPv4 which operates at the Internet layer in the TCP/IP stack uses an eight bits field named Type of Service (ToS) in the IP packet header in providing better QoS in the network. However, it has been redefined by (Nichols, Blake, Baker, & Black, 1998) for Differentiated Services (DiffServ) and known as Differentiated Services Code Point (DSCP). Originally as ToS field, the field is used to specify parameters such as precedence, delay, throughput and reliability as shown as Figure 2.4. It has the precedence to show the importance of the diagram received. Delay bit is to convey a prompt delivery is need while throughput shows high data rate is needed for this datagram. Reliability demands a higher level of effort on this delivery. However, as aforementioned the eight bits field is redefined into six-bit Differentiated Services Code Point (DSCP) field and a two-bit Explicit Congestion Notification field. Apart from utilizing the IPv4 ToS field for QoS request, in (Partridge, 1992)defined a data structure which is called as the flow specification for the host to negotiate type of service from the internet. In this flow specification it is used to indicate the service requirements for a single direction flow. Hence, QoS will be provided at layer three in the internetwork. Depletion of IP addresses in the current network usage is alleviated by the

introduction of IPv6. Migration from version 4 to version 6 is implemented where IPv6 promises efficiency and QoS provisions. In section 2.4, IPv6 will be explained in details to show how IPv6 could provide QoS provisions to the transmission the network.

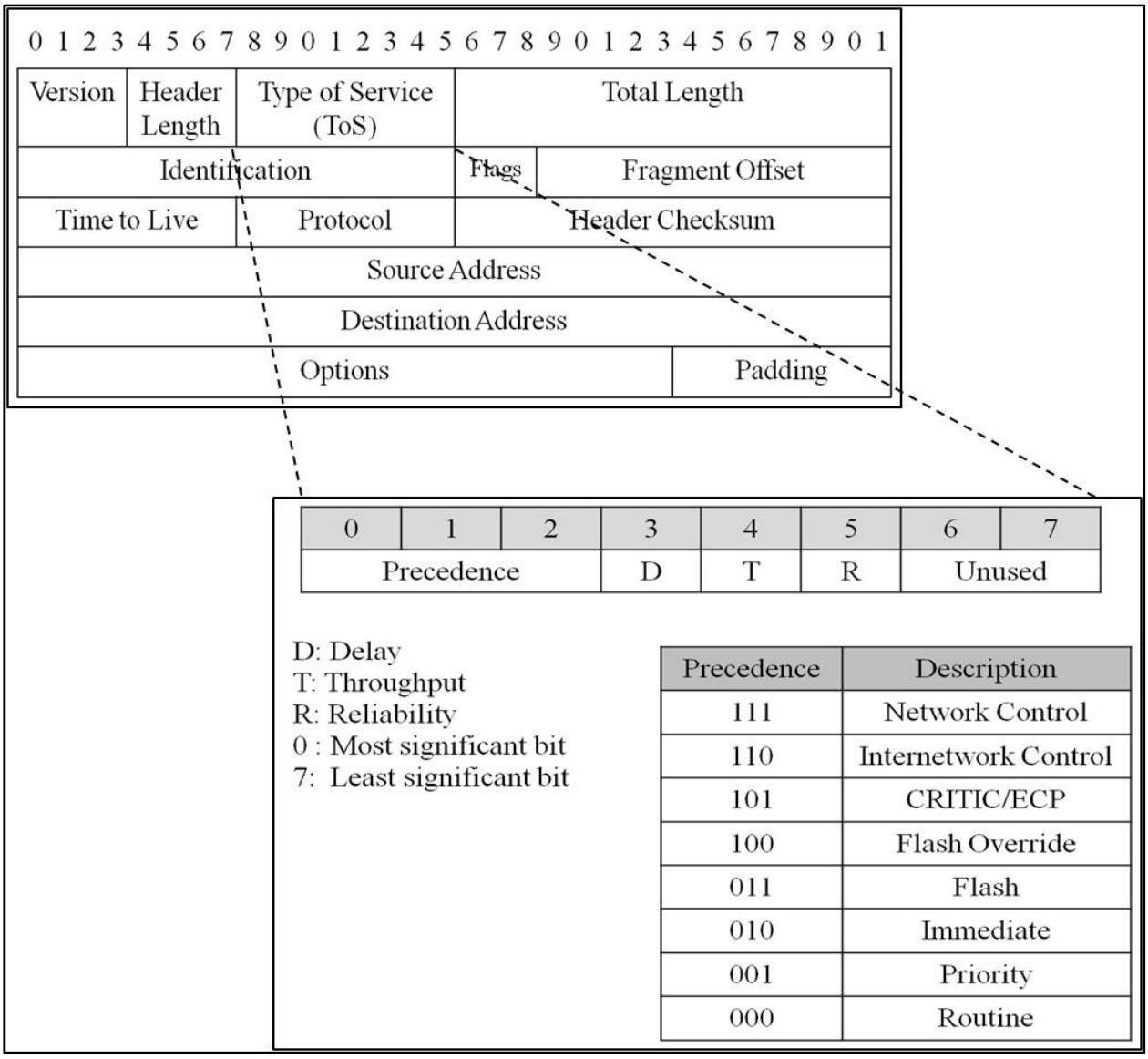

Figure 2.4: Type of Service (ToS) field utilizations in RFC 791 (1981)

2.3.1.4 Integrated Services (IntServ)

Internet under IPv4 implements best-effort delivery on all type of packets sent by end users despite insufficient assurance in delivery. Such service is however unacceptable when it comes to real time applications' transmissions where they require higher QoS requirements and certain assurance on delivery as compared to the non-real-time applications. Integrated Services (IntServ) is then proposed in 1994 to alleviate this problem as an extension to the Internet architecture. This extension is to provide integrated services as in a flow based where link-sharing is controlled with resource reservation and admission control implied explicitly in the Internet (Braden, Clark, & Shenker, 1994). Integrated Services (IS) term included three different services which are the Best Effort Services, Guaranteed Services and Controlled Load Services. In realizing the integrated services, reference implementation framework is shown in Figure 2.5. It is constructed with four components listed as below:-

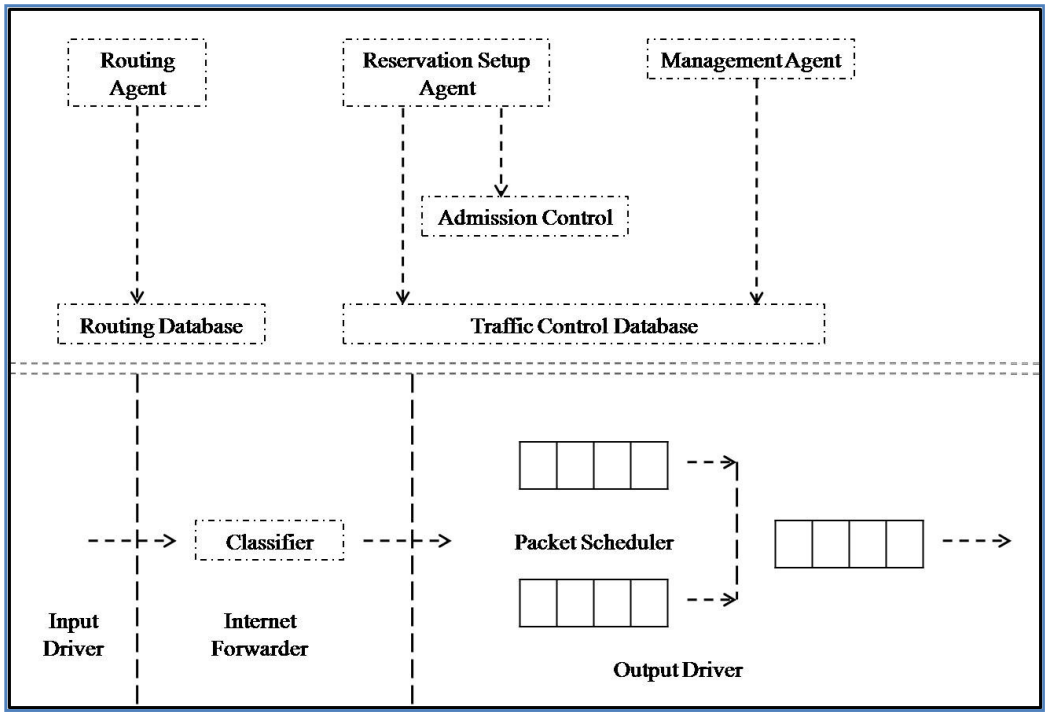

Figure 2.5: Implementation Reference Model for Routers in Integrated Services (IntServ) illustrated in (Braden, Clark, & Shenker, 1994)

IntServ provides per-flow QoS guarantees to individual application sessions where they defined a few classes of services needed by various applications. Applications are to request QoS at their respective routers along their path where it utilizes RSVP (resource reservation protocol) to conduct a two-way handshake in establishing a flow connections with certain type of service applied. Flows in this IntServ are defined as distinguishable streams of datagram that required same QoS over the transmission. In implementing flow based QoS, routers must be able to create different qualities of service where this function is known as traffic control. Packet Scheduler, classifier and admission control as shown in Figure 2.5 are components in traffic control. As defined by Braden, Clark and Shenker (1994) overview, Packet Scheduler manages the packet streams with a set of queues or other mechanisms. In the other hand, classifier is used to map all incoming packets into their distinctive classes. Admission control works out the decision algorithm on creating flows with the requested QoS granted without causing confliction any other flows. Lastly, RSVP will create and reserve

resources based on flow-specific state request in between the endpoints along the flow. Source reservations require high overhead on routers(Braden, Clark, & Shenker, 1994). Even though IntServ is limited by the scalability of the vast Internet, IntServ that employs flow based in wireless ad hoc network is stated to perform equally well.

2.3.1.5 Resource reSerVation Protocol (RSVP)

RSVP is used to reserve resources in the Internet in boasting the QoS for end users where they applied IntServ together in the mechanism. It is defined as several data objects are required to carry resource reservation information with the Controlled-Load and Guaranteed QoS control services (Wroclawski, 1997)for the purpose. They are three RSVP protocol objects known as FLOWSPEC, ADSPEC and SENDER_TSPEC. Mentioned in Wroclawski(1997), FLOWSPEC object carries reservation request information generated by the receivers while ADSPEC object carries information generated by the sources or intermediate network elements such as availability of specific QoS control services and parameters of the required QoS. Lastly, SENDER_TSPEC carries the same traffic specification generated by the source throughout the RSVP.

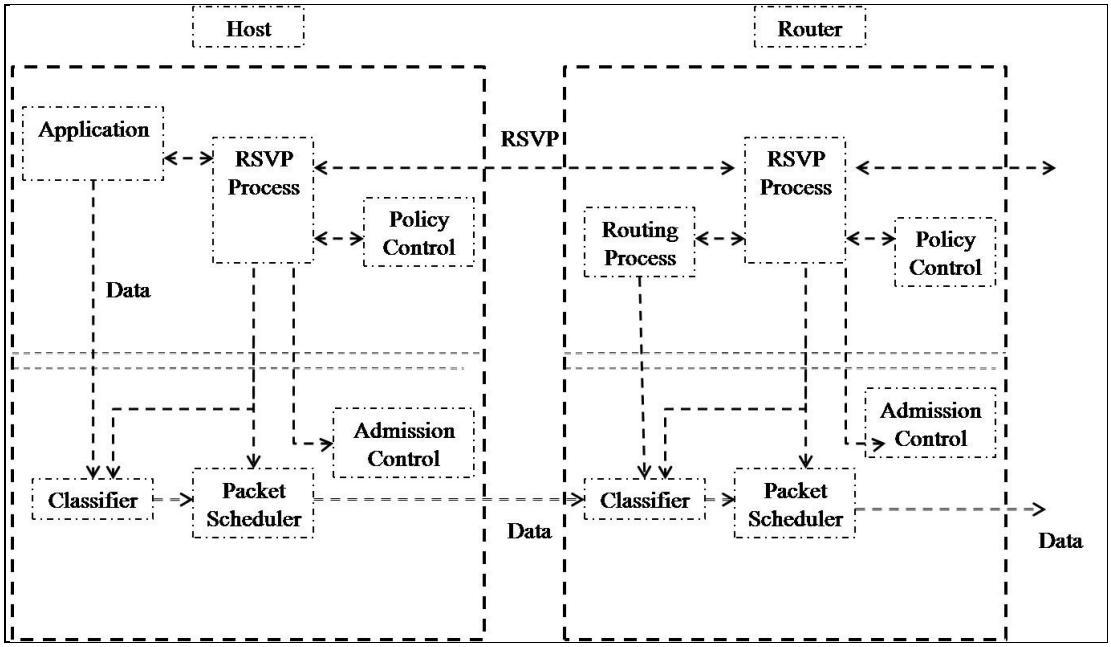

Figure 2.6: RSVP between host and routers as illustrated in (Braden, Zhang, Berson, Herzog, & Jamin, 1997).

QoS is supplied collectively on a particular data flow by mechanism known as "traffic control" where it has packet classifier, admission control and packet scheduler. As shown in Figure 2.6, Packet classifier determines the QoS class where the packets belong to while the packet scheduler will provide the promised QoS to the flows. Similarly defined in Integrated Services (IntServ), traffic control with the request information supplied will trigger RSVP to reserve the resource in the routers for the whole transmission and implement the QoS service model. Moreover it will maintain the requested state of service together with IntServ architecture. Hence, QoS is ensured for all transmission requested. It is not a routing protocol but it works with routing protocols altogether in preserving resources in the network.

2.3.1.6 Differentiated Services (DiffServ)

"Service" is defined by Blake et.al (1998) as the significant characteristic implied on the packet transmission within a network across all paths. In the Internet, service differentiation on variety of applications is needed by the end users as the requirements for transmission vary among them and QoS requirements are required. In this scalable Internet, DiffServ is introduced with an architecture that is constructed in a way to aggregate the traffic classification state with differentiation by utilizing the IP-layer packet marking in the Differentiated Services Code Point (DSCP) in the IP packet header as discussed in Blake et.al(1998). DSCP utilization in the IP packet header is illustrated in Figure 2.7. In contrast to IntServ, DiffServ is a class-based mechanism for traffic management while IntServ is flow-based. Out of 8 bits, 6 bits are used as DSCP value while the last 2 bits are used as code points in Congestion Notification in case of congestion in an overloaded router. It is similar to Frame Relay's usage of Backwards and Forwards Explicit Congestion Notification.

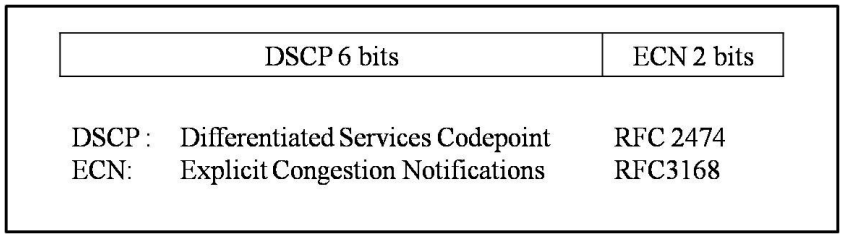

Figure 2.7: Differentiated Services Code Point (DSCP) format from Hagen (2006)

It marks DSCP in all the data packets with different classifications in the traffic by applying Per-Hop Behaviors (PHB). Packet that has the same DSCP will be grouped as the Behavior Aggregates (BA) and treated with the same resources. Each of the router on the network will be able to differentiate the traffic according to the respective class while edge routers will aggregate the traffic accordingly where buffers will be allocated as well as the bandwidth resources at different nodes into their respective flows of the traffic while core

routers in the network will forward the packets based on the DSCP. A traffic conditioner exists to perform traffic conditionings based on different requirements and it has the right to remark the traffic flows or discard the particular characteristic in the streams as long as it complies with the traffic requirements.

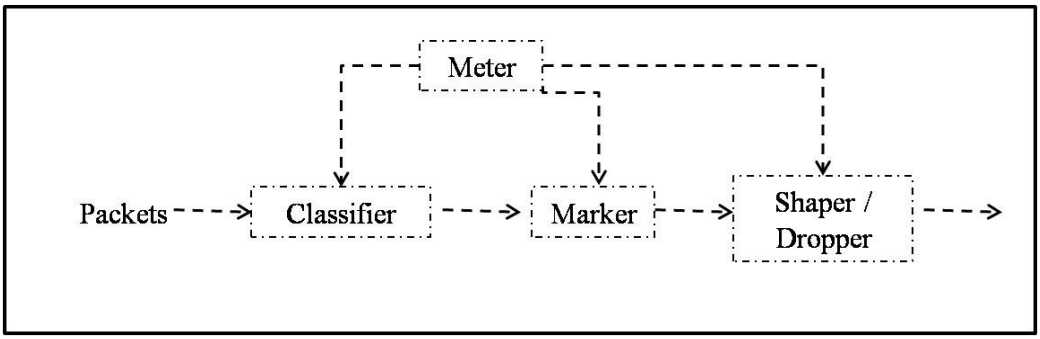

Figure 2.8: Logical View of the Traffic Conditioner and Packet Classifier as illustrated in RFC 2475.

Illustrated in Figure 2.8 logically, Traffic Conditioner has Meter where it holds the temporal properties of the flow of packets such as rate and burst size (henceforth referred to traffic profile) selected by the classifier and uses them to trigger a particular action that are suitable to each packet passed through. Markers are used to set the DSCP and adding the packet to their respective BA. Shapers manage the traffic streams by delaying the stream of packets while Droppers discard packets in order to ensure the traffic profile is compromised.

Bandwidth broker (BB) is then introduced by Nichols, Jasobson and Zhang (1999)into the framework of DiffServ where it acts as an agent that can be configured with the organizational policies to keep track of all current allocated resources to the marked traffic in the network. Control on the resources can be done independently with some knowledge on the organization. Agents can be used to allocate bandwidth for end-to-end connections with far less state rather than deploying them per-flow based. However, in this mechanism it is hard to ensure the QoS provided over both ends on different networks are similar as policies implemented varied despite being able to classify different needs in all transmissions.

2.3.1.7 Multiprotocol Label Switching (MPLS)

MPLS is introduced to be working between Layer 2 and Layer 3 where it works well with different protocols used in the network. It supports many different protocols such as Asynchronous Transfer Mode (ATM) and Frame Relay (Rosen, Viswanathan, & Callon, 2001). This is done by the mechanism of MPLS where it uses label to the data packets and create virtual links in between the nodes accordingly. Working alongside with IP, all packets are required to have an MPLS header where it contains information for the routers to gain knowledge about its transmission. This emphasized that all routers are required to understand the MPLS mechanism. In the header, MPLS labeled packets contains 4 components which are the 20-bits label value, 3 bits from the Traffic class field and Explicit Congestion Notification (ECN), 1 bit bottom of stack flag and 8 bits of time to live (TTL) fields. Packets that come into the Edge Label-Switched Router will be assigned a label and forwarded to Label-Switched Path (LSP) where forwarding decision are based solely on label at each Label-Switched Router (LSR) as discussed by Alwayn (2002).

2.3.1.8 Summary

Based on the reviews on different QoS architecture made, there are advantages and disadvantages on different architectures. As summary in pointing out the QoS provisions that have greater potential in improving QoS to the transmission classification on type of traffics helps to prioritize the data transmission are one of the possible ways such as proposed in DiffServ. In addition, resource reservations as proposed in RSVP helps in providing higher QoS to higher prioritized data packets to route over the network. These have prompted better initiatives to be taken in further utilizing them in routing over high dynamic wireless ad hoc network.

2.3.2 Ad Hoc Routing Protocols with QoS mechanism

Wireless ad hoc networks are transmitting data wirelessly where all transmissions are exposed to the environment disruptions. Thus, QoS is much needed to provide special treatments on real time applications where they need lower packet drops and end-to-end delay. In (Crawley, Nair, Rajagopalan, & Sandick, 1998), QoS-based routing is known as the routing mechanism which manages the flows based on the knowledge of the resource availability in the network as well as the QoS requirements of the flows. Therefore, routing mechanism should be able to allocate sufficient QoS and make available to those applications in the network. As reviewed beforehand, some QoS architectures are included in the routing mechanism in improving the performance. Below are some of the mobile ad hoc routing protocols that support QoS in transmitting data.

2.3.2.1 QoS for Ad Hoc Networking Based on Multiple Metrics: Bandwidth and Delay

Using Optimized Link-State Routing (OLSR) protocol, QoS routing scheme is proposed by enhancing OLSR, known as QOLSR (Badis, Munaretto, Agha, & Pujolle, 2003). Traditional OLSR uses hop distance as the significant metric for routing selection but in the QOLSR it proposed more appropriate metrics which are delay and bandwidth measurements as the constraints. The generic QOLSR proposed algorithm is presented in Figure 2.9. Bandwidth is measured by IEEE 802.11b as the medium access control by calculating the bandwidth between node and its neighbor that has direct and symmetric link. While delay is measured by introducing Hello message into the neighbor discovery phase with a creation time stamped in it. Messages received by nodes will use the time stamped in the message and calculating the difference between the time and current time received will be the “measured_delay”. With more information for the routing protocol, QOLSR manages the resource efficiently but it

does require more computations and it does not differentiate real-time and non-real-time services of IP in the ad hoc networks.

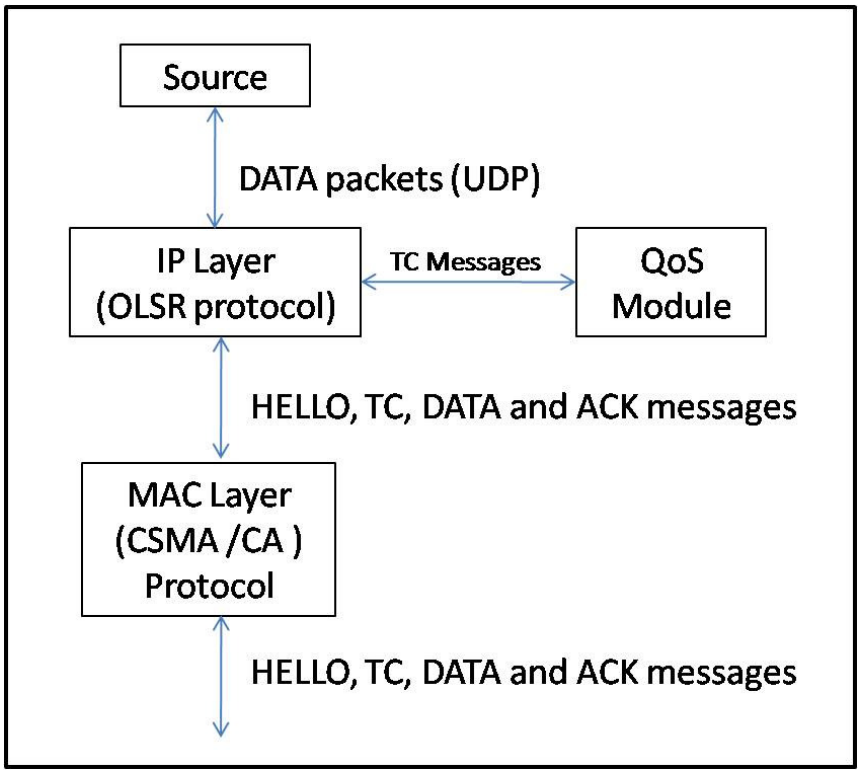

Figure 2.9: Generic QOLSR scheme

2.3.2.2 On-Demand Link Weight (OLDW)

Targeted for real-time-multimedia applications, ODLW proposed a routing protocol that takes QoS parameters such as bandwidth, delay and node lifetime (NL) based on the battery as constraints. OLDW is an on-demand routing protocol that selects the best path based on the link weight parameters as mentioned (Al-Khwildi, Loo, & Al-Raweshidy, 2007). It uses route request (RREQM) message to acquire the required bandwidth, minimum available route bandwidth, accumulated route delay and minimum route lifetime from node to node in attempt of constructing a QoS route list. Destination chooses the best optimum path that meets the requirements even if the route is longer as long as they fit the requirements. When the selection of route is made, route reply (RREPM) will be sent back to the first available

intermediate route back to the source node with all intermediate nodes update their respective table as depicted below. Given routes that are available fulfilled the requirements of the transmission, the one with the longest node lifetime will be selected which in this case the smaller number of NL will be preferred.

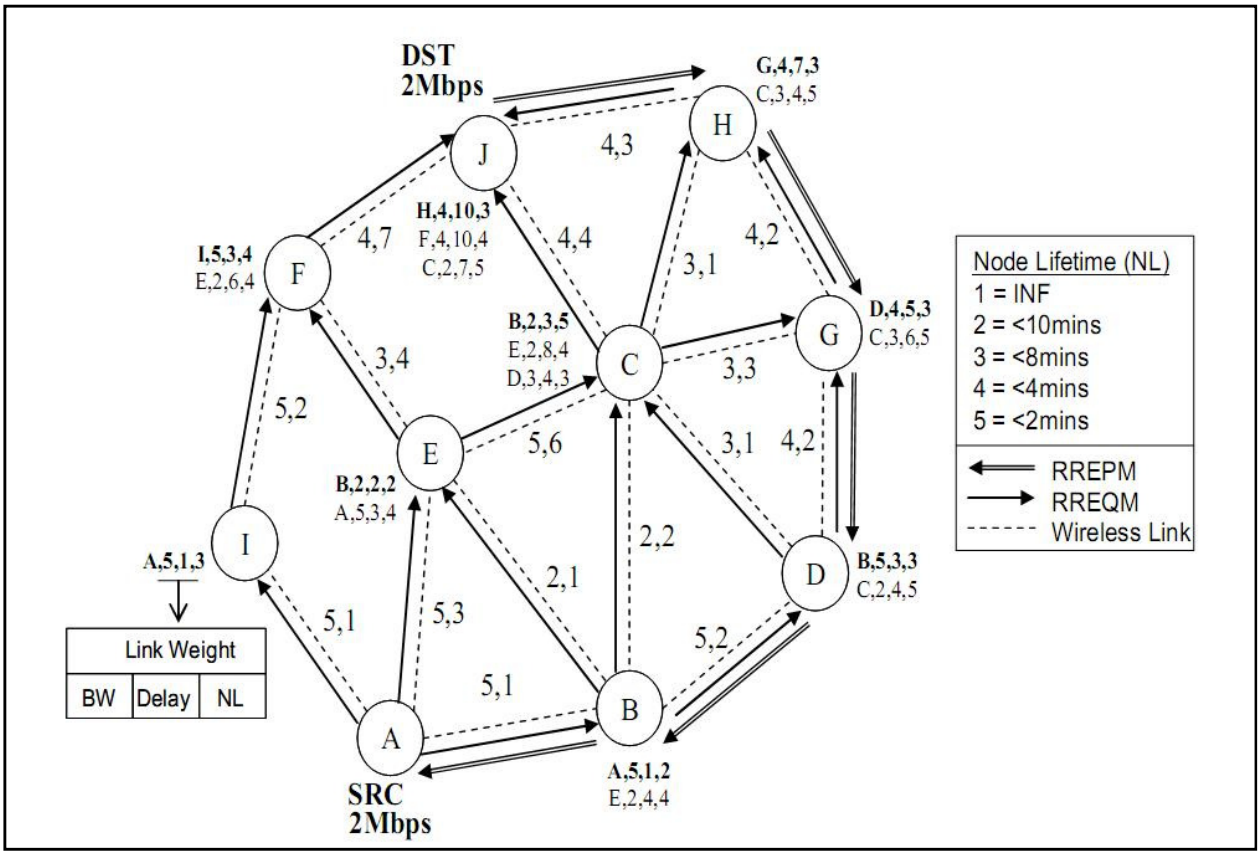

Figure 2.10: ODLW Routing Mechanism
taken from (Al-Khwildi, Loo, & Al-Raweshidy, 2007)

Figure 2.10 as shown illustrated a routing mechanism using ODLW routing protocol. Node A is introduced as source where it sends out RREQM with QoS parameters to its neighboring nodes while Node J as the destination node. Computation is required in this algorithm where bandwidth and delay are calculated as a link weight together with cross layer interfaces to gather information Physical and Network layers. As it is taking accounts on the required bandwidth, delay and other parameters, it shows QoS provision made in this routing mechanism. However, node lifetime that is determined by batteries focuses on the node

availability. There are situations where the node lifetime is sufficient but changes the route because of node mobility. This protocol employs route maintenance mechanism to handle the changes of route.

2.3.2.3 Ad Hoc QoS On-Demand Routing (AQOR)

In supporting QoS in MANET, AQOR introduces a resource reservation-based routing and signaling algorithm by proposing a detailed computation that estimates the available bandwidth as well as the end-to-end delay (Xue & Ganz, 2003). QoS provisions introduced in this includes the temporary reservation of resources and destination initiated recovery processes. As an on-demand routing protocol, AQOR uses hop-by-hop routing and signaling function for resource reservation and maintenance. Similar to INSIGNIA, AQOR adopted in-band signaling approach in reducing signaling overhead and uses similar RSVP mechanism in resource reservation part. As type of traffics are not differentiated between flows, QoS violation can take place in which AQOR uses route recovery in avoiding it. Routing overhead that are caused by QoS violations will be alleviated by destination-initiated recovery which increases the route sequence number and broadcast the unsolicited route update packet from the destination to the source. Neighbors' information is sent out periodically in traffic management, QoS violation detection and recovery which increase the overhead in the overall network. Besides, route request packets are meant to travel along all intermediate nodes between the source and destination nodes. Once route request packets are received, destination node will initiate a route reply to be sent back to the respective sender. Upon receiving the route reply, route registration mechanism will take place to mark the route status in preserving the bandwidth for the imminent transmission.

2.3.2.4 QoS-Aware Dynamic Source Routing using DiffServ Principles

Cross-layer solution is proposed in this QoS-aware DSR routing protocol using DiffServ principles where bandwidth allocation is done based on the QoS levels. Bandwidth reservation is done during route discovery similarly to the conventional Dynamic Source Routing (DSR) however, QoS levels of services and DSCP are used by every node in the network where source node is an edge router and the rest will be the core routers during the operation(Bhavanam, Potluri, Yeruva, Masupha, & Negi, 2008).

In admission of a real time flow, the source will check their cache for available route otherwise it will initiate the route discovery and reserve the bandwidth for the path. Source node will handle the admission control with the minimum bandwidth carried in the broadcasted route request (QRREQ) packets. Destination will send back route reply (QRREP) packet by reversing the path if the minimum bandwidth is satisfied. If the admission control fails, the *ac* bit will be set and source receives the QRREP will reinitiate the route discovery again. Once route is found, the source will mark the data packets with corresponding DSCP and all packets will be treated according to their respective policy. Remarking with the different drop precedence will be based on the profile and queued accordingly. For queue mechanism, Weighted Round Robin (WRR) is used. This routing protocol requires all nodes to accept the DSCP marking mechanism on the IP packets in classifying them respectively and acknowledge the QoS levels of service implemented in the network.

2.3.2.5 INSIGNIA

INSIGNIA is known as an in-band signaling system for mobile ad hoc networks where they support QoS by putting the control information into IP packets along with the data(Ahn, Cambell, Lee, & Zhang, 1999). In contrast to RSVP that uses out-of-band approaches, it is said to be more suitable to use INSIGNIA to provide QoS in highly dynamic environments where in-band signaling is included with flows reservation, and adaptive algorithms for both ends. Placing control information in the IP option, flows in the network are required to be closely monitored with highly responsive mechanism in order to re-establish the reservation made to the respective flows. This requires high computation and complexity where the reservation is divided into two services. They are the adaptive services (AS) and the best-effort (BE) packet based on their request as stated in the IP option field. Figure 2.11 shows how INSIGNIA utilizes the field in the IP option in conveying information for QoS provision. Real-time flows with AS bandwidth request will have bandwidth reserved for the flow and will be degraded to BE if the flow is rerouted during insufficient resources in the path. BE does not required any resource reservation to be made in the network by default while AS bandwidth request allows flows to have maximum (MAX) and minimum (MIN) bandwidth requirements (Lee & Campbell, 1998). Payloads are indicated according to the type of packet being transported. They are base (BL) and enhancement layers (EL) where the type of packet can be assured their admission control and resource reservation in the flows. MAX is required to support both type of packet while minimum is required only for BL.

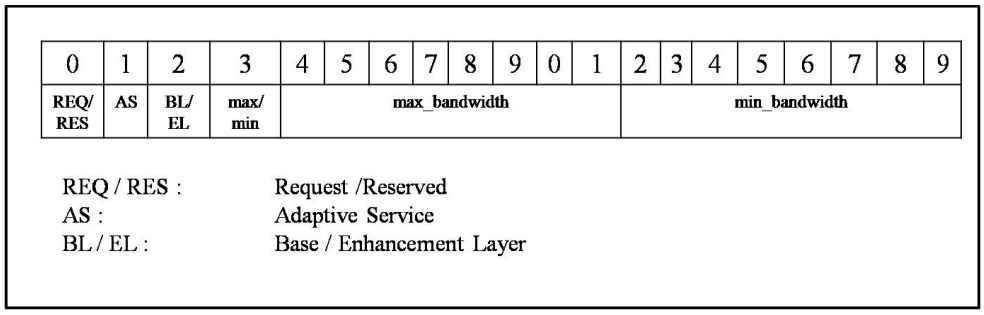

Figure 2.11: INSIGNIA Packet Requesting MAX/MIN reservation
from (Ahn, Cambell, Lee, & Zhang, 1999)

INSIGNIA keeps flow state information within the nodes of a certain path where it leads to scalability problem. Before transmission has taken place via a particular route, the extra reservation on the path itself is an inefficient way of utilizing the bandwidth even if the reservation is not for long. Besides, more defined levels of service are much needed for numerous type of applications' requirement (Studi, 2003).

2.3.2.6 Adaptive Reservation and Pre-allocation Protocol (ASAP)

ASAP, an adaptive QoS protocol for mobile ad hoc network (MANET) proposed two signaling messages in providing QoS to MANETs. It is said to provide QoS support with flexible adaptation in the ad hoc network and to minimize wasted reservations caused in the network changes. Similar to INSIGNIA, ASAP uses signaling system with a two-phase reservation mechanism where soft and hard reservation can be made (Xue, Stuedi, & Alonso, 2003). Soft reservation is considered as a claim of certain flows for a certain amount of bandwidth to be used in the coming future while hard reservation is to allow exclusive reservation on bandwidth for the flows. Signaling is being done by two messages known as SR (soft-reservation) messages and HR (hard-reservation)messages (McTasney, Grunwald, & Sicker, 2009) where SRs are in-band signaling periodically inserted to setup and repair paths in the flow and collect QoS information whereas HR messages are out-of-band signaling sent

in opposite direction of the flow to update the reservation state. Besides, it does as well as sending the QoS adaption information back to the sender. Bandwidth is reserved according to the request as soft-reserved or hard-reserved is required. Combination of both soft and hard reservation is proposed to avoid the waste of extra reservations in host other than the bottleneck where it is an effort made to attend problem occurs in INSIGNIA. Even with a good classification between soft and hard reservation, ASAP faces substantial end-to-end delay when it comes to flow restoration in case of a broken link. Both end nodes detect missing reservation only when soft reservation is refreshed. Moreover, missing reservation detected on one end of the transmission cannot be notified to the other end in when there is another path that provides actual end-to-end reservation.

2.4 IPv6 Quality-of-Service (QoS)

With IPv4 used in the current network deployment, it does only provide best of effort delivery and required other QoS architectures to be included in supporting QoS requirement. As IPv4 addresses deplete, migration to IPv6 is taking place. Internet Protocol version 6 (IPv6) is introduced to succeed the previous IPv4 that has 32 bits of IP address. As explained in the RFC 2460(Deering & Hinden, 1998), it is conceived by the Internet Engineering Task Force (IETF) to address the problems that arise from the IPv4 implementation. As the demands of Internet grow exponentially, Internet is expected to cope with the overwhelming demand. However, IPv4 address exhaustion has taken its toll thus IPv6 is developed to alleviate it. Besides alleviation of IP addresses, IPv6 is expected to fulfill and enhance the current state of Internet in aspects such as QoS requirements, security features, and many more. IPv6 major changes are:-

1. Expanded Addressing Capabilities

   From the size of 32 bits, IP address is increased to 128 bits in order to support more levels of addressing hierarchy with a much greater number of addressable nodes and a simpler auto-configuration of address. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses respectively and a new type of address known as "anycast address" is defined where it is to send a packet to any one of a group of nodes.

2. Header Format Simplification

   Some of the IPv4 header fields are dropped or made optional in IPv6 header in order to reduce the common-case processing cost of packet handling as well as to limit the bandwidth cost of the IPv6 header.

3. Improvement on support for Extensions and Options

IP header options are changed to allow more efficient forwarding, less stringent limits on the length of option and a greater flexibility for introducing new options in the future.

4. Flow Labeling Capability

A new 20 bits field is added to enable labeling on packets belonging to particular traffic "flows" for which the sender requests special handling such as non-default quality of service or "real-time" service.

5. Authentication and Privacy Capabilities

Extensions to support authentication, data integrity and (optional) data confidentiality are specified for IPv6 are added.

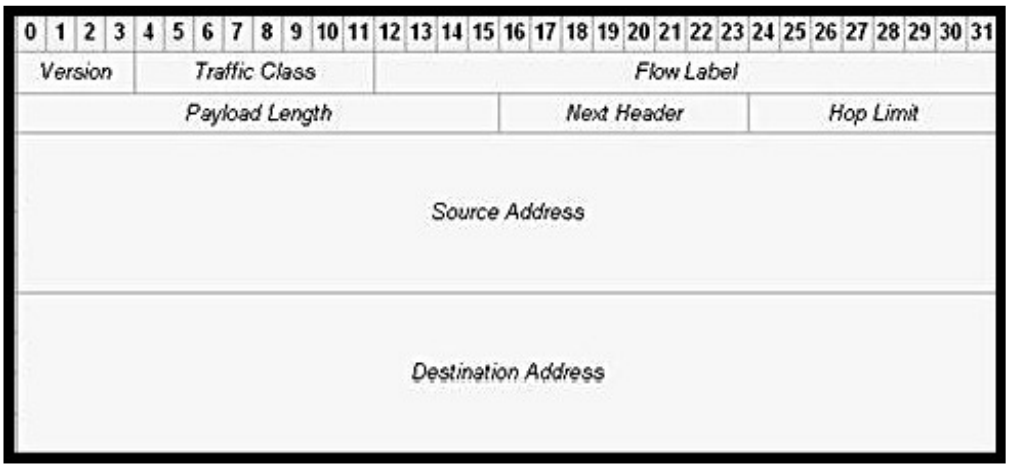

Figure 2.12: IPv6 packet header format

IPv6 header format as shown in Figure 2.12 consists of a 4 bits version field which is used to label the Internet Protocol version number, 6. Next to version field is Traffic class which is 8 bits field used for nodes to distinguish the different classes of priorities in IPv6 packets. It is intended to function similarly to the IPv4 Type of Service (ToS) field in the IPv6

version (Deering & Hinden, 1998). In implementing DiffServ into the IPv6 network, Traffic class field is used as illustrated in Figure 2.13:-

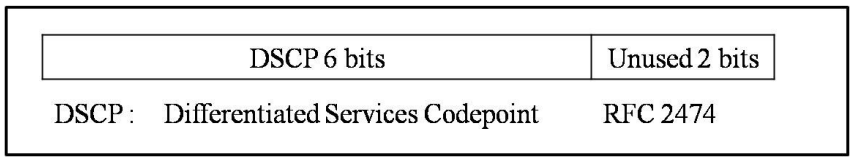

Figure 2.13: DiffServ field structure as illustrated in (Nichols, Blake, Baker, & Black, 1998)

Flow Label as one of the fields in previous Figure 2.12 is newly added into the IP header format while payload length is 16-bit unsigned integer to indicate the length of IPv6 payload. Next header is an 8-bit selector where it identifies the type of header following immediately after using the same values as IPv4 Protocol field. Hop limit is superseding the Time-to-Live field in IPv4 where it will decrease by one in each node that forward the IPv6 packet and will be discarded if Hop Limit is equal to zero. Source address and destination address are both 128-bits to address the originator and recipient of the packet.

2.4.1 IPv6 Flow Label

Latest migration from IPv4 to IPv6 in the Internet, IPv6 packet format is constructed efficiently to increase the network performance. It has fewer fields as compared to IPv4 and the arrangement of the fields is taken into consideration for a speedier processing. Besides, extension header mechanism will provide support for QoS demanded in the Internet and "Time to Live" field in IPv4 is renamed to "Hop Limit" in IPv6 where it does not enforce the maximum packet lifetime. Instead of stating the limit period of time that the data stays for, Hop limit decreases by one and will be discarded when it reaches zero(Deering & Hinden, 1998). In addition, a special 20-bits field is introduced in the IPv6 header to provide special handlings to the IPv6 routers known as Flow Label.

2.4.2 Flow Label Definition

A sequence of packets sent from a source to a particular unicast, anycast or multicast destination that the source labels are known as a flow (Rajahalme, Conta, Carpenter, & Deering, 2004). Flow classifiers have been using 5-tuple of the source and destination addresses, ports and transport protocol type traditionally but Flow Label field is subsequently introduced in IPv6.It consists of 20 bits in the IPv6 header located right next to Traffic Class field to provide special handling required by the intervening routers in the nature of flows. It requires only 3-tuple of Flow Label, source and destination addresses now to enable efficient IPv6 flow classification. Flow label is assigned to a flow by the flow's source node where it must be chosen (pseudo-)randomly and uniformly from the hexadecimal range of 1 to FFFFF(Deering & Hinden, 1998). All the packets in the same flow will share the same 3-tuple values. Using the 3-tuple in the header, IPv6 routers will be able to classify and assign the packets based of the flow-state as per QoS requirements set for the specific flow. The value in the flow label set not to remain unchanged until it reaches the destination node(s). Lifetime of the flow is 120 seconds and hence nodes that keep dynamic flow state must never assume packets arriving then or more after the previous packet of a flow to belong to the same flow unless it is explicitly refreshed the lifetime of the flow (Rajahalme, Conta, Carpenter, & Deering, 2004). Besides, the usage of it doesn't necessarily signal any requirement of the packet reordering and if any IPv6 node that is not participating the flow specific treatment process they must ignore the Flow Label when receiving or forwarding the packets in transmission.

2.4.3 Flow Label Provisions

Several ways in utilizing IPv6 Flow Label have been made along the years as the requirements for flow support in the Internet changes. Various proposals are made to the IETF to define their usage in the header where they consider various aspects. Proposals are made and reviewed accordingly.

2.4.3.1 A proposal for IPv6 Flow Label Specification "Draft-conta-ipv6-flow-label-02.txt"

In this proposal proposed by Conta and Carpenter(2001), IPv6 Flow Label is said to be able to alleviate the problem in DiffServ usage as it requires 5 or 6 element tuples to work and the problem arises when the packets have extension headers. Besides, it proposed a change in the Flow Label specification to allow changes in the value set in the field by intervening nodes with condition that its purposes and originality are preserved or restored whenever necessary. This suggestion allows intervening nodes to convey information between the nodes themselves and such flexibility does include some complexity in the processing in the field. These proposals of utilizing the Flow Label are illustrated in Figure 2.14.

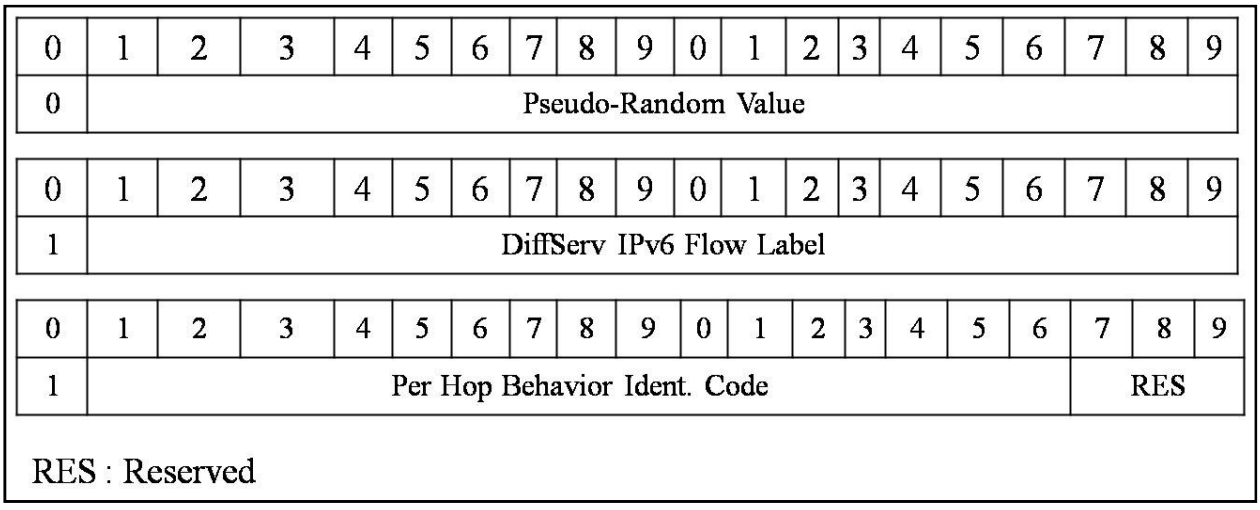

Figure 2.14: IPv6 Flow Label format and the DiffServ IPv6 Flow Label Format suggested

Random number method preserves the compatibility with the design mentioned in (Deering & Hinden, 1998) but it does not contain much information for transmission. In the DiffServ IPv6 Flow Label, the field is used by a number contrasted based on the Differentiated Services "Per Hop Behavior Identification Code" (PHB ID). It is either derived directly from the DSCP or from the "IANA Assigned Value" that occupied 16 bit in the Flow Label field. 3 bits are reserved for future use. This approach provides efficient processing in forwarding the data packets but it requires the nodes to ensure the correctness of the Flow Label where they might require more computation processes. There are more suggestions proposed in the draft for the flow label usage.

a) Server Port Format – Short Format

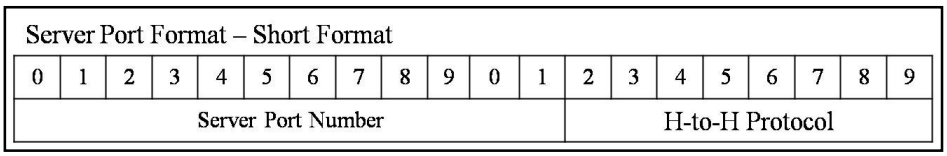

Figure 2.15: Server Port Format – Short format

"Server Port Number" is the port number assigned to the server side in the client/server application for identification on the particular application as well as the type of it which is a good indication of the type of QoS characteristics needed. Field utilization is shown as Figure 2.15. H-to-H Protocol values in the field is the Host-to-host Protocol such as TCP, UDP and so forth. This format makes aggregation with IPv4 and IPv6 classification rules possible but as more transmissions initiated from the same application that is communicating on the same end nodes this format is not able to differentiate the instances.

b) Server Port Format – Long Format

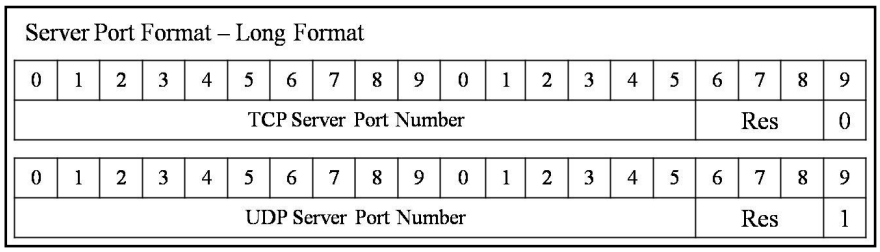

Figure 2.16: Server Port Format – Long Format

In Figure 2.16, long format is used to represent the TCP or UDP port number assigned to the server side of the client/server application since TCP and UDP are observed to be the two major host-to-host protocols. 3 bits are reserved for future use and the last bit in the field is used to represent TCP port (by 0) and UDP port (by 1).

Long format requires no new classification rule but it has the same problem of differentiating the instances similarly to short format. Besides, confining the labels to either TCP or UDP will be limiting the QoS capability as there are more than the two widely used protocols.

c) Header Length Format

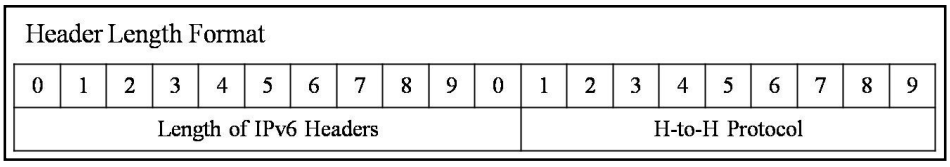

Figure 2.17: Header Length Format

Header length format is proposed to use the first 16 bits in the flow label to represent the length of IPv6 headers which is adding main headers and the extension headers preceding the host-to-host or transport header. It is illustrated in Figure 2.17 where it is to provide the information for DiffServ QoS mechanism to locate and retrieve the source and destination port along with the source and destination address as well as the host-to-host protocol from

the flow label to match the source and destination address, source and destination ports and the protocol identifier elements of a DiffServ Multi-Field Classifier. However, in this format extra computation is needed to calculate the length of the IPv6 header.

2.4.3.2 A modified Specification for use of the IPv6 Flow Label for providing an efficient QoS using Hybrid Approach from "draft-banerjee-flowlabel-ipv6-qos-03.txt"

This draft proposed by Banerjee, Malhotra and Mahaveer (2002) proposed a hybrid way to utilize the Flow Label in IPv6 header for providing efficient Quality of Service. Besides, it includes options for IntServ and DiffServ to be provided with IPv6 QoS supports. In this hybrid approach, the first three bits of the IPv6 Flow Label are used to define the type of approaches used while the rest are used to define the format used according to the approach. Table 2.1 showed the ways of usage for the first three bit in the IPv6 Flow Label in this proposal.

Table 2.1: First three bits usage in the IPv6 Flow Label

Bits	Description
000	Default.
001	A random number is used to define the Flow Label.
010	The value given in the Hop-by-Hop extension header is used instead of the Flow Label where it is ignored.
011	DiffServ PHB ID is used.
100	A format that includes the port number and the protocol number in the Flow Label is used.
101	Bandwidth, Delay and Buffer Requirements.
110	Reserved for future use.
111	Reserved for future use.

17 bits left in the IPv6 Flow Label are defined according to the type of approaches they used as indicated in Table 2.1. As stated by Deering and Hinden(1998), packets that have all zero bits marked in them are not participating in flows and hence ignored by the intervening nodes. First initial 3 bits labeled is the default value and will be ignored.

a) Random Number

Random numbers which is generated in the range of 1 to 1FFFF hex can be used to label the packets based on the IPv6 specification. With limited information that can be shared between intervening routers, this approach may not be feasible or efficient to be deployed as compared to the other proposed approaches. Pseudo-random value with the initialized first three bits of the Flow Label field is illustrated in Figure 2.18.

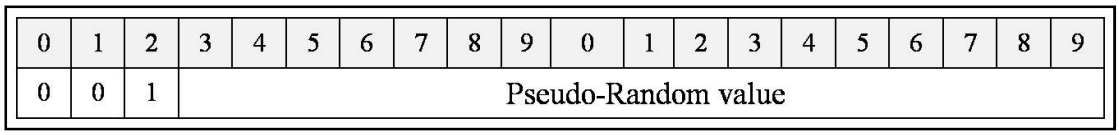

Figure 2.18: Pseudo-random value in IPv6 Flow Label Field format

b) Hop-by-Hop Extension header

Hop-by-Hop extension header as shown in Figure 2.19 can be used to define the Flow Label with IntServ included. With the extension, Flow Label value will be ignored(Banerjee, Preethi, & Sethuraman, 2002) and proceed with IntServ or DiffServ mechanism processing respectively.

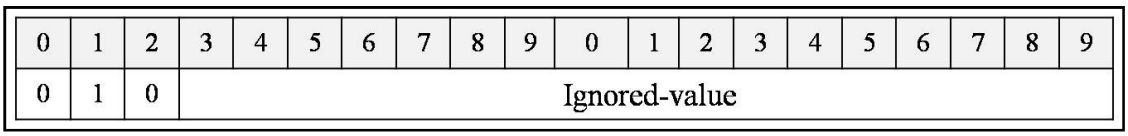

Figure 2.19: Hop-by-Hop Extension header approach

2.4.3.3 Bandwidth, Delay and Buffer Requirements Approach

Multiple parameters are selected to be labeled in the Flow Label as information of the request by application. Bandwidth, Delay, Buffer requirements are proposed by Banerjee, Malhotra and Mahaveer (2002). It defined the 20-bit Flow Label field that includes the options to provide IntServ and DiffServ based on the IPv6 QoS in a hybrid approach where first three bits are using the definition according to Table 2.1 above. Packet Loss and Jitters are

considered initially but are desired to be of minimum value by any application hence the two parameters are left out for the label. Then, three parameters which are bandwidth, delay and buffer requirements are used instead. Soft and hard real time applications in utilizing the Flow Label field are illustrated in Figure 2.20. Bandwidth is expressed in multiple of kbps, delay is expressed in nanoseconds and buffer requirements are expressed in bytes. In optimizing the 17 bits left in Flow Label, it is divided into two categories by using one bit in the flow label.

- Bit 0 is for soft real time applications where it is also known as the Real Time Tolerant (RTT) services in which an average bandwidth requirement and an intermediate end-to-end delay for arbitrary packets are required.

- Bit 1 is defined as hard real time applications or Real Time Intolerant (RTI) which demand minimal latency and jitter as constant bandwidth for the application and a deterministic minimum delay are expected.

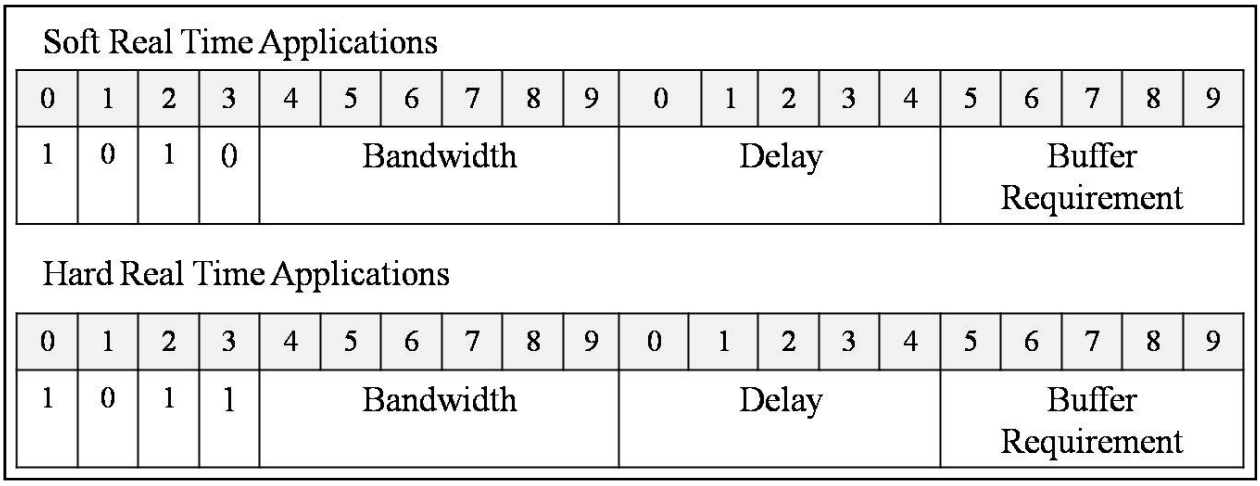

Figure 2.20: Soft real time and hard real time application format

Bandwidth is defined to occupy six bits in the field specifying the bandwidth value required by the transmission while buffer requirements are using five bits out of 16 bits to specify the buffer value required. Delay will use the last five bits to specify the minimum delay requirement.

2.4.3.4 A radical approach in providing QoS over the Internet using the 20-bit IPv6 Flow Label Field

Bandwidth, Delay and Buffer requirements approach in (Jagadeesan & Singh, 2002) is suggested to use the entire 20 bits of instead of the 16 bits as mentioned above. This draft suggested eight bits of the Flow Label to label bandwidth requirements where the first bit is used to specify whether the bandwidth required is minimum or maximum value. The next five bits are suggested to label the delay requirements whereas the last seven bits are used like as proposed before buffer requirements. A look-up table is used to map the Bandwidth, Delay and Buffer requirement values globally and uniformly distributed all around the network. This approach as shown in Figure 2.21 required all nodes in the network to acknowledge the usage of this Flow Label regardless if the node supports it or otherwise.

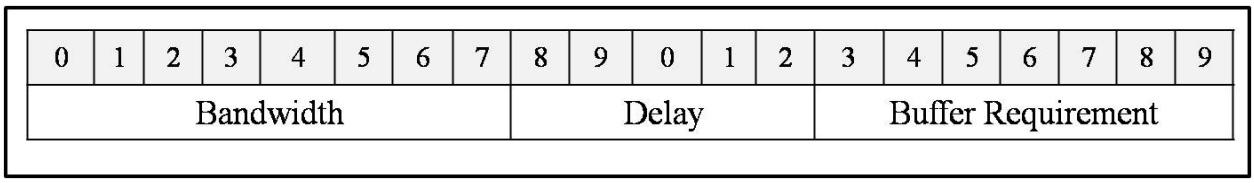

Figure 2.21: 20 bits Bandwidth, Delay and Buffer requirements format

2.4.3.5 Using the 20-bit Flow Label field in the IPv6 header to indicate desirable Quality of Service on the Internet

Prakash(2004)proposed a new specification in order to support different models and requirements to define the QoS support by extensively using the 20 bits field. First few bits are used as identifier in this approach and detailed explanation is listed as Table 2.2:-

Table 2.2 : Classification for identifiers

Bit Pattern	Approach
00	No QoS requirement (default QoS value)
01	Pseudo-Random value is used for the value of Flow Label
10	Support for Direct Parametric Representation
1100	Support for the DiffServ Model
1101	Reserved for future use
111	Reserved for future use

Bit pattern 00 of initial bits in Flow Label will be used to label all applications that do not request for any QoS support in the network while bit pattern of 01 is an approach used for pseudo-randomly generated value as specified in RFC 2460 (Deering & Hinden, 1998)and (Rajahalme, Conta, Carpenter, & Deering, 2004). The generated value will be in range of 0 to 3FFFF and this approach support IntServ model. In bit pattern of 10, the author proposed four parameters to be used instead. They are One Way Delay (OWD), IP Delay Variation (IPDV), Bandwidth (BW) and One Way Packet Loss (OWPL) parameters. Out of 20 bits, two bits are used to set this approach for intervening routers and the third bit is used to distinguish the types of application traffic as in RTT (Real Time Tolerant) or RTI (Real Time Intolerant). Third bit with a 0 labels the traffic as RTT where the requested values of OWD, IPDV, BW and OWPL are not strictly needed and it may work with the best QoS available in the network. In contrast, value of 1 indicates the traffic to be RTI as the four parameters in the label are needed to be stringently met. They are illustrated in Figure 2.22.

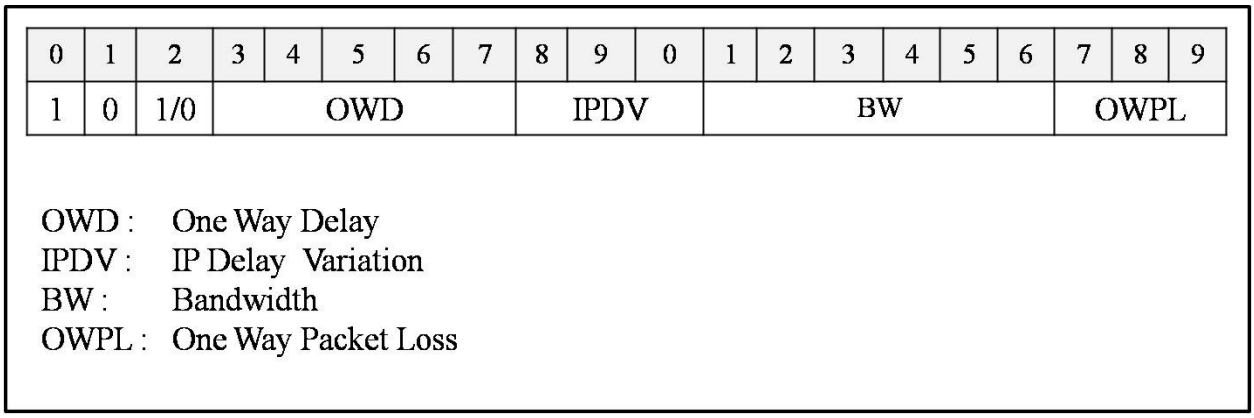

Figure 2.22: Direct Parametric Representation of desired value Flow Label Format

One Way Delay (OWD) is represented by five bits in the field where it is measured in nanoseconds and the following three bits are used to label the IP delay variation (IPVD) calculated in decimal with values ranging 1 millisecond to 128 milliseconds with the formula shown in (1) is IP Delay Variation (IPDV) formula.

$$IP\ Delay\ Variation\ (in\ decimal) = 2^n \times 1\ millisecond$$
$$(n\ is\ specified\ in\ the\ IPDV\ field) \quad (1)$$

Bandwidth (BW) occupied the next six bits in the field. First of the six bits are used to denote the requested value is the minimum or maximum expected value as specified. Remaining five bits are used to specify the value of bandwidth where initial 0 bit specified it as minimum and 1 as the otherwise and calculated in Kbps. Final 3 bits are utilized to represent the probability of One Way Packet Loss (OWPL) measures in percentage of loss.

DiffServ Model support is included to specify the use of the field as DiffServ PHB-ID. As stated, this approach allows the DiffServ MF Classifier which in the incoming flow label value to be matched for QoS requirements with initial of 1100 in the field followed by the DiffServ PHB-ID. Bit patterns of initial 1101 are reserved for future usage hence the field will be ignored as reservation for future(Prakash, 2004). This proposal required computations on calculating the different metrics and this QoS provision requires the intervening nodes to

be able to signal the source node of the transmission on the failure in providing the requested QoS with ICMPv6 (Internet Control Message Protocol version 6).

2.5 Current implementation of IPv6 Flow Label QoS in Mobile Ad Hoc Network

Routing mechanism over the IPv6 network is implemented as an extension to the previously proposed, QOLSR that takes in Bandwidth and Delay as their QoS parameters. However, routing protocol QOLSR extension is the routing protocol that provides a similar intention in improving network performance with IPv6. Thus, this protocol will be evaluated and studied for their utilization in this research in improving QoS services over highly dynamic wireless ad hoc network with IPv6 features.

2.5.1 An Efficient QOLSR Extension for QoS QOLSR (ext-QOLSR)

Link State routing approach makes network information easily available as the state of the link is crucial in establishing transmission in this approach. QoS routing over Optimized Link State Routing (OLSR) is enhanced by Badis, Munaretto and Agha(2003) in their paper to provide QoS where they use multiple metrics over the selections of MPR. Besides, an extension is proposed to this QOLSR protocol in order to provide integrated services in supporting multimedia applications over the Mobile Ad hoc Networks (MANET). This extension makes use of various QoS parameters by utilizing IPv6 Flow Label.

QOLSR is a proactive routing protocol with advantage of immediate availability routes due to its nature. Multiple Point Relays (MPRs) are selected as the traffic controller where it reduces the flooding of information in the network. With a subset of node in their vicinity, the MPRs are the only points that retransmit the packets across the network. QOLSR

protocol includes the quality parameters into the link state routing protocol where delay and bandwidth metrics are taken into consideration.

In the paper, Flow Label utilization adopts an approach which is presented by Banerjee, Malhotra and Mahaveer(2002). Parameters that are presented to label are Bandwidth (expressed in kbps), delay (expressed in nanoseconds) and buffer requirements (expressed in nanoseconds). One out of six bits in the bandwidth field will be used to label whether the labeled value is maximum or minimum while five bits are for buffer requirements and lastly five for delay value. It is shown in Figure 2.23 in the QOLSR extension on the routing protocol.

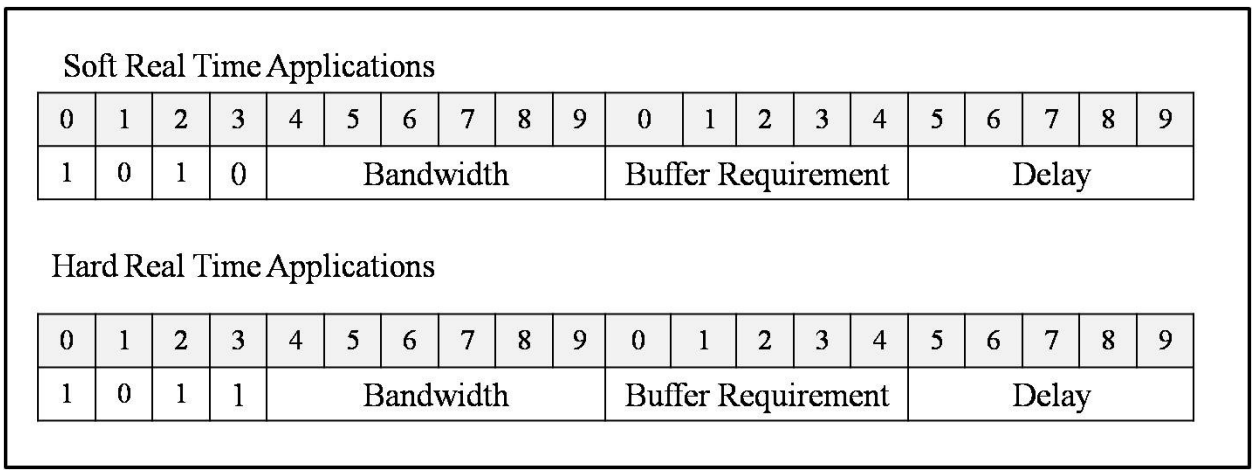

Figure 2.23: QOLSR extension in Flow Label IPv6

In this QOLSR-extension algorithm, hard real time applications will be using a shortest path algorithm. Delay between the nodes has to be minimum for this case and it is proposed to use the well-known Dijkstra routing algorithm whereas the bandwidth metric in the routing will apply a variant-Dijkstra algorithm. For soft real time applications, it considers the Delay and Bandwidth Constrained Least Hop (DBCLH) where given two constants, the minimum bandwidth and maximum delay. The DBCLH will find a path that satisfies both the

requirement of delay lesser than the given constant and bandwidth more than the given bandwidth. It proposed an efficient heuristic based on the Lagrange Relaxation algorithm to solve this routing and if there's no path to satisfy the constraints, Best Effort method will be used. In this algorithm, nodes selected to be MPR are required to be as stable as possible as they are the relay in forwarding the data packets. However in highly dynamic nature mobile ad hoc network, selected MPRs might be the cause of unstable routes in the topology. As they move out of the network, the topology changes and affects the whole network subsets formation. Besides, it has no multiple instances to differentiate the same application running two communicating ends in this way of utilizing the Flow Label.

2.6 Conclusion

Reviews on different QoS architectures and routing protocols are done with pros and cons analyzed. In an ad hoc network where resources are scarce, QoS for real-time applications are important. Under a highly dynamic ad hoc network, resource allocation has to be done in effort to have a better network performance. After analyzing the routing protocols available, reactive routing protocol is more suitable to be used in dynamic ad hoc network where it does not use up the resources to update all nodes in the network. Besides, in reserving the resources for a better QoS transmission, IPv6 Flow Label field can be used to provide special handlings to the requesting applications. In this proposed routing scheme, Implicit Source Routing with Dynamic Source Routing is enhanced. Flow Identifier that is used in flow establishment can reduce the per-packet overhead in the route. At destination node, appended source route can be analyzed to select the best route before proceeding to the transmissions. Hence, reliable and more stable routes are needed in the dynamic ad hoc network with high frequencies of link breakages.

CHAPTER 3

ENHANCED DYNAMIC SOURCE ROUTING (En-DSR) IMPLEMENTATION

3.1 Introduction

Routing in mobile ad hoc network, adversity such as link failure is inevitable. Transmissions are often halted and reinitiated over the network which leads to poor performance for end users. Mobility in the wireless ad hoc network is a convenient yet destabilizing feature where mobility adversely affects the network topology. Changes in node position or whenever a node leaves a network, link breakages are often encountered. It is because of the changes in topology that involves the particular moving node in transmission that unable to transmit data packets as usual without any notification. Poor performance that is caused by the dynamic nature of the mobile ad hoc networks requires improvising the QoS in order to alleviate the problems encountered. In this chapter, an adaptive routing scheme which enhanced the well-known Dynamic Source Routing (DSR) routing protocol is proposed to route the data in the highly dynamic mobile ad hoc network. Changes in the topology required transmissions to use more stable routes for the transmission to take place. Therefore, proposed routing Enhanced-DSR (En-DSR) takes advantages of the flow based transmission with optimized Flow Label in IPv6 packet header to manage sufficient QoS for transmissions to take place in the ad hoc network. IPv6 flow label was introduced to provide special handlings for requesting applications where it is yet to be standardized. As several proposals are discussed in Chapter 2, information can be included via the Flow Label field for resource reservation. It is therefore proposed here to alleviate the shortcomings brought in by the wireless connection and improve the network performance by inducing QoS into the routing algorithm. Flow Label will be utilized in providing QoS in the network where we assume all packets are using the Flow Label extensively for the transmission.

3.2 IPv6 Packet header

IPv6 packet header contains fields that have been reconstructed and revised from the previous IPv4 packet header format. The structure involves the IPv6 packet header in providing QoS feature when handling special requests made by real-time applications across the network where IPv4 is only providing "best-of-effort" services without any specialized field on the QoS provision. Thus, effort was made as such introduction of 20 bits Flow Label is made to provide special handlings to the requesting applications.

3.2.1 Enhanced Flow Label Structure

Flow Label is a 20 bits field located in the IPv6 packet header for QoS implementation yet been exploited. As discussed in Chapter 2 section 2.4.3, a few Internet drafts proposed definitions on how to utilize the field as aforementioned have their own advantages and disadvantages. Flow Label is used to include various information specific requests on transmission and distinguish the flows from the rest of the transmission in within the mobile ad hoc network. Hence in the proposed Enhanced DSR (En-DSR) algorithm, IPv6 Flow Label is redefined as a hybrid approach in providing QoS in ad hoc networks as shown in Figure 3.1. The proposed ways of utilization are elaborated subsequently in Table 3.1.

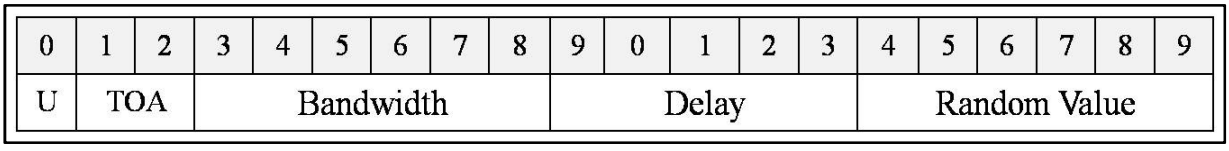

Figure 3.1: En-DSR Flow Label Utilization format

Table 3.1: Flow Label bits Utilization

Columns	Number	Remarks	
Used (U)	1 bit	Indicate usage of Flow Label that either used (bit 1) or ignored (bit 0) by the receiving nodes	
Type of Applications (TOA)	2 bits	00	Non real-time applications
		01	Soft real-time applications
		10	Hard real-time applications
		11	Reserved for future
Bandwidth	6 bits	1 bit to indicate maximum or minimum bandwidth required where the rest of 5 bits label the value of bandwidth required	
Delay	5 bits	Maximum tolerable delay	
Random Value	6 bits	Differentiate transmissions made by the same application between two communicating end nodes	

3.2.1.1 Type of Applications (TOA)

As explained in Table 3.1, soft real-time application services are indicating Real Time Tolerant (RTT) applications where they can manage with the QoS provided despite requested QoS is not met. They have an average bandwidth requirement and an intermediate end-to-end delay (Agha & Badis, 2004). To the contrary, hard real-time applications services are indicating Real Time Intolerant (RTI) which demands minimal disruptions to preserve their transmissions. Hard real-time applications need requirements to be strictly reserved despite being not feasible it provides a prioritize treatment in reserving for variant of application requests.

3.2.1.2 Bandwidth

Bandwidth in the Flow Label field occupied 6 bits where one of the bits is used to specify the minimum or maximum value required. A zero bit is indicating minimum required bandwidth whereas a 1 bit indicates maximum required bandwidth. 5 bits of the value can be represented with a formula as below where it is implemented in (Prakash, 2004).

$$Bandwidth\ (decimal) = 2^n \times\ 4\ Kbps \qquad (2)$$

Formula (2) shows bandwidth utilization where *n* is the value of the 5 bits in the Flow Label. According to implemented (2) formula, the bandwidth value in the Flow Label can be calculated in range of 4 Kbps to 8 Tbps and calculation results are shown in Table 3.2.

Table 3.2: Bandwidth Value

Binary Value	Decimal Value	Bandwidth Value
00000	0	$2^0 \times 4$ Kbps = 4 Kbps
00001	1	$2^1 \times 4$ Kbps = 8 Kbps
00010 ….	2	$2^2 \times 4$ Kbps = 16 Kbps
01000 ….	8	$2^8 \times 4$ Kbps = 1 Mbps
10010 ….	18	$2^{18} \times 4$ Kbps = 1 Gbps
11100 ….	28	$2^{28} \times 4$ Kbps = 1 Tbps
11111	31	$2^{31} \times 4$ Kbps = 8 Tbps

3.2.1.3 Delay

Delay occupied 5 bits out of 20 bits in the Flow Label field where the bits can be calculated for a value that specifies tolerable amount of delay. Below is the formula implemented in (Banerjee, Malhotra, & Mahaveer, 2002)where the tolerable delay has range of 4 nanoseconds to maximum of 8 seconds with a calculation as shown in (3) and their respective values in Table 3.3.Formula (3) shows the calculation for tolerable delay where *m* is the value of the 5 bits.

$$Delay\ (decimal)\ =\ 2^{\mathrm{m}}\ \times\ 4\ nanoseconds \qquad (3)$$

Table 3.3: Delay Value

Binary Value	Decimal Value	Delay Value
00000	0	$2^0 \times 4$ nanoseconds = 4 nanoseconds
00001	1	$2^1 \times 4$ nanoseconds = 8 nanoseconds
00010 ….	2	$2^2 \times 4$ nanoseconds = 16 nanoseconds
01000 ….	8	$2^8 \times 4$ nanoseconds = 1 microseconds
10010 ….	18	$2^{18} \times 4$ nanoseconds = 1 milliseconds
11111	31	$2^{31} \times 4$ nanoseconds = 8.58 seconds

3.2.1.4 Random Value

In order to distinguish different flows created by the same applications for two communicating end nodes, 6 bits are used to differentiate the multiple instances in decimal form. 6 bits are able to uniquely identify 32 flows for a single transmission if they would have the same bandwidth requirement and delay tolerance.

3.2.2 Replacing Flow ID in Implicit Source Routing

In alleviating the disadvantage of increase per-packet overhead in DSR where the source route header has to be attached to every packets traversed in the network, Implicit Source Routing is proposed to preserve the advantages of source routing with flow identifier (flow ID) field. This presented a reduction of associated per-packet overhead in DSR as flow ID is used to label flows initiated from a source to the destination along the route. Flow ID uniquely identified the combination of the source node address, destination node address and together with the flow ID. However, flow ID is located in conventional IPv4 DSR Flow State Header after DSR Options header where a bit is used to flag the usage of the DSR Flow State Header as show in Figure 3.2 (Hu, Johnson, & Maltz, 2001).

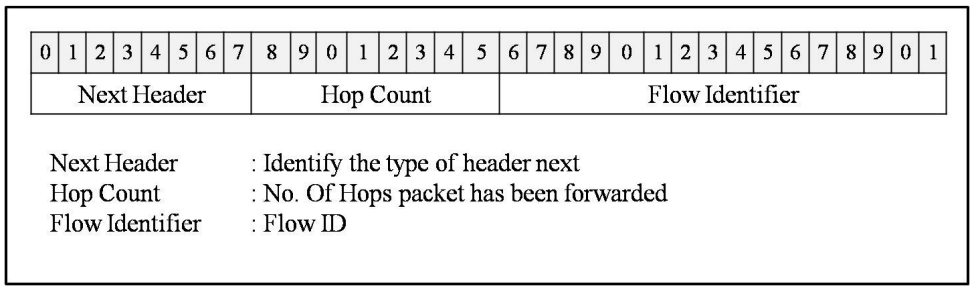

Figure 3.2: Conventional DSR Flow State Header in Data Packet IPv4

Reading the combination required the IP header packet to get the Next Header value for navigating through different extension headers. In this proposed algorithm, Flow Label is proposed to replace the flow ID. With unique combination of source node address, destination

node address and Flow Label value, distinguishing flows are simplified where the 3 fields are located in the IP packet header as shown in IPv6 proposed packet header format in Figure 3.3.

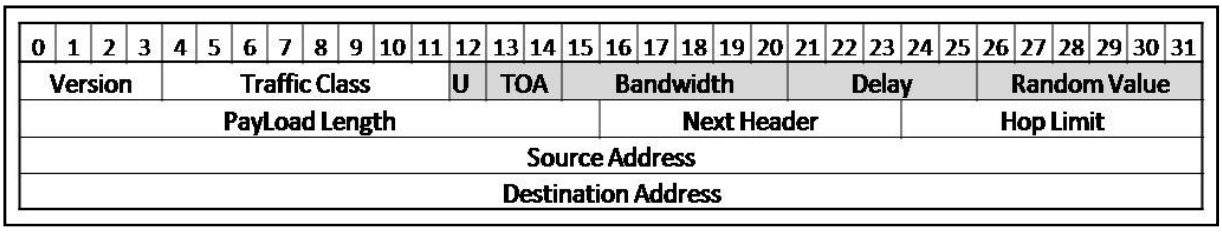

Figure 3.3: IPv6 packet header with proposed Flow Label utilization

3.3 Enhanced Dynamic Source Routing (En-DSR) Algorithm Description

En-DSR is a routing algorithm enhanced from the well-known routing protocol DSR. Hence, it is inheriting the features structured in it. DSR is a reactive routing algorithm that routes based on-demand efficiently over multi-hop ad hoc networks where they do not rely on routing table at each intermediate device (Johnson, Maltz, & Broch, 2001). This protocol allows nodes to search for routes across multiple hops in the network for any destination respectively. All nodes are able to self-organize and self-configure in the wireless ad hoc network without any centralized management from access points and so forth where the routes are established according to the mechanism of DSR protocol. This eases the connection configuration process where the nodes in the wireless ad hoc network can be formed and get connected among themselves as all nodes in the ad hoc network mutually agreed upon in relaying data packets on behalf of the source nodes. Changes in the topology in such either a new node joins in or leaves the network, the routing protocol will maintain all the routing process. In this protocol, it is divided into two phases that will involve the whole routing mechanism for the network which are the Route Discovery and Route Maintenance as shown back in Figure 2.2. En-DSR algorithm supports the migration of IPv4 to IPv6 in the network connections. With introduction of IPv6 Flow Label field in effort to provide QoS to the

network transmission, information conveyed by the field itself helps in managing resources for the requesting applications in the ad hoc networks. Besides, as mentioned beforehand flow id will be retrieved with the combinations in the IP packet header instead of involving Options header in the transmission. In a topology which not all nodes are moving vigorously at one time but several nodes move at different speeds and directions unpredictably, a stability value is suggested in this algorithm where it estimates the mobility values of nodes in the highly dynamic ad hoc networks.

3.3.1 Route Discovery

Route Discovery happens based on the demand made by the originator node of the transmission whereby no pre-existing information about the link condition as well as information of previous used routes available is supplied. All information is only enquired at the time of the demand being made. Routes are discovered in this phase where a node **S** (source) is initiating a transmission to another destination node **D** (destination). When node **S** initiated a transmission to destined node **D**, node **S** will firstly go through its cache to search for possible routes reaching node **D** that was taken beforehand. Source route will be placed at the packet header giving the sequence of hops for the packets to follow in order to reach node **D**if route is available in the route cache. However, if there is no route reaching the destination in the cache node **S** will ignite the Route Discovery protocol to find a new route destined to node **D**.

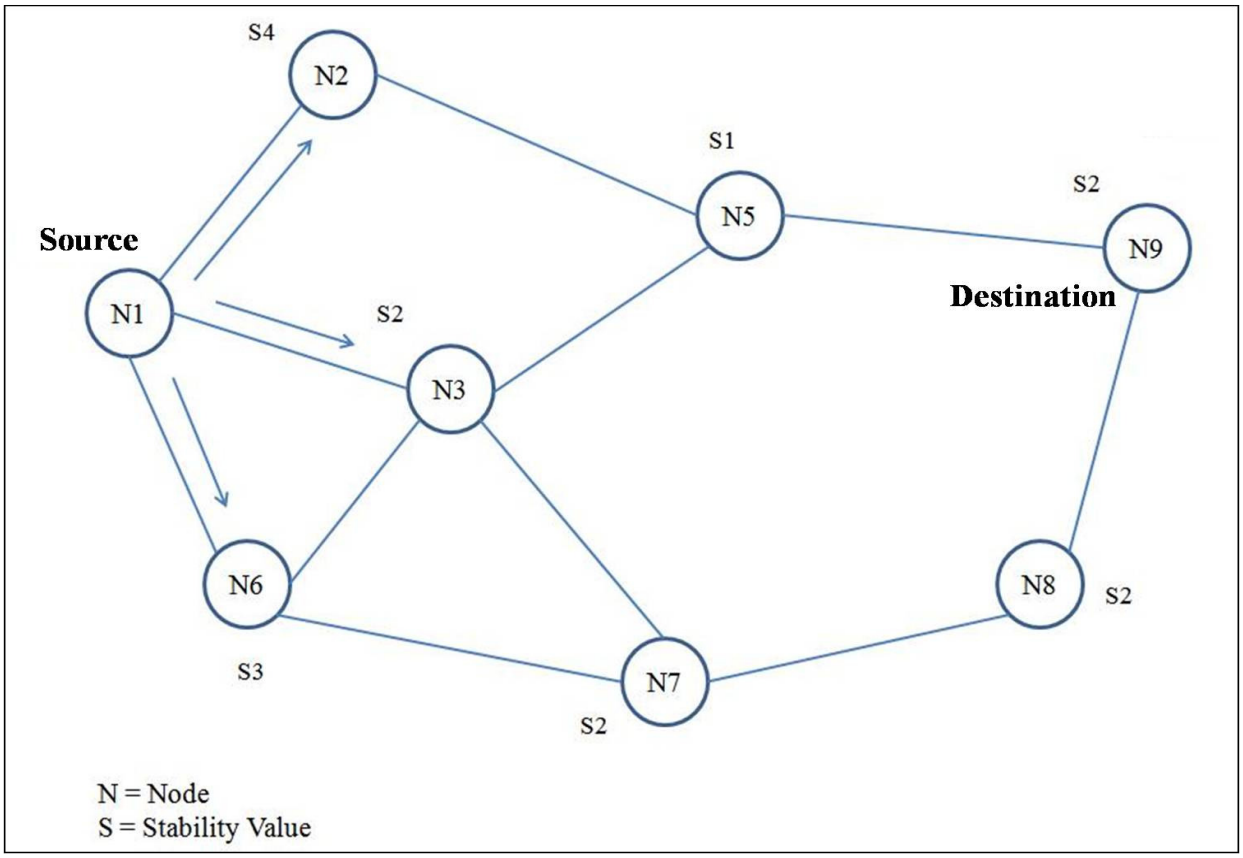

Figure 3.4: Route Discovery example where N1 is originator and N9 is the target node

As an example given in Figure 3.4, 8 nodes connected to each other wirelessly. Source node, N1 will send out Route Request (RREQ) packet which is a single local broadcast packet to N1's respective neighbors within the transmission range. They are N2, N3 and N6. Each of the RREQ message identifies the initiator as source node and target as the destination node with a unique *request id* attached and determined by the initiator of the transmission. All nodes that receive RREQ packets which they are not the destination node they will rebroadcast the packet after piggybacking their own address to the list of node constructing a source route in the end. Specific target node will keep the source route which indicates a sequence of hops into the cache before initiating a unicast Route Reply (RREP) packet back to the originator.

For a RREP packet received by a node which has a route to the specific destination node in its route cache, the node will send RREP packet back to the source along with a concatenated route from the RREQ packet together with the route from the cache. This way allows route to be returned and discovered sooner as well as reducing the overhead from the RREQ rebroadcasting. Optimization in DSR is utilized where a "promiscuous" receive mode will allow nodes to update their respective route cache based on the source routes learnt from the surrounding when receiving packets that are not intended for it as the next hop or by forwarding the packet along the route(Hu, Johnson, & Maltz, 2001).

3.3.1.1 Stability Value

Route selection process will take place where the destination node returns a RREP packet back to the source node which is subsequently followed by the data transmission. Contrasting to the conventional DSR where it uses lowest hop count as a route selection metric, an enhancement is proposed in En-DSR where a stability value will be calculated as one of the metric.

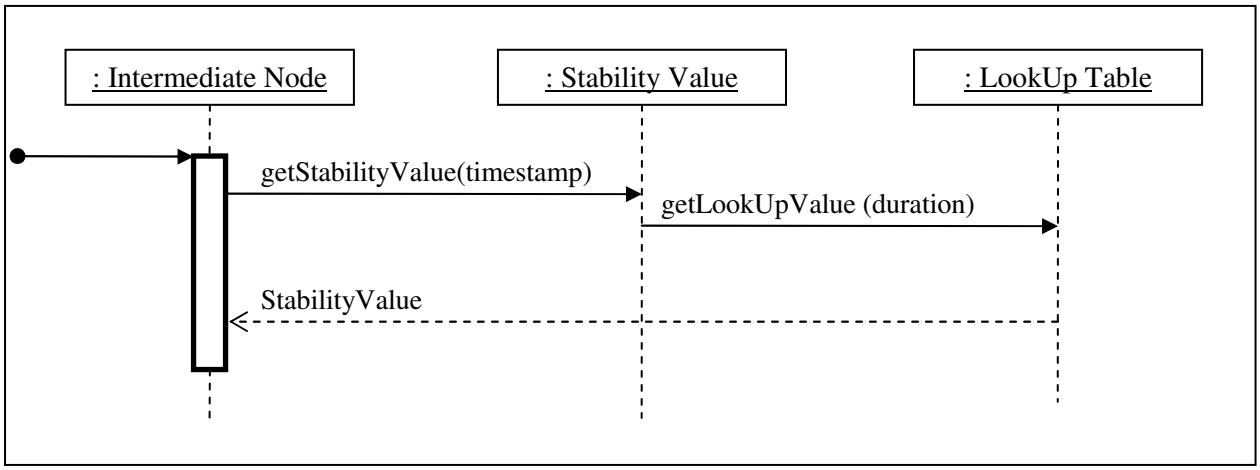

Figure 3.5 Sequence Diagram for Stability Value calculation in node

Stability value is a value that reflects the stability of a node in terms of the duration that the node has been statically located at one position. In Figure 3.5, it shows the sequences when an intermediate node receives a RRER packet and carries out a calculation on stability of the node. A timestamp of the node will be used to calculate a duration whereby it will then be used subsequently to lookup for a value in a lookup table. Formula (4) shows the calculation of the duration in a node being statically located at one place and then the duration will be used to find a value as the stability value from a look up table in Table 3.4.

$$Duration\ of\ being\ static\ = Request\ receiving\ time - Node's\ Timestamp \tag{4}$$

The longer a node has been located at the location, the smaller stability value will be assigned to the node. Here it is assumed that the node will update the timestamp of the most recent movement for the stability calculation. Timestamp of most recent movement will be subtracted for the time of the packet arrival to find the duration of the node stability value. Whenever the node's movement is taking place, the timestamp will be updated to the current time. Each route request packets received, a stability value will be added according to a lookup table in Table 3.4 along the route of data packet delivery for their specific value based on their respective time of arrival. According to the Figure 3.4, stability value illustrated is written with initial of an alphabet S (e.g. S4 for stability value 4). Below is the proposed stability value look up table:-

Table 3.4: Stability Value Look Up Table

Duration (seconds)	Stability Value
>= 10	1
<10	2
<8	3
<6	4
<4	5
<2	6

A lower stability value reflects the stability of the intervening nodes along the route between the originator and target nodes. It indicates that those intermediate nodes in between have been there for a period of time. Stability value has been chosen to reveal the link lifetime as mobility comes in as one of the reasons of link failures. As links live up to transmission in the ad hoc network, the nodes between the routes have to be located within reach. Therefore, a time difference is used as a value to estimate the length of time for each node in the ad hoc network. Route selection is made based on an average of stability value of 2. If on average of all node is two in a route, the route will be used and if otherwise, shortest route path will be selected to avoid route selected that contains imminent moving nodes. Durations in Table 3.4 is set in the specific ranges after evaluating the node lifetime as at ODLW Routing Mechanism in Figure 2.10 as well as the duration taken for the whole simulations to be done. With a lower stability value on a route, it reflects that intermediate nodes within the route will be located at the position on a relatively longer time for the transmission to take place. With reliable route, transmission will be retransmitted over the network. Based on the Figure 3.4, possible routes are listed as below in Table 3.5:-

Table 3.5: Possible Routes of the scenario

Route No.	Possible Route	Route Stability Value
1	N1-N2-N5-N9	0 + 4 + 1 = 5
2	N1-N3-N7-N8-N9	0 + 2 + 2 + 2 = 6
3	N1-N6-N7-N8-N9	0 + 3 + 2 + 2 = 7
4	N1-N6-N3-N5-N9	0 + 3 +2 +1 =6
5	N1-N6-N3-N7-N8-N9	0 + 3 + 2 + 2 + 2 = 9
6	N1-N3-N5-N9	0 + 2 + 1 = 3

In Table 3.5, first potential route discovered consists of nodes numbered 2 and 5 between the source and destination nodes where each respective node has their own stability value. Summation of the first route gives a total value of 5 whereas second route has three intermediate nodes in between. It summed up a stability value of 6 for this route. Third route discovered 3 intermediate nodes in between with different accumulated stability values which is 7. Fourth route has a total of 6 stability value while fifth route has the longest intermediate nodes in between with a summation of 9 as total stability value. Last route discovered has the lowest total of stability value of 3 where it shows that those two intermediate nodes have been relatively static for a period of time longer. Hence, based on the stability value the route selection will take into consideration of the nodes in between before routing data packets across the ad hoc network. This stability value is appended onto the requesting packets without incurring additional data packets into the ad hoc networks in order to get a reliable route therefore, the overhead is reduced in comparison to table-driven routing protocols. Besides, flow label information will be conveyed to the intermediate nodes that process the data packet whereby the resources will be reserved and managed accordingly.

3.3.1.2 Flow Label QoS Provision in Route Discovery

Once a route discovery is done, the flow state mechanism will take place to allow routing to occur without the explicit source route header being attached to the packet. Flow state in En-DSR is similar with the conventional DSR where it is initialized by the first packet using the source route without the list of nodes for the transmission. Instead of generating the flow ID in the Option Header, Flow Label will be used. All field will be labelled with the required bandwidth, tolerable delay and unique random value that contributed to a Flow Label value by the application when tramission is initiated. Subsequent packets are routed along the same flow under the same Flow Label. Establishment of the state will automatically expire itself when it is no longer in need as it is created on “soft-state”. All intervening nodes will reserve the required handlings labelled in the Flow Label fields to ensure the QoS is provided to the transmission. If the node is unable to reserve the resources needed, “best-of-effort” service will be used as last resort. In order to create a new flow, node needs to generate the Flow Label value by increasing one on the random value generated to avoid creating the same value on different transmission. Combination of the three tuples (source node address, destination node address and Flow Label value) needs to be uniquely labelling the transmission flow.

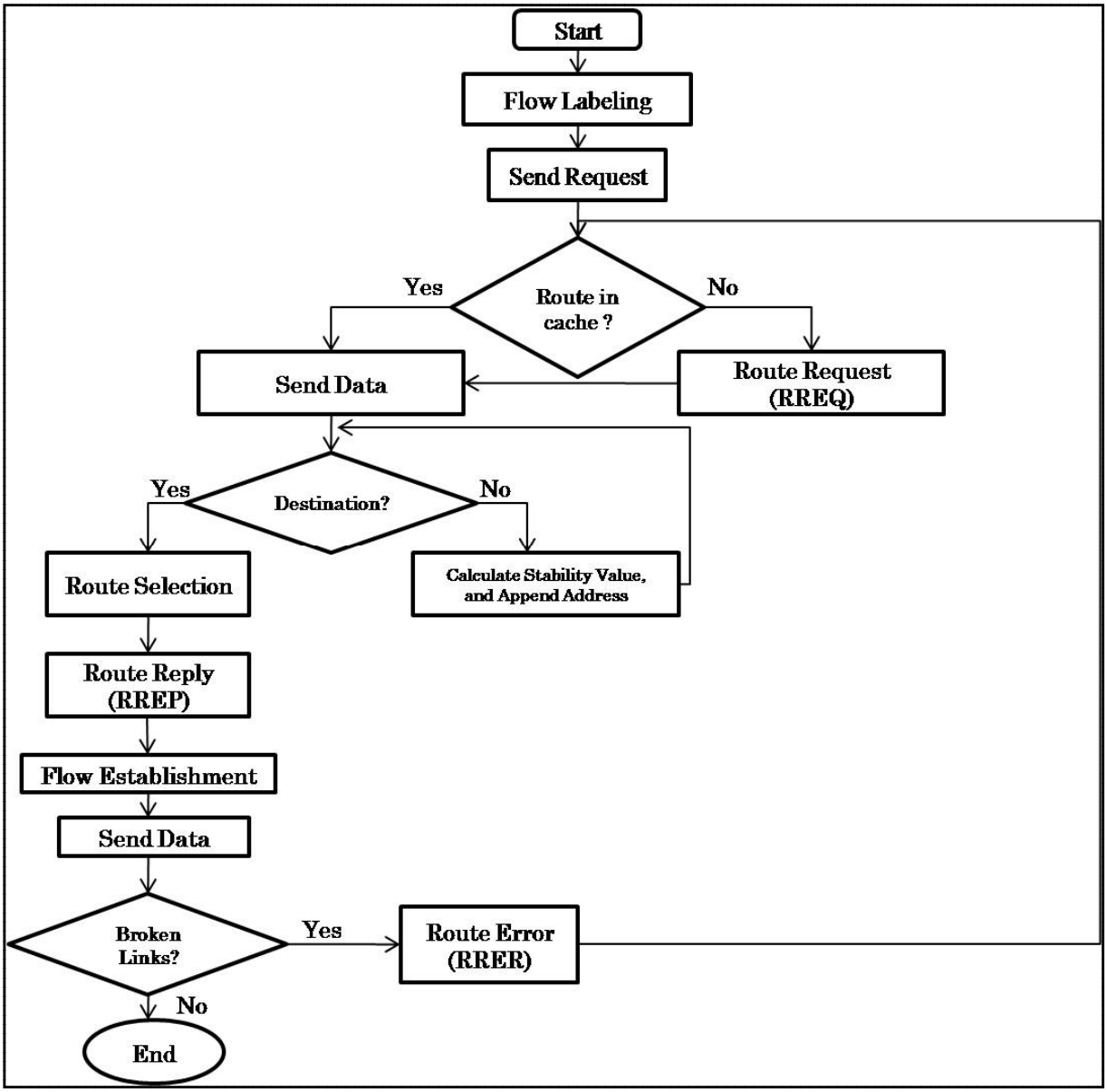

Figure 3.6: Flow Chart of En-DSR algorithms

Differing from the conventional DSR flow state mechanism, En-DSR does not implement the default flow option to allow more adaptive routing to take place. Stability value allows routing to take place via a more stable route instead of the lower hop count route. This avoids the frequent shifting of route selection for default flow between more stable route or shorter route. Generally, the data packets are treated normally from sender to receiver but it will contruct as a flow establishment after discovering the route selected for transmission as shown in Figure 3.7. According to RFC 2460, Flow Label value must not be changed by

intermediate nodes hence, it preserves the flow to only be treating packets specially under the same flow. Upon receiving the data packets as establishing flows, Flow Table will store the state information like conventional DSR as described in Section 2.2.1.2.2 of the flow route and it will be removed as the delivery of the packet ends. The whole flow of the algorithms are shown in the flow chart as illustrated in Figure 3.6 with different route selection metric and Flow Label field utilization.

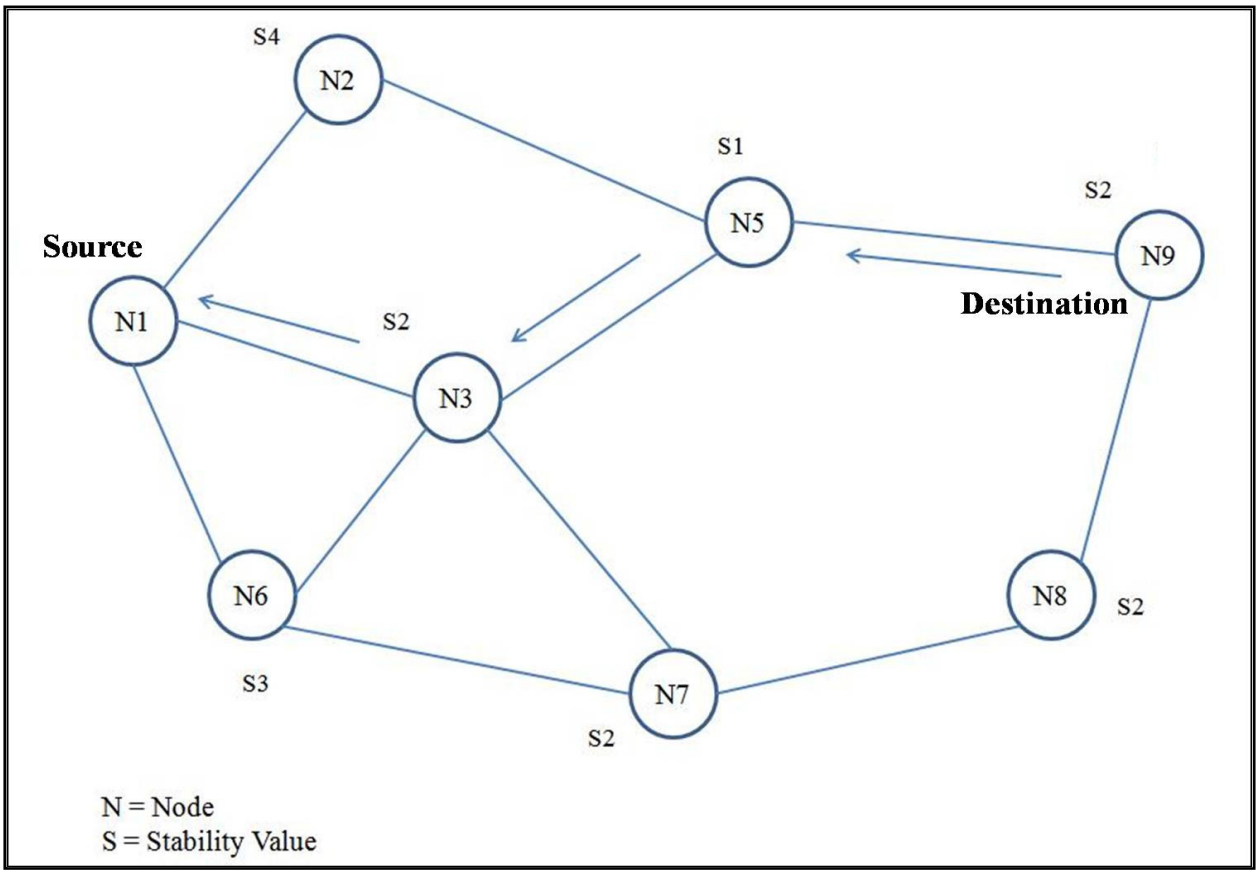

Figure 3.7: Route Reply via route 6

Among the potential routes discovered over the route discovery phase, target node has 6 possible routes for routing over the network where it will choose the most stable route of all. In Figure 3.7, route 6 will be chosen as it has the lowest accumulated stability value, 3 that indicates a lower possibility of link breakages along the route. Conventional route will choose the shortest path where the number of nodes in between is the smallest. However, in En-DSR

accumulated stability value will be considered first. If the shortest hop count has a stability value of an average of 2, the node will be used otherwise the most stable route will be selected. When there is no stable path available, it means the stability value of the nodes is more than a threshold calculated by the length of possible route multiplying 2 it will select the shortest route instead. For an example, a route has 4 nodes to reach the target node will be multiply by 2 equalized or less than to stability value threshold of 8 the route will be chosen. Value of stability equal to a total number of 8 is to show that of all intermediate nodes they have duration of being statically located at that position relatively for more or equal to ten seconds according to Table 3.4 Stability Value Look Up table. If all potential paths are exceeding their threshold, shortest path will be used to route the transmission instead. Below Figure 3.8 shows a flow chart for route selection with average stability value of 2 as one of the evaluated parameters for selecting a reliable route.

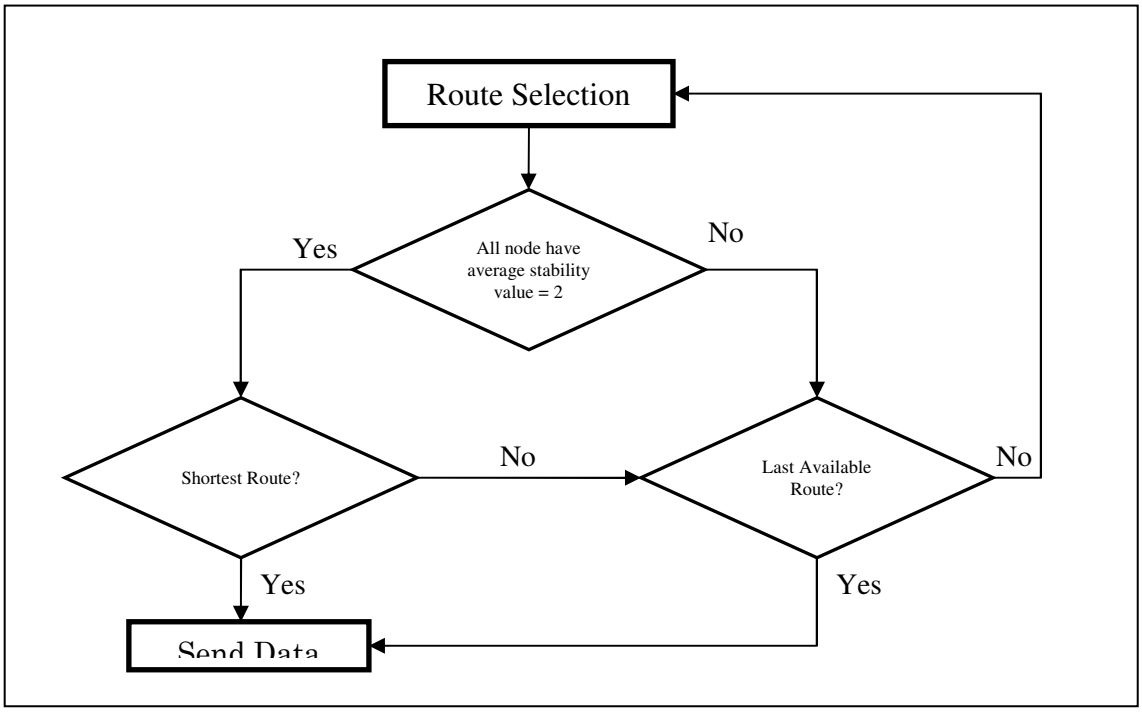

Figure 3.8: Route selection flow chart with stability value as metric

When a route is selected, target node will notify the originator to proceed for the transmission by returning a Route Reply (RREP) packet. The sequence of hops will be copied and sent back to the originator subsequently store into the Route Cache for the following transmission. Intervening nodes that have routes leading to the target node when they receive the RREQ packets may reply from its cache known as "cached Route Reply" (Maltz, Johnson, & Hu, 2004) where it will reduce the overhead of particular phase by reducing the RREQ rebroadcasting. Route can be updated in their cache based on the received source routes in the packet where the optimization known as "promiscuous" is allowed. Featured in the conventional DSR "promiscuous" mode is to allow intervening nodes to overhear to the surrounding transmission and eventually learn the possible routes for future usage.

3.3.2 Route Maintenance

When a transmission is interrupted by the changes of topology at an unpredictable time, route maintenance will be carried out where it will notify node S that current route is broken. Acknowledgement is used to confirm the receptions of data by the respective nodes. As acknowledgment request has been retransmitted to the maximum number of attempts, the link will be considered as broken. Such route will be deleted from the route cache and Route Error packets (RRER) will be sent to each node that has sent a packet over the link since the last acknowledgement. In the scenario given, Node S will remove the broken link from the route cache and sender will use other possible routes that reaches node D to resume the transmission. However, if there is no possible route reaching node D in their cache after the interruption Route Discovery will be initiated once again to find a possible route for the transmission.

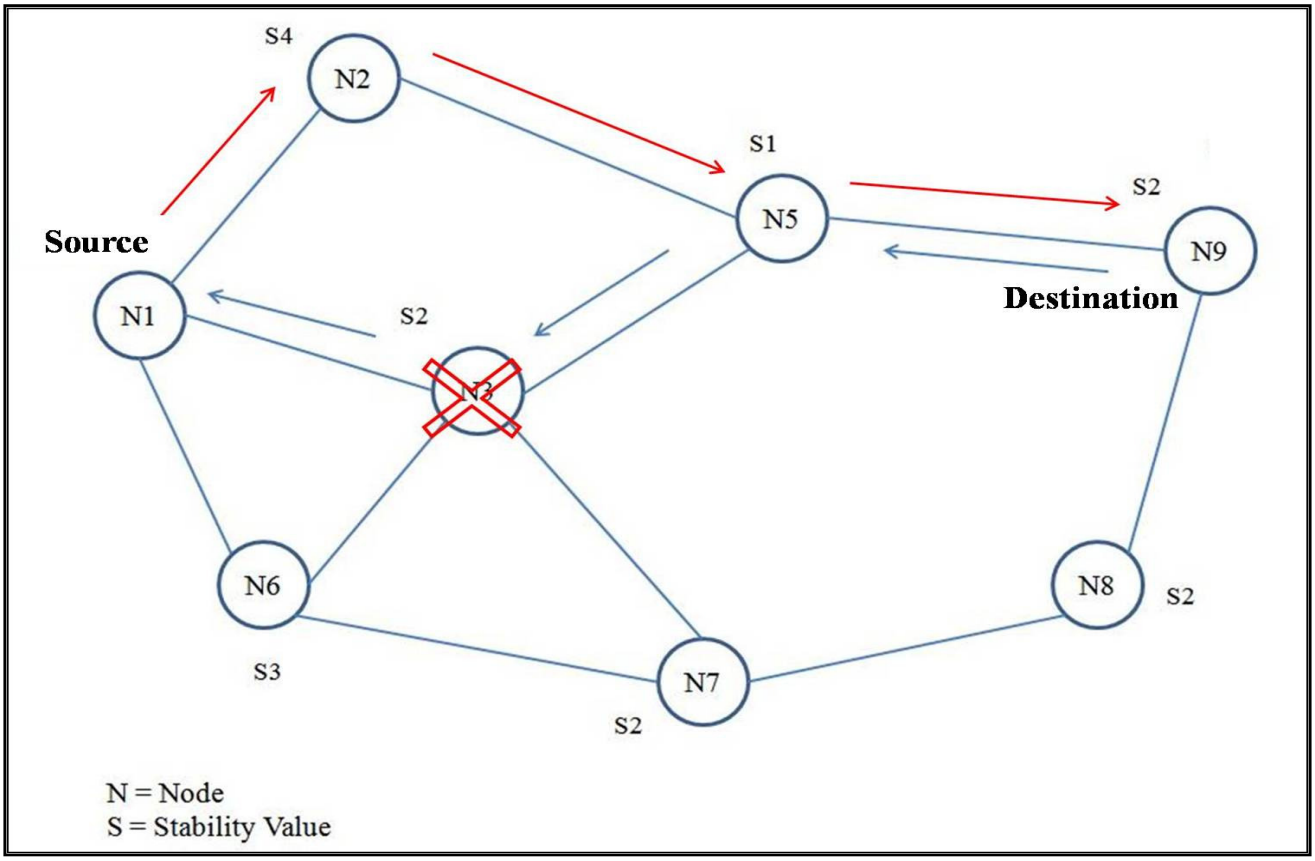

Figure 3.9: Node 3 (N3) moved out of the topology

In Figure 3.9, node 3 (N3) moved out of the network and changed the topology. If after the attempts of acknowledgement request are made unanswered, the sender will take the link as a broken link. Hence, route maintenance will be taken place where the link will be removed from its Route Cache and Route Error (RRER) is sent. As illustrated in Figure 3.9, link between node 1 and node 3 is broken and new route is selected to resume the transmission. Route 1 is selected to route the data transmission over the network topology. As the source node receives a RRER for its transmission, this source node will propagate the RRER packet to its neighbor by piggybacking the RRER onto the next RREQ to notify all nodes about the broken link in avoiding using the stale link.

3.3.2.1 Automatic Route Shortening (ARS) Settings in En-DSR

In En-DSR, ARS will shorten the route whenever source route available may be shortening similarly to the conventional DSR. However, in En-DSR the shortened routes will be stored in the route cache as a reference. When most stable route in the network is failed to be identified, shorter path will be used in effort to be adaptive in routing over the highly dynamic ad hoc network. This allows shorter route to be selected when no reliable routes can be found in the ad hoc network.

3.3.2.2 Salvaging in En-DSR in Route Maintenance

When a node received an unknown Flow Label value without the source route attached in the packet, it will generate a Route Error (RRER) packet of UNKNOWN_FLOW type. Information of the destination and error destination will be included into the RRER packet to be returned to the previous node. No salvaging with route cache will be used in this En-DSR as it routes data packets over reliable path within the route cache. Rerouting with routes from the route cache is not efficient since route selection has selected the reliable path to route the data initially. Therefore, in avoidance of route usage which might shift between most stable and shorter paths over the time in salvaging that will cause lower network performance, salvaging is not enabled. Hence, no salvaging with route cache whenever a failed packet delivery encountered.

3.4 Conclusion

This chapter described the mechanism of the En-DSR algorithm where it enhances the DSR structure together with involvement of QoS provision considered for routing process in ad hoc network. In a topology where some nodes are dynamically moving with unpredictable manners, En-DSR route discovery is triggered only when there is demand on transmission across the ad hoc network which reduces the constant flooding of information in the network. Unlike proactive routing protocols that require periodic information updates, En-DSR only sends out requests when transmission is needed. Under an ad hoc network where some mobile nodes are moving actively, proactive routing protocols required a significant amount of information for the maintenance whenever mobile nodes traverse. En-DSR uses route cache efficiently with stability value that provides a relatively stable route for each transmission. With stable route in the transmission, QoS will be provided to the end users on requesting applications especially real-time applications that are demanding high performance. Moreover, QoS is improved by the resources allocation for the real-time applications via Flow Label in IPv6 IP packet header instead of relying on IPv4 “best-of-effort” services in the ad hoc network.

CHAPTER 4

IMPLEMENTATION RESULTS AND ANALYSIS

4.1 Introduction

En-DSR algorithm is tested for their performance in the mobile ad hoc network under different scenarios. Simulations are carried out using Network Simulation (NS-2) version 2.34 (Fall & Varadhan, 2009) to create mobile nodes connecting each other wirelessly with bidirectional links. Each of the nodes is considered as host where they react like a forwarding unit (e.g. router) as they mutually agreed to forward data packets on behalf of nodes in within the mobile ad hoc network. NS-2 is widely used for simulating wireless and wired networks, TCP routing, multicast protocols and so forth(Issariyakul & Hossain, 2009).

4.1.1 Network Simulator 2

The development of NS-2 simulator is supported by The Defense Advanced Research Projects Agency (DARPA) through the Virtual InterNetwork Testbed (VINT) project at Xerox Palo Alto research Center (PARC), Information Sciences Institute (USC/ISI) of the University of Southern California and currently collaboratively supported by National Science Foundation (NSF) with Collaborative Simulation for Education and Research (CONSER). It is an object-oriented, discrete event-driven simulator written in C++ language and OTcl (Tcl - Tool command language) script language with object-oriented extension developed by MIT (Massachusettes Institute of Technology). NS-2 simulation runs under Cygwin application where it works for Microsoft Windows. It is a Unix-like environment and command-line interface with native integration of different Windows-based applications, software tools and other resources. In addition to NS-2 simulator, bash shell and text utilities such as awk are used to run the simulations as well as collecting and reporting data extraction.

Reviews of the MANET simulation under NS-2 is done substantially, simulations are done with the closest of imitation to the real system. Simulations are done repeatedly for the closest relative comparison under same environment and parameters. Evaluation of performance has been done repeatedly in multiple simulations. In Chapter 2 literature review as discussed, current implementation on QoS in mobile ad hoc network under IPv6 Flow Label provision, extended version of QOLSR (ext-QOLSR) by Agha and Badis(2004)proposed to utilize the Flow Label in the IPv6 packet header. Thus in implementing En-DSR for a performance comparison, ext-QOLSR is used in comparison with this newly proposed En-DSR. Simulations are carried out in different scenarios and traffic patterns under the same environments.

4.2 Simulation Parameters

Simulations are carried out in 6 different scenarios under the same connection Constant Bit Rate (CBR) pattern file that defines the connections generated and sorts. Environment parameters are configures accordingly to have same environments configuration for the simulations.

4.2.1 Node Configurations

All nodes in the network are simulated with different settings as stated in Table 4.1. Settings in the simulations are made with some default values pre-set. Under NS-2 simulator, ad hoc routing protocol can be chosen with their respective interface queue type. In using En-DSR ad hoc routing protocol, Interface Queue Type CMUPriQueue is used whereas ext-QOLSR used Queue/DropTail/PriQueue settings. En-DSR makes use of the CMUPriQueue type to give priority to En-DSR in queuing the packets in each of the mobile node according to the routing

protocol. Topology size is fixed for all simulations with fixed number of nodes deployed in the network. Every simulations run for 300 seconds under different scenarios of topology and traffic patterns.

Table 4.1: Node configured settings

Settings	Value
Channel	Channel/WirelessChannel
Propagation	Propagation/TwoRayGround
Network Interface	Phy/WirelessPhy
Ad Hoc Routing	DSR / En-DSR / ext-QOLSR
Mac Type	IEEE 802.11
Interface Queue Type	Queue/ DropTail/ PriQueue/ CMUPriQueue
Antenna	Antenna/ OmniAntenna
Propagation	Propagation/TwoRayGround
Topology	1000m x 1000 m
Mobility Model	Random Waypoint
Node	50
Simulation durations	300 s
Propagation Delay	4 us
Basic rate	2 Mbps
Data rate	11Mbps

4.2.2 Traffic Patterns

Traffic connection comprises Constant Bit Rate (CBR) packets that are generated by the traffic generator script with a CBR traffic generator agent (Fall & Varadhan, 2009). Transmission Control Protocol (TCP) or CBR connections are implemented between the wireless mobile nodes where the traffics' parameters are predefined respectively. Traffic generated in the simulations is data packet type of CBR where CBR applications are defined to be real time and required QoS guaranteed. Traffic settings are defined as stated in Table 4.2. Total connections generated at specific time are random between the whole simulation duration at different starting time at the rate of 4 packets per second. Each of the packet size is 512 bytes and in the simulation each transmission can send up to 10000 packets. Whole CBR traffic patterns can be found in Appendix A.

Table 4.2: Traffic Settings

Settings	Value
Type of Connections	CBR
Number of nodes	50
Random Seed	1.0
Maximum number of connections that are allowed	40
Sending rate (packets per second)	4
Packet size	512 bytes
Maximum packet of payloads	10000

4.2.3 Scenarios Files

Node movements are generated by setdest script to create movements for each node within MANET for the simulations. This script uses the random waypoint algorithm where the algorithm is a random-based model mostly used for the mobile communication systems in mobility management schemes as it describes the movement pattern of mobile users, location, velocity and acceleration change over the duration of time (Greis, 1995).

Different initial positions are generated along with random time movements. Speed and direction of movements are randomly generated on each distinct node. In the simulations, six different categories of node movements are generated which are 20% mobile nodes, 40% mobile nodes, 60% mobile nodes and 80% mobile nodes movement. In all random movement scenarios, all nodes are randomly located and it will be moving during the whole simulation randomly in different directions. In the rest of the simulations, maximum speeds are set accordingly based on Table 4.3. All these four categories are simulated to test En-DSR routing where all nodes within the ad hoc network have highly dynamic movement where not all nodes are mobile at one time. It is to show with nodes that are statically located will be able to provide a more reliable route for transmission. Scenarios simulated are made according to below categories:-

- 20% mobile nodes

  20% of nodes within the network topology have 10 nodes mobile according to the pre-generated scenario file while the rest of the nodes remained statically located at their positions during the simulation.

- 40% mobile nodes

  40% of nodes within the network topology have 20 nodes mobile according to the pre-generated scenario file while the rest of the nodes remained statically located at their positions during the simulation.

- 60% mobile nodes

  60% of nodes within the network topology have 30 nodes mobile according to the pre-generated scenario file while the rest of the nodes remained statically located at their positions during the simulation.

- 80% mobile nodes

  80% of nodes within the network topology have 40 nodes mobile according to the pre-generated scenario file while the rest of the nodes remained statically located at their positions during the simulation.

Table 4.3: Scenario settings

Settings	Value
Pause time	0.0
Speed Range	1m/s to 10m/s (Uniformly distributed)
Number of mobile nodes (%)	20, 40, 60, 80
Number of scenarios	100

Transmitting data across a wireless ad hoc network is done via the routing process where each intermediate node will forward the data to their respective destinations. However, nodes move from one position to another at unpredictable time and routing process is required to include estimation on nodes' movement. In this research, random mobile speed is taken from a uniformly distributed range between 1m/s to 10m/s. Besides, different numbers of mobile nodes are simulated to move at one time and therefore five categories are being carried out repeatedly under different topologies and nodes distribution. Averages are calculated for the performance parameters in all simulations in order to get the best estimation values.

4.3 Simulation Results

Simulation results will show efficiency of the proposed routing algorithm in improving the network performance and thus, network performance in wireless ad hoc network has to be measured. Real-time applications demand high network performance as QoS needs to be provided to their transmission over the wireless and mobile ad hoc network. Multimedia data transmissions have to be delivered without much delay across the network and nonetheless delivering the exact data transmitted by the source. Hence, evaluation on the network performance has to done to show the routing efficiency. Evaluation will be done based on Packet Delivery Fractions (PDF), Normalized Routing Load (NRL), Average end-to-end Delay and Average throughput. Performance of the ad hoc networks will be able to be measured for the simulated scenarios.

4.3.1 Packet Delivery Fraction (PDF)

Packet Delivery Fraction percentage is the measurement on total number of CBR data packets delivered to destination with the total number of packets generated by the source node as show in Formula 5

$$Packet\ Delivery\ Fraction\ (\%) = \frac{total\ no.of\ packets\ delivered\ to\ destination}{total\ no.of\ packets\ generated\ by\ source} \times 100 \quad (5)$$

Data packets that are successfully received by the destination node is calculated and divided over the number of data packets sent. It shows the loss rate that happens in the transport protocols where it keeps track of the node mobility continuously for the data packet. The higher PDF value shows the better transmission as more data packets are successfully transmitted over to their respective destinations. This improves the ad hoc network performance.

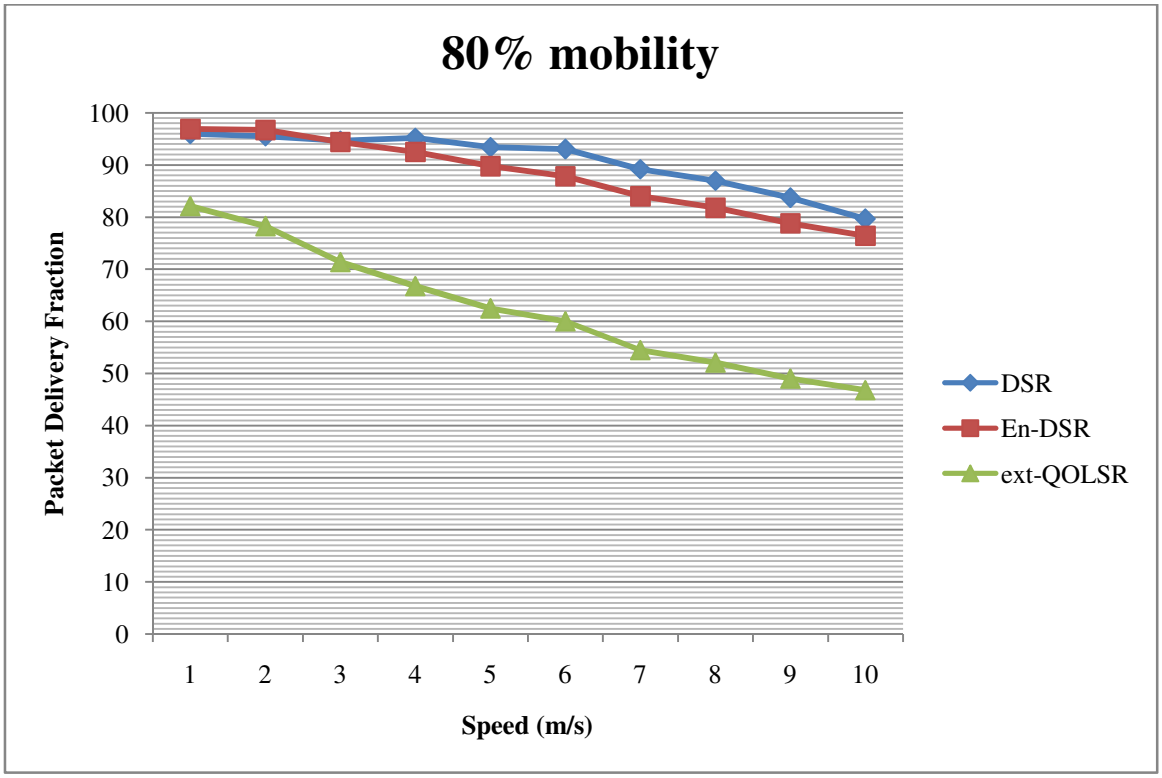

Figure 4.1: Packet Delivery Fraction for 80% mobile nodes

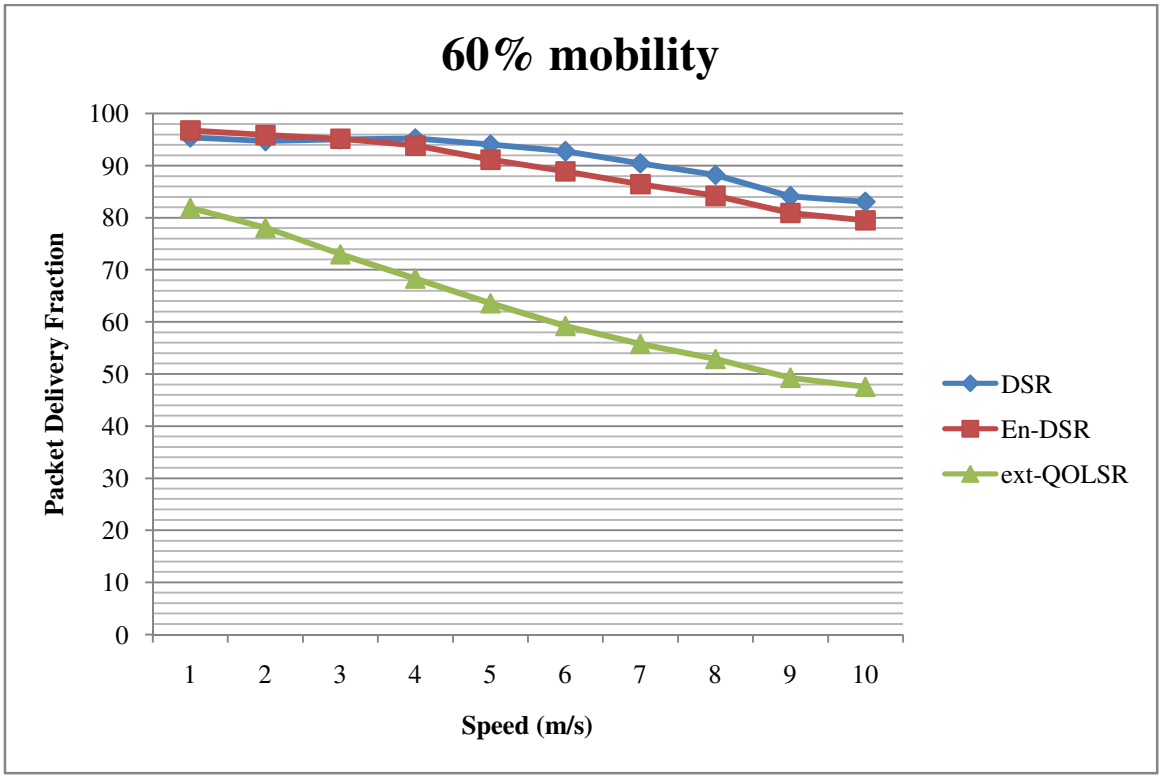

Figure 4.2: Packet Delivery Fraction for 60% mobile nodes

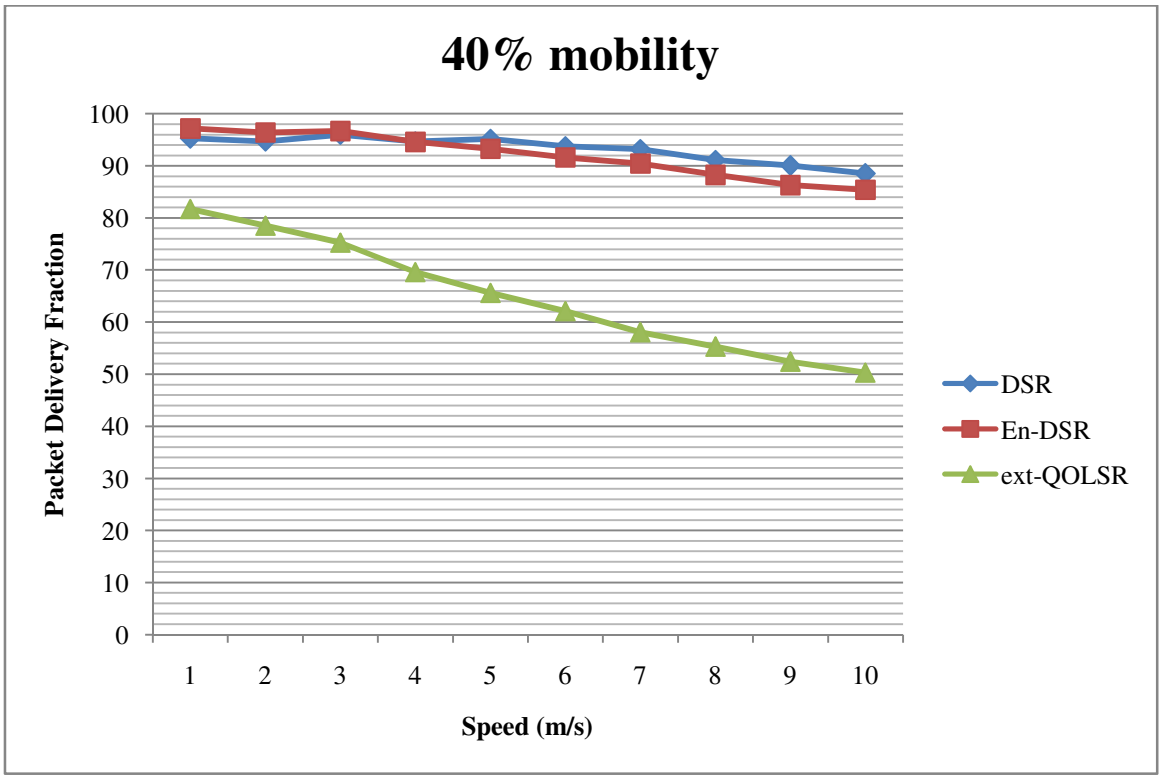

Figure 4.3: Packet Delivery Fraction for 40% mobile nodes

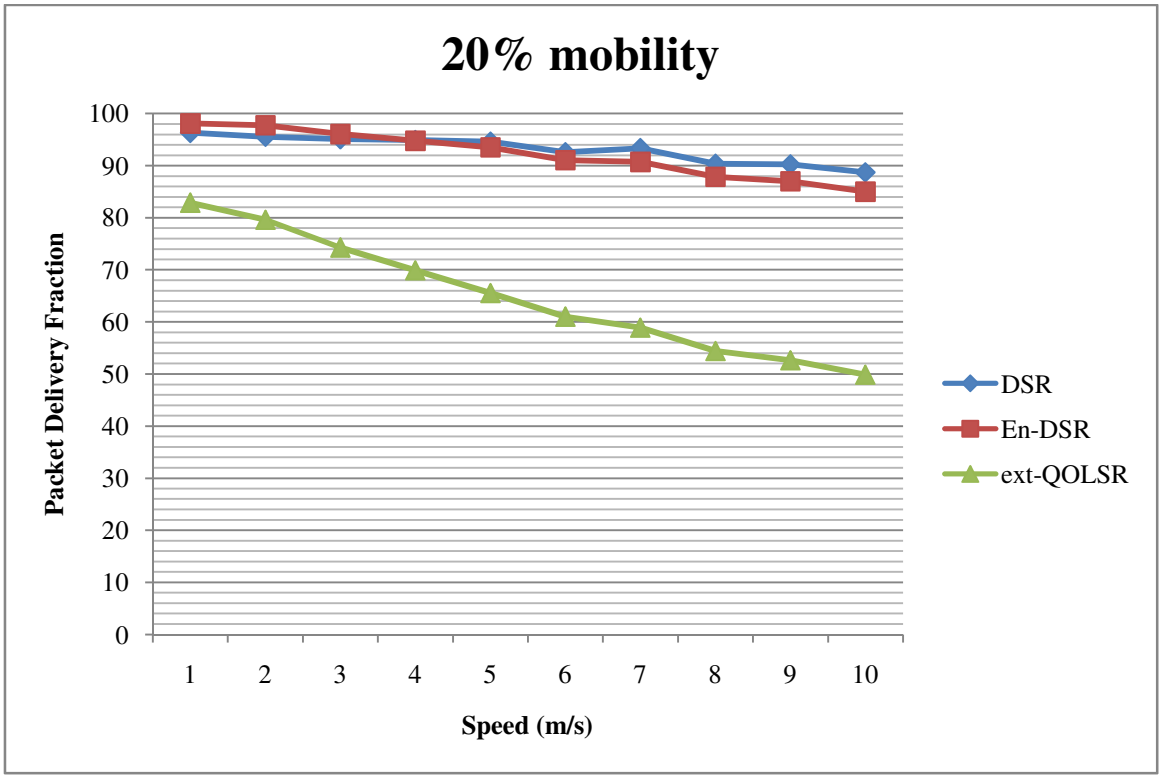

Figure 4.4: Packet Delivery Fraction for 20% mobile nodes

In Figure 4.4, the initial performance of En-DSR showed a better PDF as compared to the other two routing protocols. Even though performance in En-DSR decreases as the speed increased, it performed as close as the original DSR. From speed range of 1 m/s to 4 m/s, En-DSR improved in the highly dynamic ad hoc network as compared to the conventional DSR whereas ext-QOLSR showed a poor performance as the unpredictable node mobility affected the MPRs selections similarly in conventional OLSR as describe in section 2.2.1.1.3. Nodes that are selected to be the MPRs within the network could be the mobile node in the ad hoc network that brings down the performance in transmission which relied on the MPR nodes. As the speed increases from the point of 5m/s onwards, all routing algorithm performances deteriorated as shown in Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4.

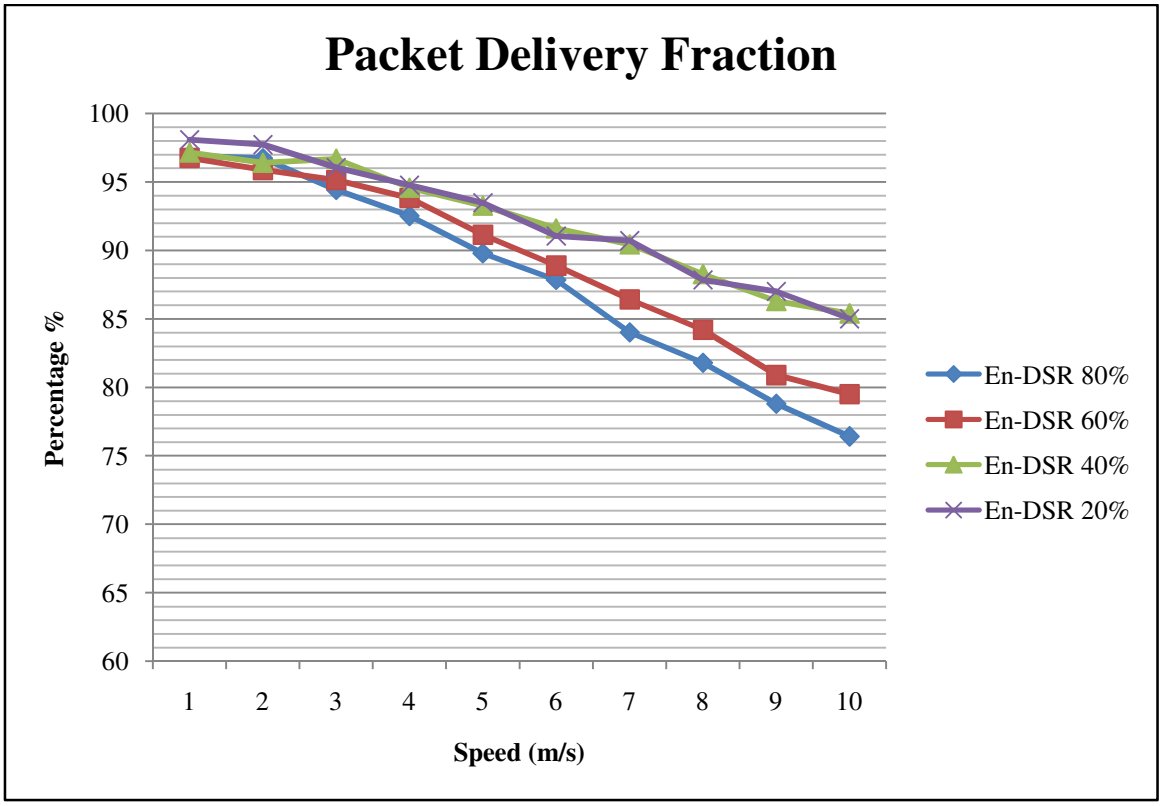

Figure 4.5: Packet Delivery Fraction for En-DSR as speed increases.

Figure 4.5 consolidated packet delivery fraction simulated in all mobility percentages. In En-DSR, node's stability showed a higher sum of stability value that reflected the unstable routes condition. Shortest path is used in the scenarios where no routes are acceptable in stability within the highly dynamic ad hoc network. In comparison between 20% mobility that has higher number of statically located nodes to 80% mobility that has lower number of statically located nodes, En-DSR can perform better up to 1.2% better whereas in ext-QOLSR improvement is 0.8% with the same mobility percentages. Initial speed range shows better results in PDF as compared to the rest of the protocols. With a certain number of statically located nodes in the dynamic ad hoc network, reliable routes are better constructed for the unpredictable networks. As speed increases, route changes are more often encountered in the ad hoc network transmission.

4.3.2 Normalized Routing Load (NRL)

Total number of routing packets sent per data packets that are sent successfully at their destinations over the network is the normalized routing load as show in Formula 6.

$$Normalized\ Routing\ Load = \frac{total\ no.of\ routing\ packets\ sent}{total\ no.of\ data\ packets\ received} \quad (6)$$

Routing packets for the routing mechanism is divided over number of packets received by the node is calculated as the NRL. Number of routing packets will be measured where the routing packet is sent or forwarded to their next neighbor within the ad hoc network. This measurement indirectly shows the efficient of the network protocol as lower NRL shows less routing packets used for a data packet delivery to its respective destination over the dynamic wireless ad hoc network. For the four different categories of mobility in the simulations, results are plotted as shown in Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9.

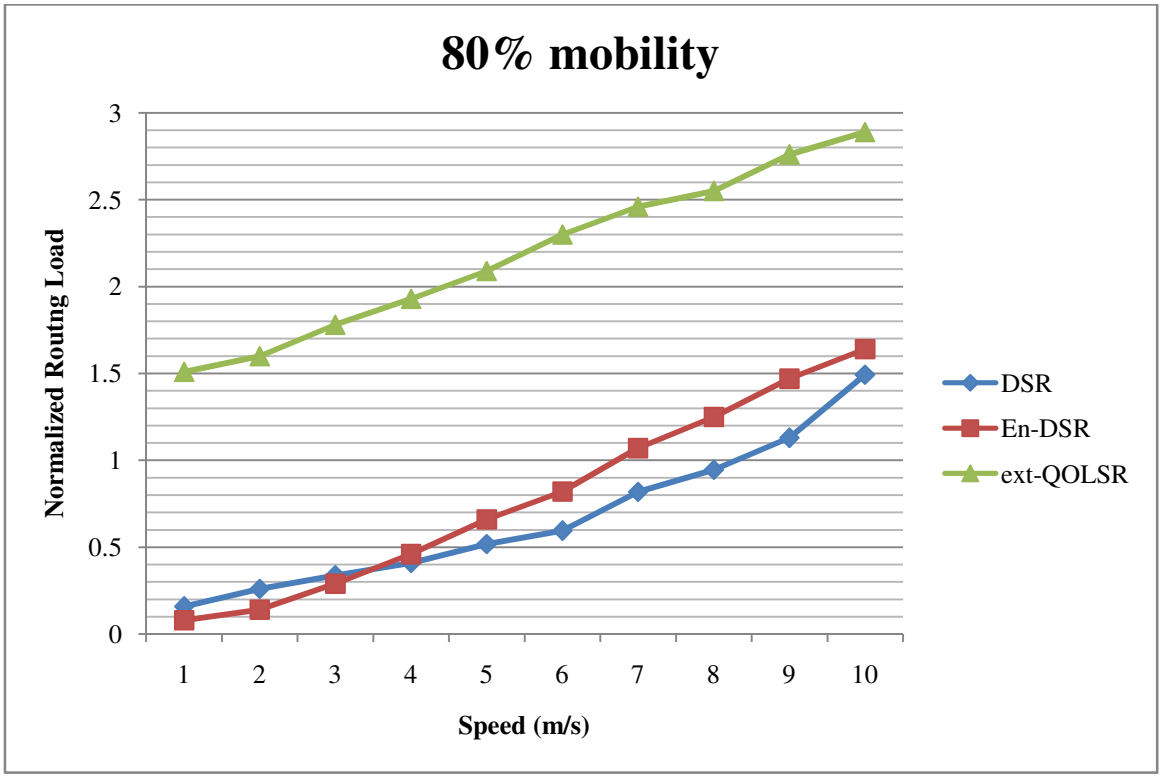

Figure 4.6: Normalized Routing Load for 80% mobile nodes

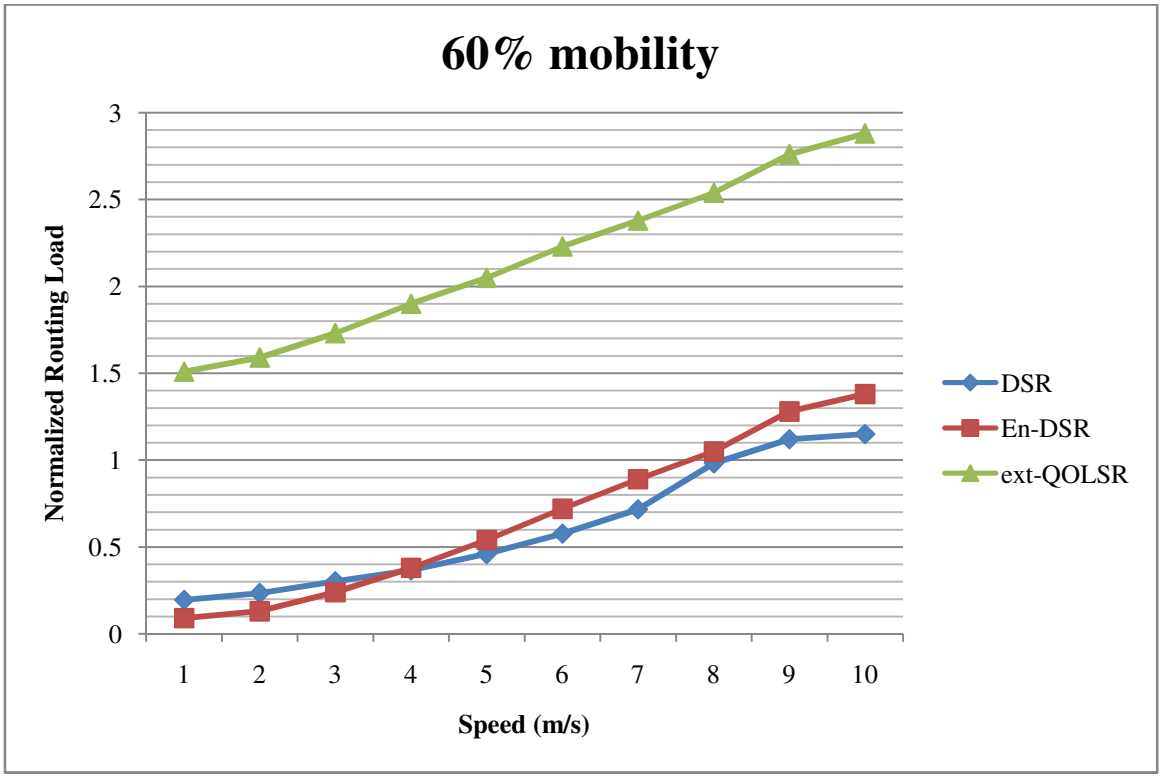

Figure 4.7: Normalized Routing Load for 40% mobile nodes

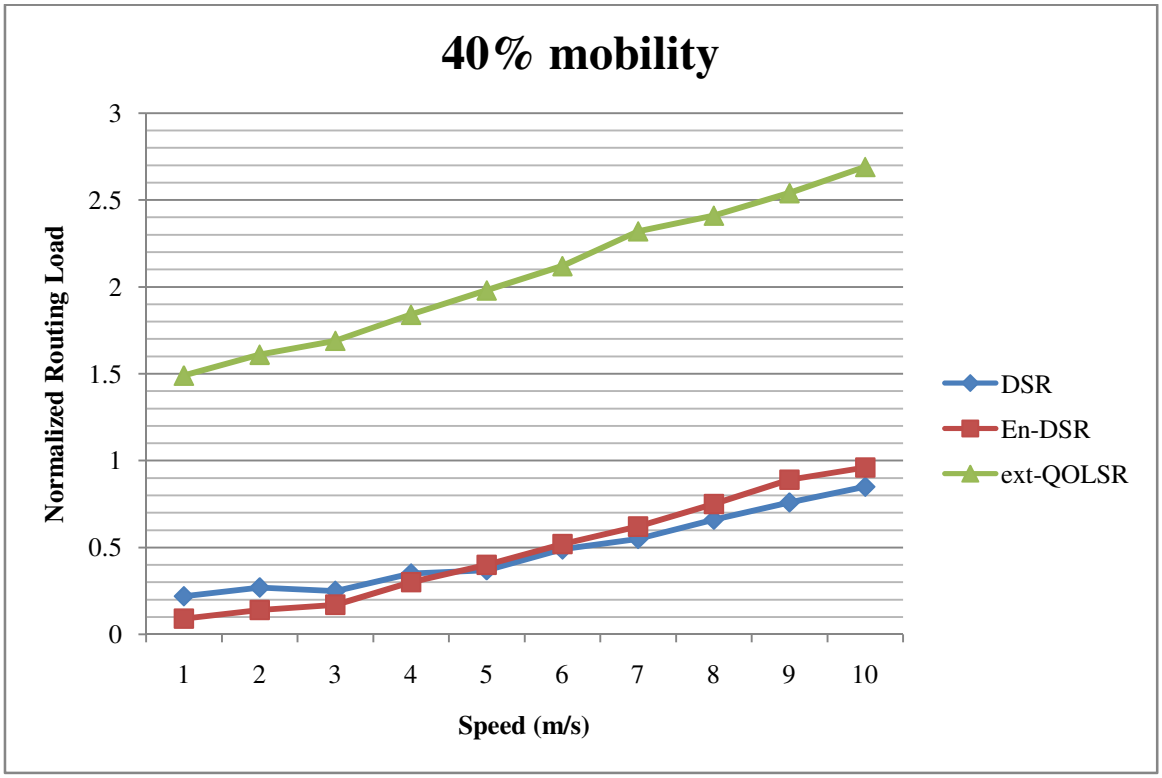

Figure 4.8: Normalized Routing Load for 40% mobile nodes

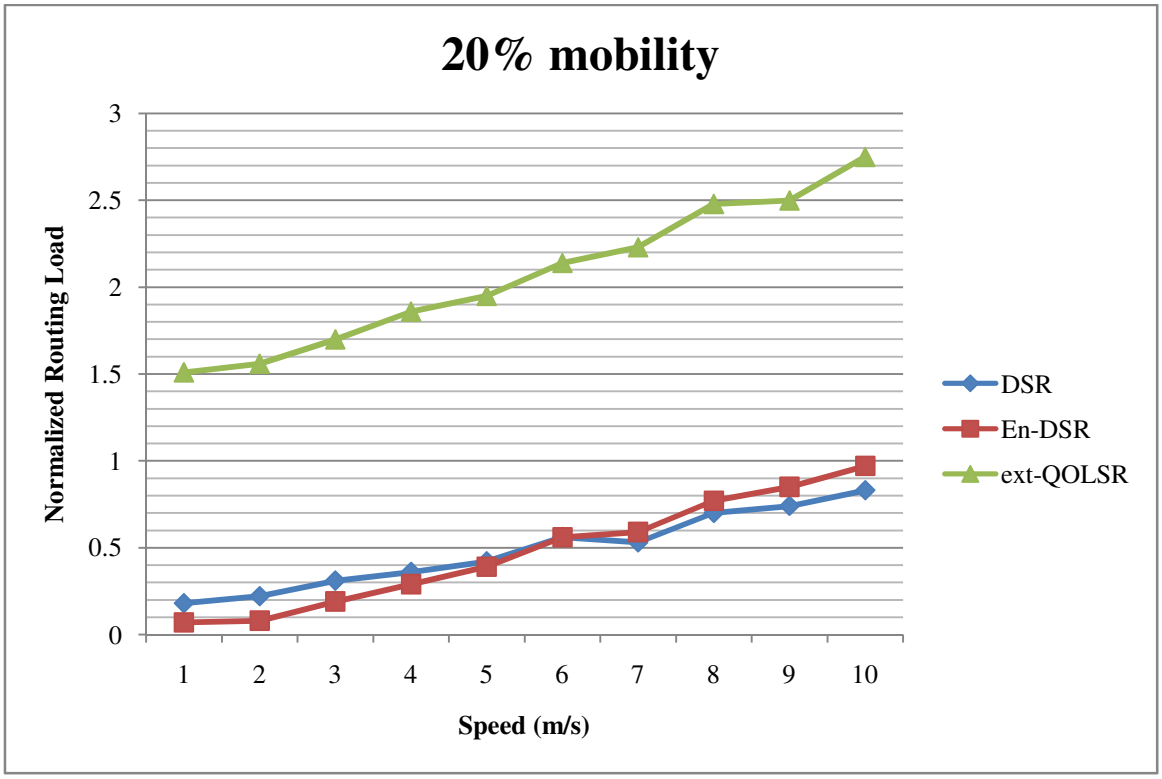

Figure 4.9: Normalized Routing Load for 20% mobile nodes

As table-driven routing protocol, ext-QOLSR involved an amount of control packets in the ad hoc network in order to maintain their connectivity frequently. As the topology changes, more information will be shared among the nodes in order to keep up with the changes. With the updates, route for transmission can always be found in their routing table but as seen in those graphs above, source consumption has to be considered. Routing packets involved in the route discovery and route maintenance mechanisms in DSR, En-DSR and ext-QOLSR are to discover a potential route for transmission to take place. With the involvement of routing packets, more resources such as bandwidth and higher priority in forwarding to the next hop are used up. Initial performance presented a lower routing packets needed by En-DSR and along the increasing speeds, it performed as close as the original DSR in routing across the ad hoc network.

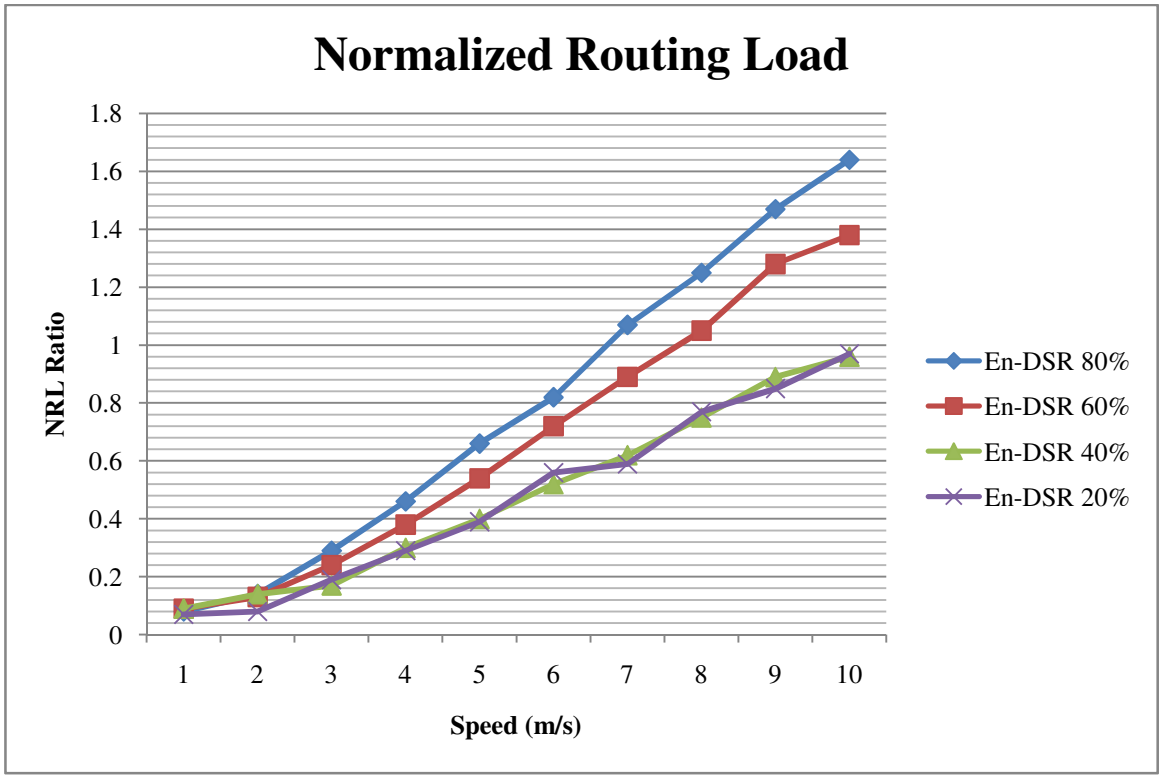

Figure 4.10: Normalized Routing Load for En-DSR as speed increases

As speed increases, Figure 4.10 shows the results of En-DSR normalized routing load increases. Nodes that move rapidly have increased the number of routing packets in the routing process but low speed showed less routing packets are transmitted across the network. Initially, En-DSR required a lower volume of routing packets where the stable routes have less possibility to be reestablished due to the link breakages. As the speed increased, amount of control packets sent to the network increased too. This is due to the increasing speed of mobile nodes in the ad hoc networks that induced higher link breakages. Much less control packets can be seen for both DSR and En-DSR as both are on-demand routing protocols that only initiates discovery when there is request for transmission.

4.3.3 Average End-to-end Delay

Average end-to-end delay is one of the critical parameters measured in the MANET where the delay is influenced by the rate of Constant Bit Rate (CBR) packets. Total delay experienced is divided by the number of packets arrived at the destination in Formula 7.

$$Average\ end-to-end\ delay = \frac{(transmission_{delay} + propagation_{delay} + processing_{delay})}{no.of\ packets\ received} \quad (7)$$

In a transmission, delay can take place in different stages such as buffers which can be quickly filled up while waiting to be delivered across a congested network. Hence, at each of the transmission minimum end to end delay is most preferable. Real time applications will require lowest possible delay to provide the best transmission to the end users. Results are illustrated graphically in Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14.

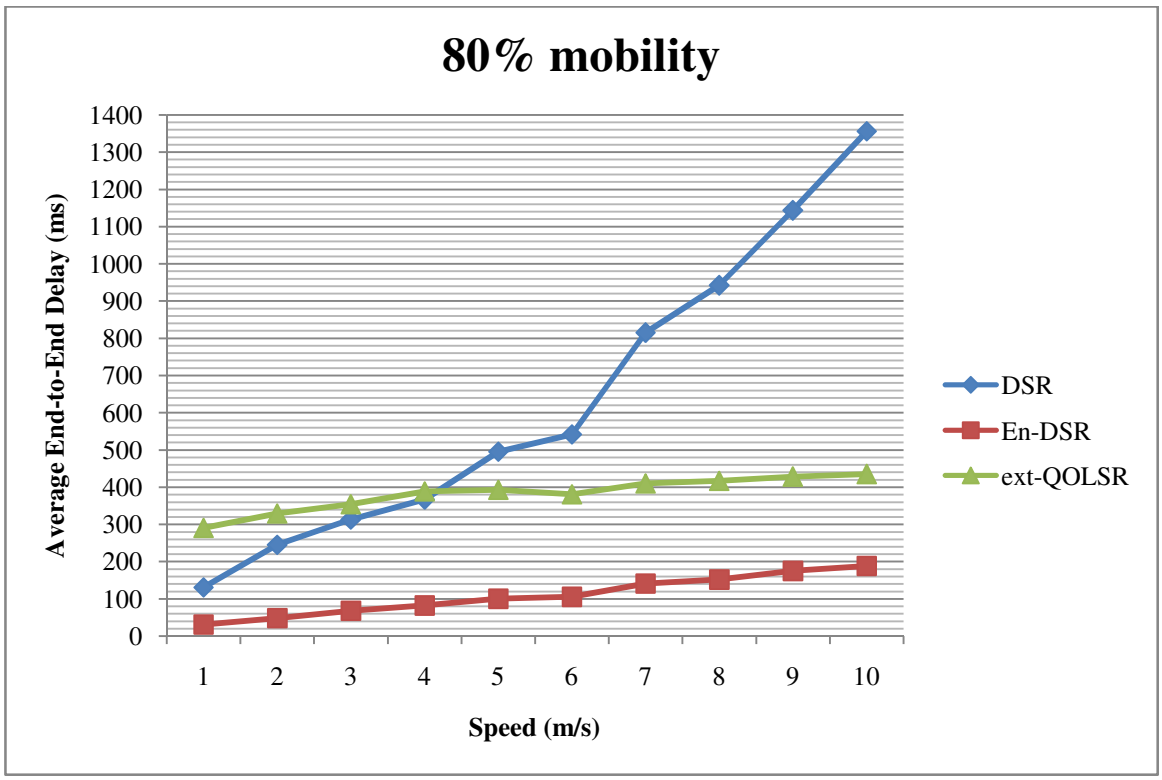

Figure 4.11: Average End-to-end Delay for 80% mobile nodes

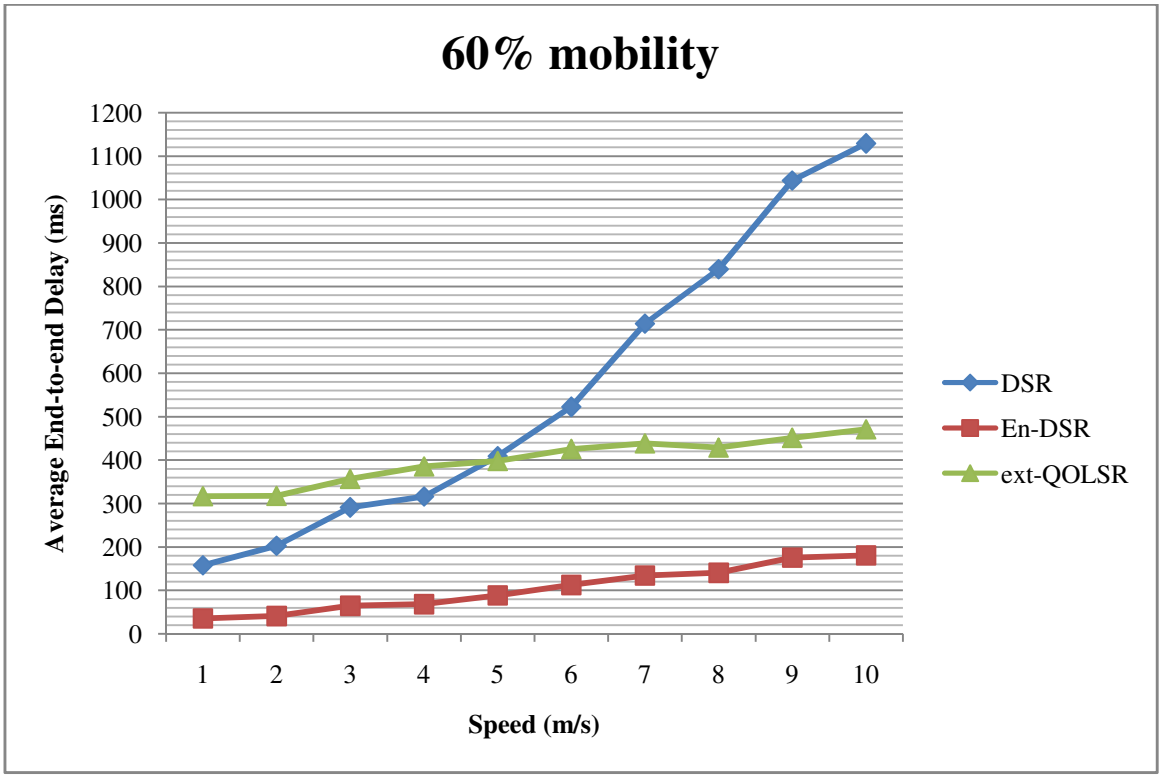

Figure 4.12: Average End-to-end Delay for 60% mobile nodes

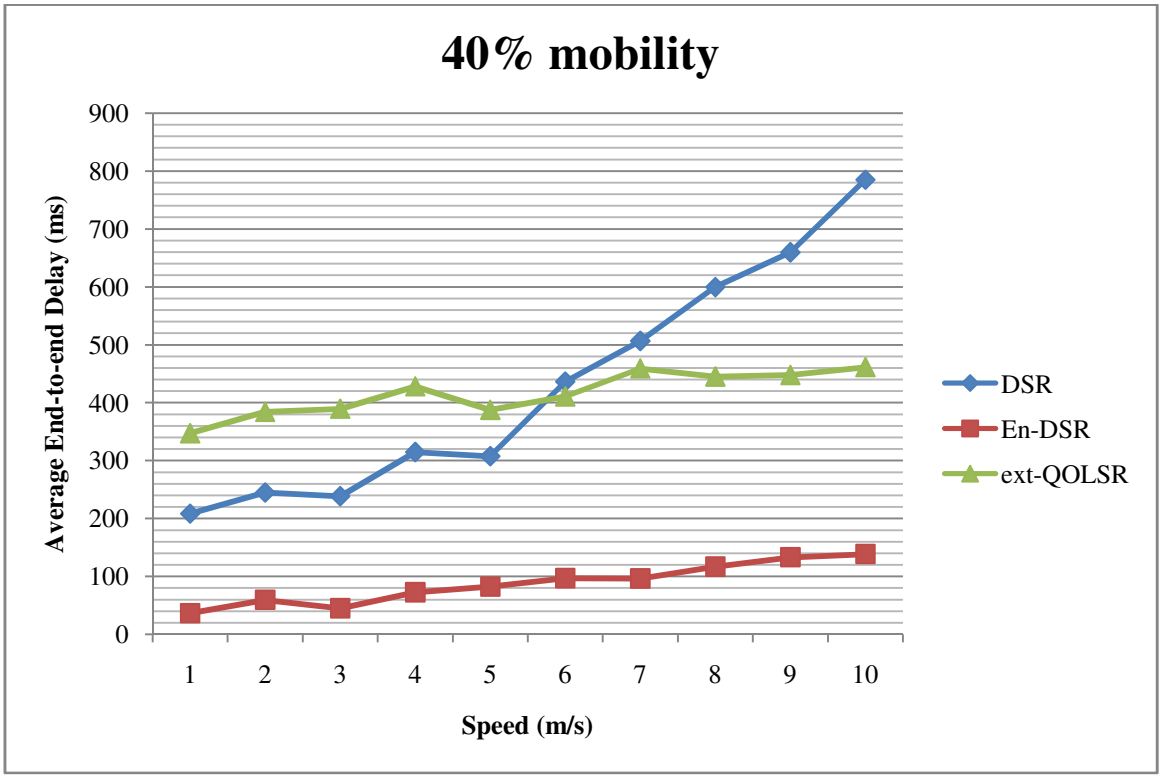

Figure 4.13: Average End-to-end Delay for 40% mobile nodes

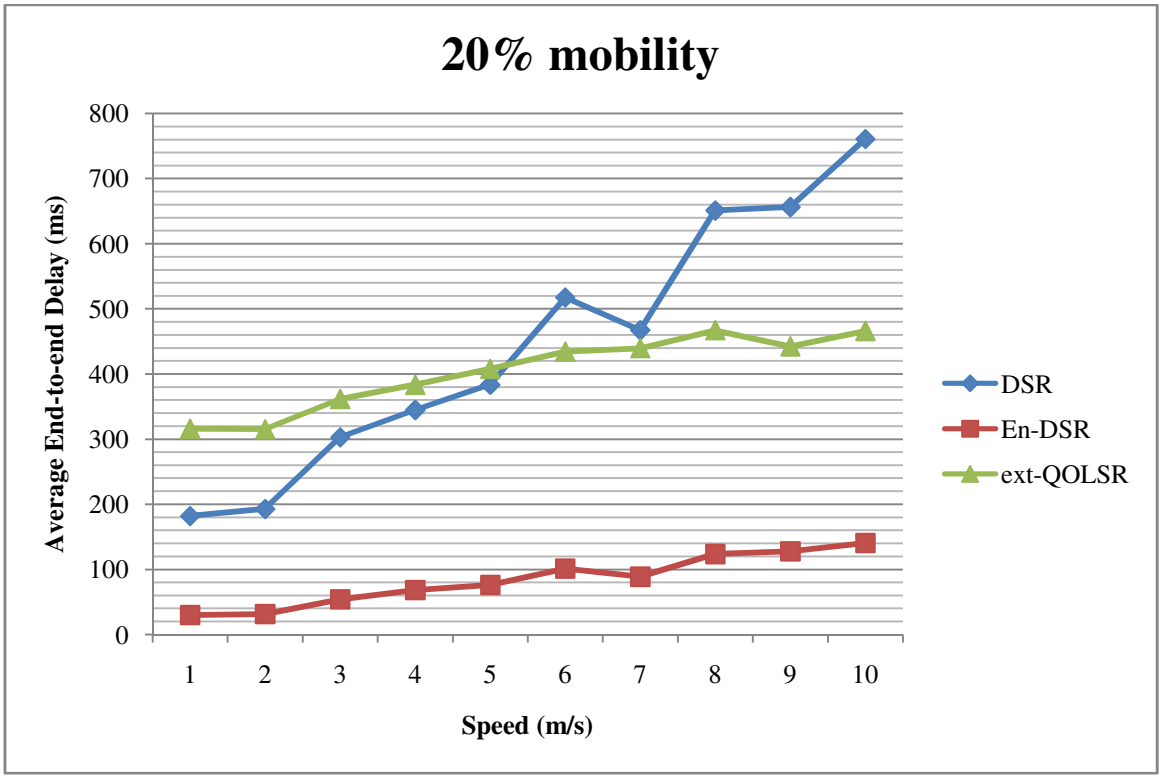

Figure 4.14: Average End-to-end Delay for 20% mobile nodes

On average, end-to-end delays for all routing protocols above are affected as the speed of mobile nodes increases. However, with IPv6 provision in QoS for the transmission in En-DSR, the particular routing performed better as compared to the rest. In treating all packets under the same Flow Label, En-DSR achieved lower end-to-end delay on all of the scenarios as shown above. 76% of improvement occurred under speed of 1m/s in En-DSR comparing the conventional DSR under 80% mobility. A significant improvement can be seen by comparing ext-QOLSR and DSR to En-DSR where the average end-to-end delays in En-DSR showed better results. Despites the changes in the topology on average En-DSR is able to route via the stable route and presented with lower average end-to-end delays. However, as the speeds increased ext-QOLSR performed better while comparing to DSR where the high node speeds increased the end-to-end delays in the network. Updated routing tables reduced the time of delay during the transmission and ext-QOLSR performed consistently as the speed increased. DSR initially has lower average end-to-end delay but as the speed increases, the changes in

the topology become more frequent. DSR that relies on the shortest path to their destination node encountered higher average end-to-end delay as the intervened nodes moved unpredictably over the time. On average, reserving resources on highly dynamic network required more effort as reserved or established flows are more prone in having link breakages. Overall, as the speed increases, the delay increases. Under such network environment where nodes mobile at unpredictable time and manners, En-DSR is able to perform on average a shorter end-to-end delay.

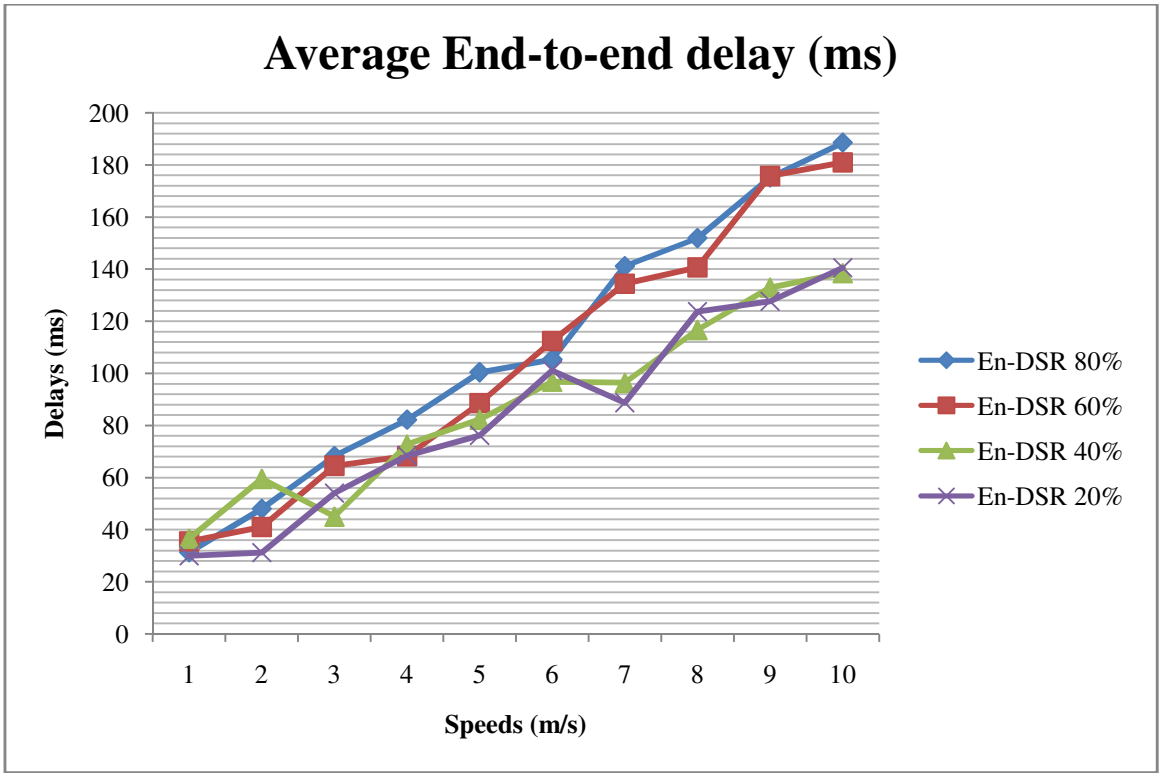

Figure 4.15: Average End-to-end delay for En-DSR as speed increases

Figure 4.15 showed all simulation in En-DSR under different mobility as speed increases. As speed increases, increasing control packets are relatively higher. As the traffic becomes dense, average end-to-end delay increases. In En-DSR, prioritized packets are treated with special handlings upon request via the IPv6 Flow Label. This utilized the designated Flow Label field in supporting better QoS network.

4.3.4 Link Throughput

Overall link throughput in the ad hoc network is measured by the rate of successful packet delivery for the whole duration of the simulation. Formula 8 shows the throughput formula.

$$Throughput\ (kbps) = \frac{total\ data\ rates\ delivered}{(stop_{time} - start_{time})} \times (\frac{8}{1000})$$

(8)

Higher throughput is favored as it shows a better transmission over the ad hoc network with successful deliveries across the highly dynamic ad hoc network. Results of the simulation are plotted in Figure 4.16, Figure 4.17, Figure 4.18 and Figure 4.19.

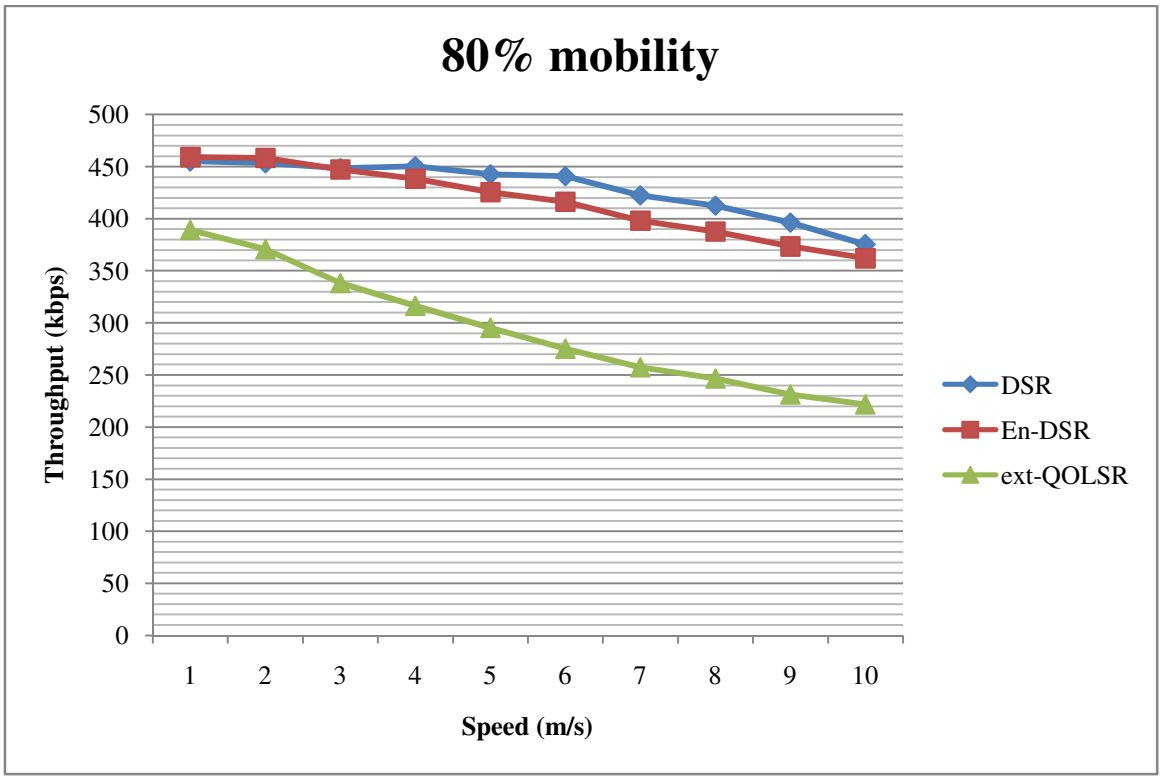

Figure 4.16: Throughput for 80% mobile nodes

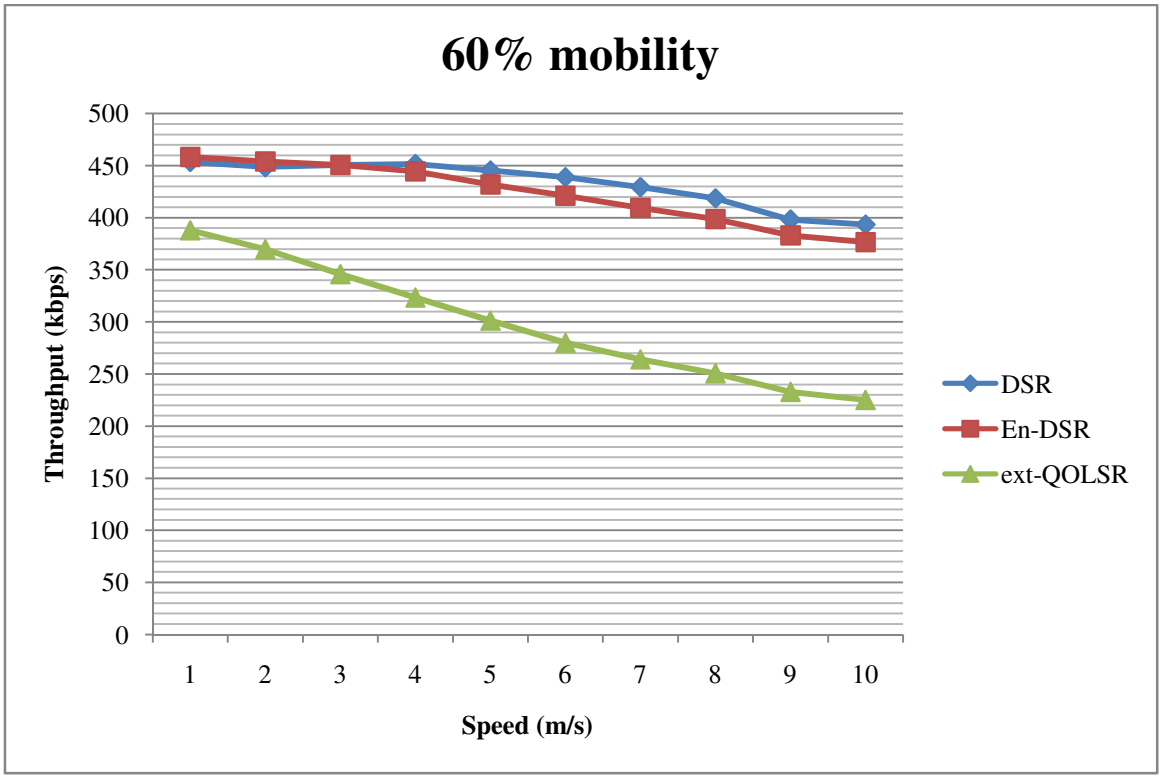

Figure 4.17: Throughput for 60% mobile nodes

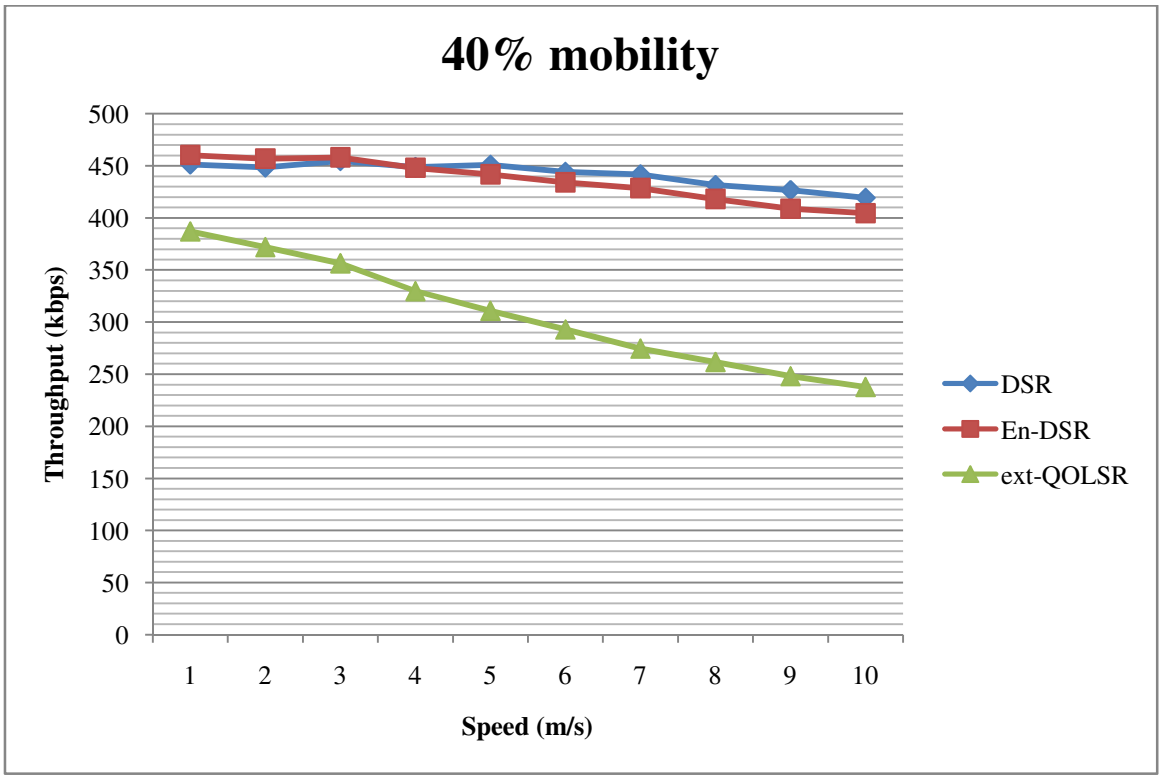

Figure 4.18: Throughput for 40% mobile nodes

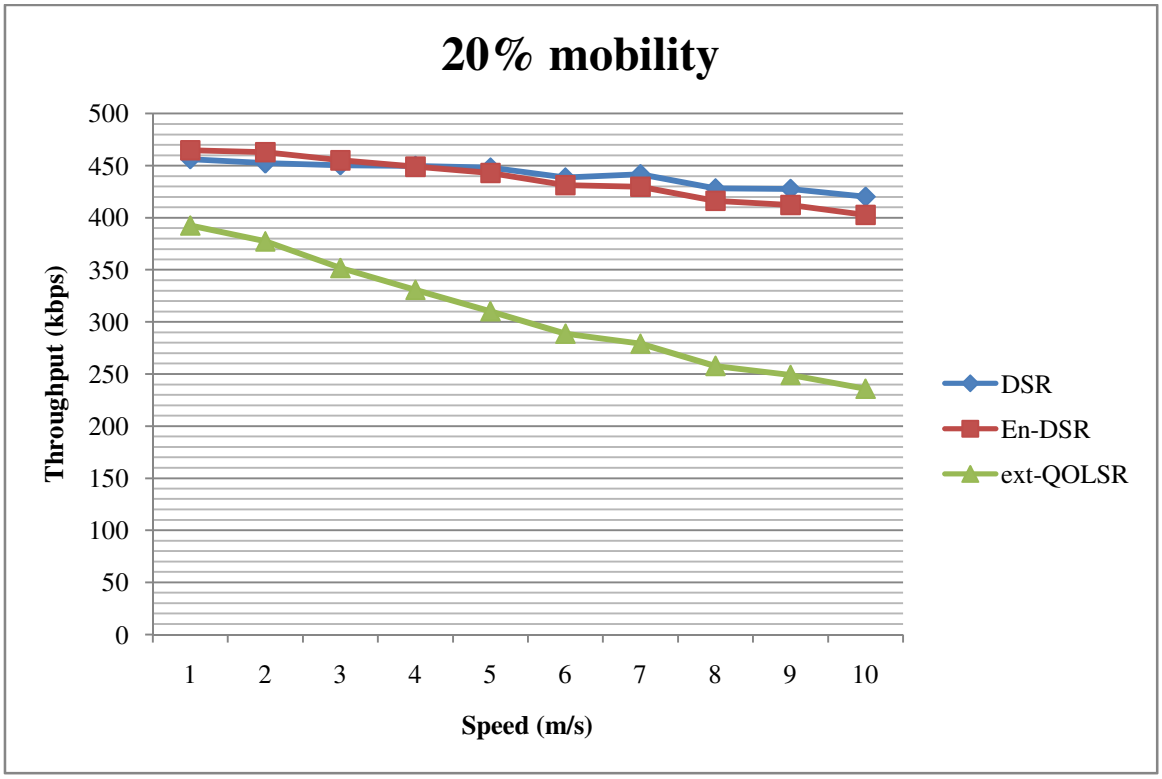

Figure 4.19: Throughput for 20% mobile nodes

Throughput in En-DSR achieved is better as compared to the other routing protocols for the speed range of 1 to 4 m/s and En-DSR performed as relatively well as DSR in the ad hoc network as speed changes. Overall, the throughput shows data packets that successfully delivered to their respective destinations decreases as speed increases. As successful packet delivered, throughput increased. On average in speed 1m/s for 20% mobility, En-DSR performed 1.8% better than conventional DSR and improved by 18.4% in comparison to ext-QOLSR. High mobile speeds in routing protocols brought in dynamic changes along increases in the network during the transmission that lowers the throughput. During simulation, different source nodes will establish their respective connections to carry out transmissions at different time. In Appendix A, total connections are 40 over 25 sources. Connections are randomly generated at different times. En-DSR performed as well as DSR with better performance at lower speed ranges. At lower speed ranges, less link disconnections are encountered. In En-DSR that favors stable nodes for transmission, En-DSR performed better .

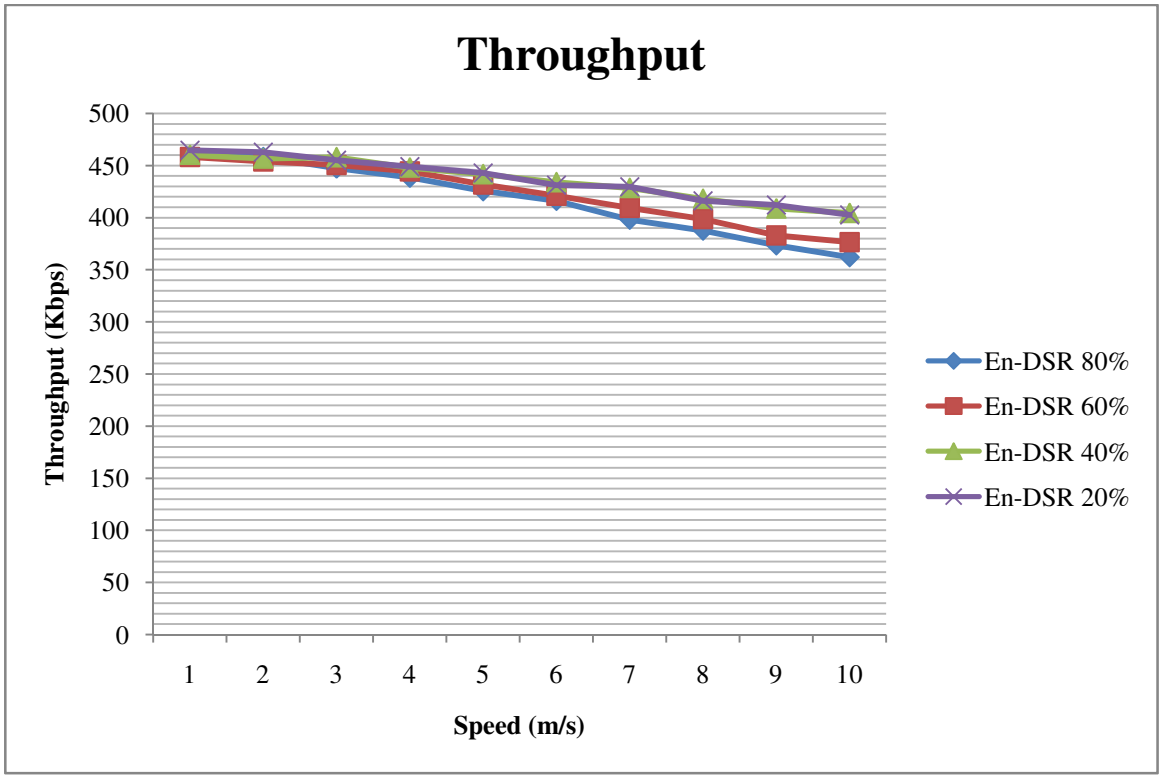

Figure 4.20: Throughput for En-DSR as speed increases

Throughput for all simulations with En-DSR under different speeds and node mobility are showed in Figure 4.20. As speed increases, average throughput decreases. With mobility involved that affects the topology, it increases the chances of tearing down on-going transmissions. Retransmissions are carried out in order to continue transmitting data across the ad hoc network. With lower mobility nodes, throughput is higher which indicated better transmission in the ad hoc network with increasing speeds.

4.4 Mobile Scenario

In this section, simulations of wireless ad hoc networks are randomly simulated over the duration of time where speed, direction and initial positions are randomly generated. Pause time is uniformly distributed to the maximum of 30 seconds. Scenario settings are stated in Table 4.4. Mobile scenarios are simulated to show performance of En-DSR and the rest of comparing routing protocol over a mobile ad hoc network where all nodes within will move at unpredictable time and do not stay at one location for long. Therefore, this section will have simulation where on average node have 30 seconds pause time at one location. They however do not remain the same location at all times. All nodes will have variety of directions and speeds. Over the mobile ad hoc network, dynamic scenarios with 50 mobile nodes are repeated with the similar configuration as stated in Table 4.3 under the settings stated in Table 4.4. The movement of all nodes within the network is randomly generated with the NS-2 software where it will simulate different topology deployments as well as the changes over the time. Results are compared accordingly below.

Table 4.4: All random scenarios settings

Settings	Value
Type of Connections	CBR
Number of nodes	50
Number of possible mobile nodes	50
Random Seed	1.0
Maximum number of connections	40
Sending rate (packets per second)	4
Packet size	512 bytes
Maximum packet of payloads	10000
Pause time	30.0 seconds
Speed	0m/s to 10m/s (Uniformly distributed)
Number of mobile nodes	50
Number of scenarios simulated	100

4.4.1 Results

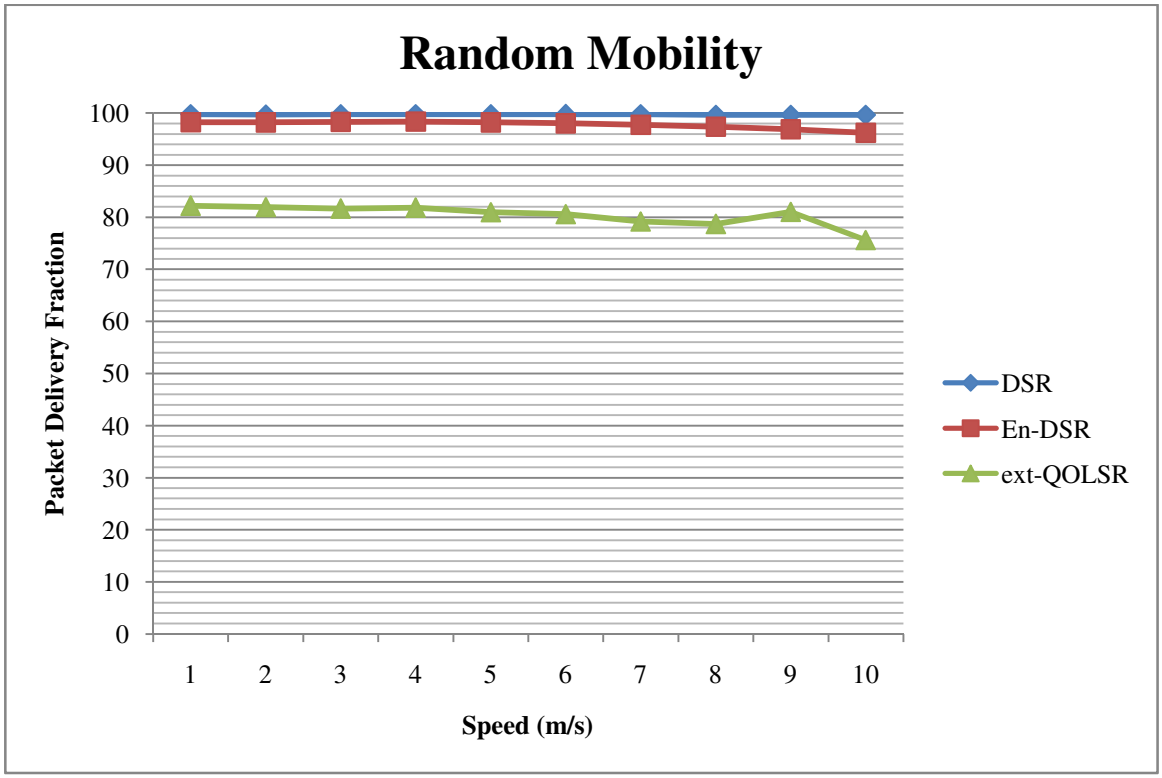

Figure 4.21: Packet Delivery Fraction for mobile scenarios

The initial performance has been maintained in En-DSR as close as the DSR routing protocol in Packet Delivery Fraction (PDF) as shown in Figure 4.21. It performed consistently as the maximum speeds changes. All nodes within the ad hoc network have the possibility of changing their respective position over time that changes the network topology. In DSR, salvaging and ARS optimizations are applied and therefore under such mobility this routing protocol still performs consistently and deteriorates slightly as speed increases. However, in this simulation each node will stay at one location for a defined pause time during the whole transmission. Different speeds run on randomized nodes with different direction as well as each node being able to stay and move after a defined time caused the stability value varies. En-DSR performed on average of 19.5% better in PDF as compared to ext-QOLSR in the all mobile scenarios and relatively as good as DSR in this case. In a dynamic network where all nodes have possibility of moving from one point to another position within the period of time,

ext-QOLSR showed lower PDF as selecting possible route required more information of the ever-changing network topology that eventually overwhelmed the routing process. Hence, in such a highly dynamic ad hoc network, topology changes so frequent En-DSR adopted the shortest path selection along with better stability value to adaptively route the data packet across the network. En-DSR performed as close as DSR in PDF in network where all nodes may be mobile at different time and directions.

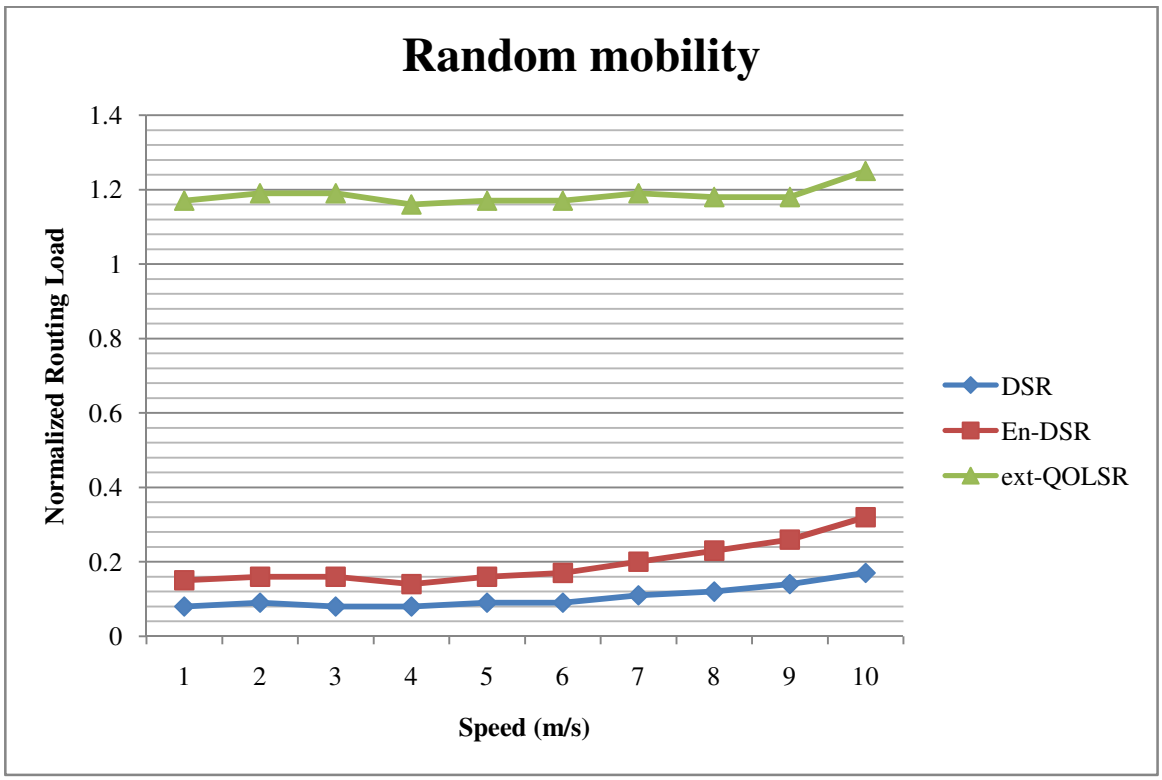

Figure 4.22: Normalized Routing Load for all mobile scenarios

En-DSR achieved a higher QoS by sacrifying a little on Normalized Routing Load as shown in Figure 4.22. Routing packets were used to get the latest information as changes were encountered in the highly dynamic ad hoc network. In the graph, it depicted ext-QOLSR has higher routing packets in comparison to En-DSR and DSR. Control packets in the all mobile scenarios are much higher in ext-QOLSR as the nature of the routing protocols required control packets in order to get all nodes in the network updated with the latest information. DSR and EN-DSR are relatively low in sending out control packets in comparison with ext-

QOLSR but En-DSR has slightly higher density of control packets in the all mobile scenarios than DSR in attempts of providing better QoS to the requesting applications.

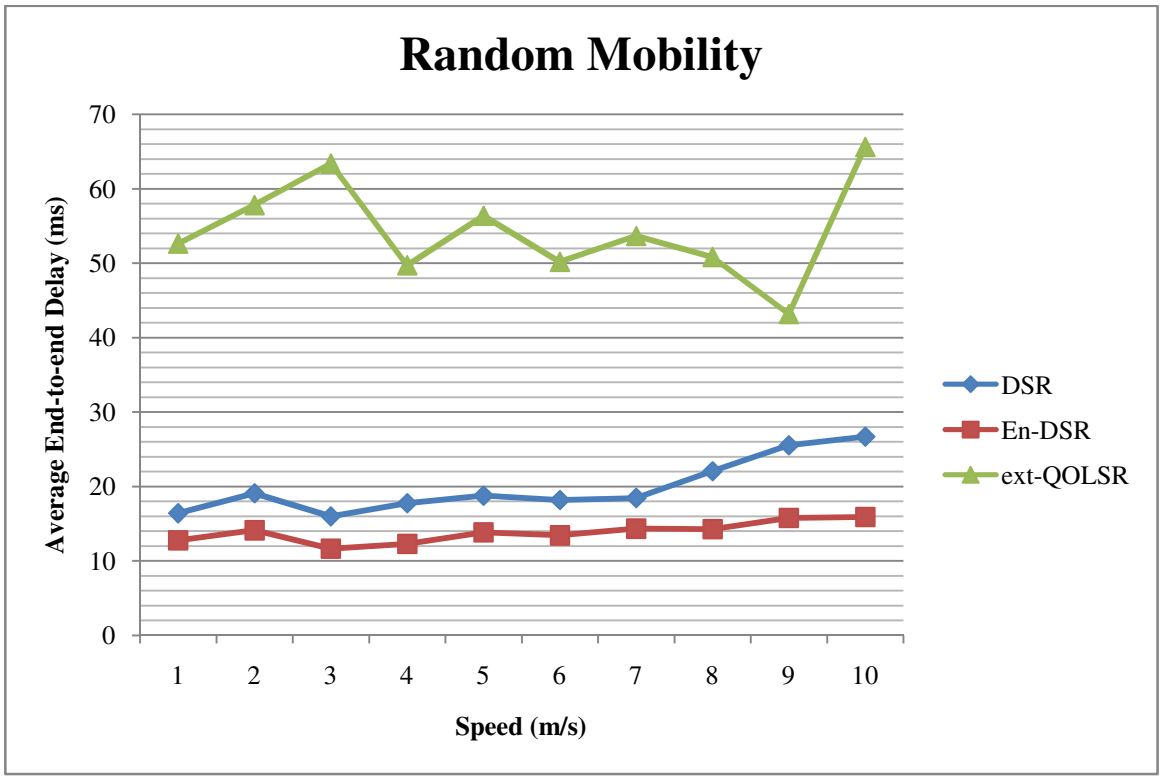

Figure 4.23: Average End-to-end delay (ms) for all mobile scenarios

The achievement in average end-to-end delay has made it pay off for the loss in performance in other aspect such as NRL as shown in Figure 4.22. In Figure 4.23, it presented ext-QOLSR with a higher average end-to-end delay in the highly dynamic network as compared to DSR and En-DSR. En-DSR performed better than all routing protocols as it included QoS provision in the transmission. Despite inconsistency showed for all routing protocols in the simulations as the speed increases, En-DSR maintained to have the lowest average end-to-end delay of all. As QoS provision is implemented in on-demand routing En-DSR, average end-to-end delays in ext-QOLSR and DSR are higher than En-DSR. Across the highly dynamic ad hoc network, En-DSR performed well on average end-to-end delay.

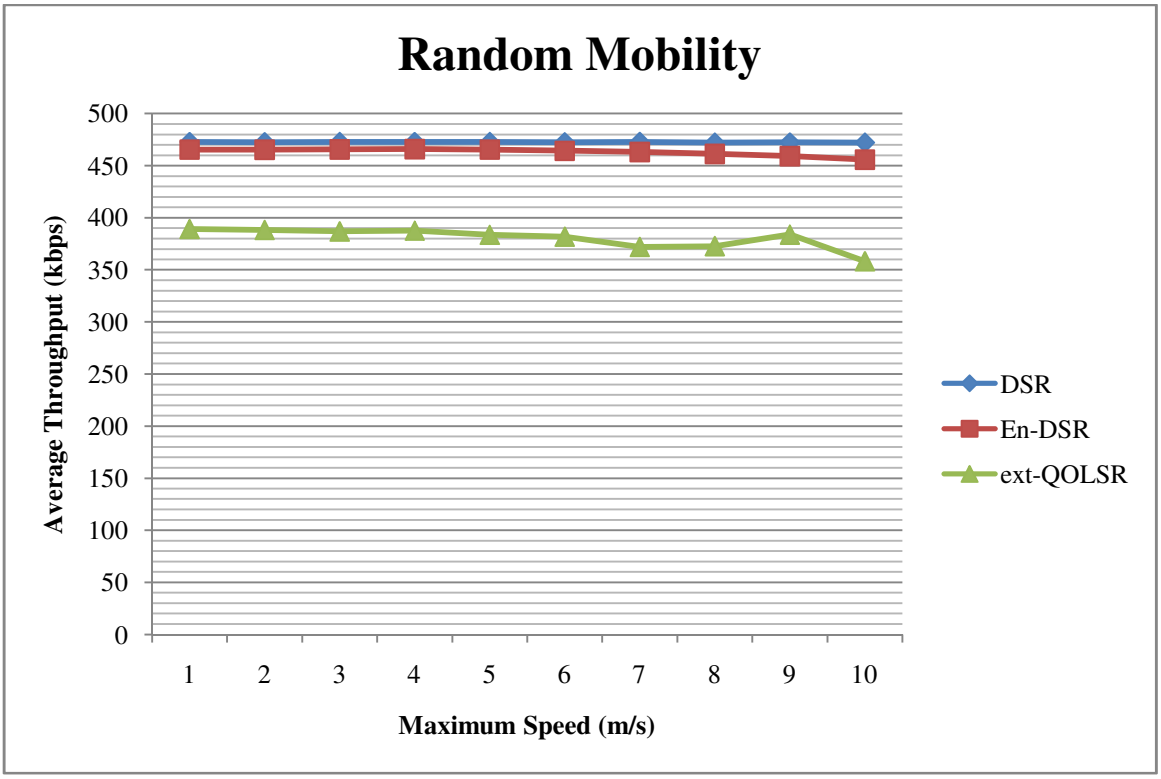

Figure 4.24: Link Throughput (kbps) for all mobile scenarios

The throughput has been maintained as close as the original DSR routing protocol as seen in Figure 4.24. It showed ext-QOLSR has the lowest average throughput in the simulated results whereas En-DSR and DSR gave a consistent performance even as the speed increased. In highly dynamic network, with all mobile nodes move at unpredictable manners En-DSR performed similarly to conventional DSR as En-DSR adaptively selected route to route the data across. Hence, in the absence of stable route En-DSR still performed relatively well as compared to DSR in this nature of mobile ad hoc network.

4.5 Conclusion

The performance in En-DSR is comparable to the original DSR as it performed as consistent in the simulations. Under lower speed ranges in the four categories of partial mobile density, En-DSR performed better than DSR and ext-QOLSR. With more stable nodes in relaying data packets En-DSR improves 1.2%. In 80% mobility density, En-DSR PDF performs around 1% better than conventional DSR and 15.2% better than ext-QOLSR. While in 20% mobility density, En-DSR PDF performs around 2% better than DSR in the initial speeds and 15.5% better than ext-QOLSR. In Average end-to-end delay, En-DSR outperformed the two routing protocols with around 70%. Despite the deteriorated performance as the speed increased, En-DSR performed relatively close to the DSR. Stability value can be used in estimating reliable route and advantageous under low mobile nodes density in the ad hoc network. Transmitting over a more reliable route in the highly dynamic mobile ad hoc network where mobile nodes move vigorously, En-DSR performs well with the ranges of speed from 1m/s to 4m/s. With high speeds, all routing protocol performances deteriorate.

Under dynamic topology, link breakages encountered more often where link re-establishment took place increases the routing packets. En-DSR achieved a better average end-to-end delay in the simulations with QoS provided to the requesting applications. Moreover, real-time applications can make use of IPv6 Flow Label field in indicating the required QoS where priority can be made higher than to those non-real time applications. In conclusion, QoS is required in order to deliver a better service and improve the network performance.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Performance achieved by En-DSR is comparable to conventional DSR in the highly dynamic ad hoc network where En-DSR adaptively utilizes stability value and shortest path route selection algorithm. Since, En-DSR is enhanced from conventional DSR with QoS improvised for ad hoc network transmission it shows a relatively close performance under highly mobile ad hoc network. At lower speed ranges, En-DSR showed a better performance in the low mobile node density as compared to the DSR and ext-QOLSR but all routing protocols deteriorated as speed increased. PDF in En-DSR outperformed 1%-2% against conventional DSR and 15% against ext-QOLSR. Packet drop in En-DSR is reduced in lower speed ranges and outperformed ext-QOLSR. En-DSR is an adaptive route selection scheme route data over a more reliable path in the mobile ad hoc network with the additional stability value as the metric to evaluate for better route. Under resource reservation, data packets are processed with better efficiency and resources can be managed more appropriately.

Furthermore, with increased speed of mobile nodes control packets increased as an overhead in the routing mechanism. Such overhead reduces performances especially link breakages are often encountered. Imminent breakages required more updates to be done and this presented that under highly dynamic mobile ad hoc network, table-driven is not advantageous. In En-DSR, normalized routing load is lower than conventional DSR and in ext-QOLSR has higher volume of data packet over the data transmissions. Control packets filled the network with latest updates and with increasing changes, updates are overwhelming. On-demand ad hoc network such as DSR in the other hand required less routing packets to be

delivered, hence it showed a lower overhead. When transmission becomes less, table-driven routing protocols are still occupying the network resources for the topology information. Distributing the latest information among the nodes, scarce resources are nonetheless required. It is recommended to adopt on-demand routing protocol in such network environment as shown in other simulation reports (Ari, Jethani, Rangnekar, & Natarajan, 2000) and (Johansson, Larsson, Hedman, Mielczarek, & Degermark, 1999).

As conclusion, En-DSR is adaptive in selecting routes for a more reliable transmission over the highly dynamic mobile ad hoc network. Packet Delivery Fraction, Normalized Routing Load, Packet Loss and Throughput perform relatively well.

5.2 Future Work

IP address migration from version 4 to version 6 is inevitable as demand of IP addresses increase day by day. Despite the packet header being newly restructured, the length does not seem to be favorable as compared to the former length. Hence, more researches have to be done to deploy them in wireless ad hoc network routing protocol with more initiatives in QoS provisions. Applications and higher layer protocols can be equipped with the ability of understanding the direct parameters labeled in the Flow Label field. Despite routing issues in the mobile ad hoc networks are being studied, security risks and mutual agreement between nodes need to be given their fair share of consideration. In wireless ad hoc network, transmission takes place upon willingness to forward data on behalf but if the mutual understanding does not exist, ad hoc network will not be able to provide the service. Lifetime of the mobile nodes that is measured in term of battery could be added into the picture and therefore more researches should be done. Furthermore, security risks should be monitored too. Laptop and smartphone can be compromised with unknown agents such as malware that affect communications between two ends.

Future research could be done with real experiments that involve Linux kernel. In real wireless ad hoc network experiments, laptops and smartphone devices can be used. Further simulations in the mobile ad hoc networks could be done using other simulators such as OPNET (2006) and Qualnet (Scalable Network Technologies, 2006).

REFERENCES

Agha, K., & Badis, H. (2004). An Efficient QOLSR Extension Protocol for QoS in Ad Hoc Networks. *IEEE Vehicular Technology Conference* , IEEE Communications Society.

Ahn, G.-S., Cambell, A., Lee, S.-B., & Zhang, X. (1999). INSIGNIA. *Draft-IEFT-MANET-INSIGNIA-01* .

Al-Khwildi, A., Loo, K., & Al-Raweshidy, H. (2007). An Efficient Routing Protocol for Wireless Ad Hoc Networks. *4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunication (SETIT 2007).* Tunisia.

Ari, I., Jethani, N., Rangnekar, A., & Natarajan, S. (2000). *Performance Analysis and Comparison of Ah-Hoc Routing Protocols.* University of Maryland and Baltimore County . CMSC 691T, Mobile Computing.

Badis, H., Munaretto, A., Agha, K., & Pujolle, G. (2003). QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay. *Proceeding of IEEE MWCN 2003, Singapore* .

Banerjee, R., Malhotra, S., & Mahaveer, M. (2002). *A Modified Specification for use of the IPv6 Flow Label for providing an efficient Quality of Service using hybrid aprroach.* Internet draft-banerjee-flowlabel-ipv6-qos-03.txt, IETF.

Banerjee, R., Preethi, N., & Sethuraman, M. (2002). *Design and Implementation for the QoS in IPv6 using the modified Hop-by-Hop Extension heade.* Internet draft-banerjee-ipv6-quality-service-02.txt, IETF.

Bhavanam, S., Potluri, A., Yeruva, J., Masupha, T., & Negi, A. (2008). QoS-Aware Dynamic Source Routing using DiffServ Principles. *Distributed Computing and*

Internet Technology: 5th International Conference, ICDCIT 2008 (pp. 133-138). New Delhi, India: Springer.

Blake, S., Black, D., Carlson, M., Daview, E., Wang, Z., & Weiss, W. (1998). *RFC 2475: An Architecture for Differentiated Services.* Network Working Group.

Braden, B., Zhang, L., Berson, S., Herzog, S., & Jamin, S. (1997). *RFC 2205 : Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.* Network Working Group.

Braden, R., Clark, D., & Shenker, S. (1994). *RFC 1633 : Intergrated Services in the Internet Architecture: an Overview.* Network Working Group.

Buckwalter, J. (1999). *Frame Relay: Technology and Practice.* Addison Wesley Longman Inc.

California, I. S. (1981). *RFC 791 : DARPA Internat Program Protocol Specification.* Network Working Group.

Chaudhuri, P. (2009). *Computer Organization and Design.* New Delhi: PHI Learning Private Limited.

Clausen, T., & Jacquet, P. (2003). *Optimized Link State Routing Protocol (OLSR), RFC 3626.* Internet Engineering Task Force.

Conta, A., & Carpenter, B. (2001). *A Proposal for IPv6 Flow Label Specification* . Internet Draft draft-banerjee-flowlabel-ipv6-qos-03.txt, IETF.

Cordeiro, C. M., & Agrawal, D. P. (2006). In *Ad Hoc & Sensor Networks : Theory and Applications* (pp. 33-37). World Scientific Publishing.

Crawley, E., Nair, R., Rajagopalan, B., & Sandick, H. (1998). *RFC 2386 :- A Framework for QoS-Based Routing in the Internet.* Network Working Group.

Deering, S., & Hinden, R. (1998). *RFC 2460: Internet Protocol Version 6 (IPv6) Specification.* Network Working Group.

Dunn, J., & Martin, C. (2000). *RFC 2761: Terminology for ATM Benchmarking.* Network Working Group.

Fall, K., & Varadhan, K. (2009). *NS Manual* . Retrieved February 2, 2010, from The Network Simulator ns-2: Documentation: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

Ferguson, P., & Huston, G. (1998). *Quality of Service: Delivering QoS on the Internet and in Corporate Networks.* John Wiley & Sons.

Greis, M. (1995). *Tutorial for the UCB/LBNL/VINT Network Simulator "NS".* Retrieved July 4, 2011, from Network Simulator Tutorial: http://www.isi.edu/nsnam/ns/tutorial/nsintro.html

H.Badis, & K.A.Agha. (2008). Optimal Path Selection in Ad Hoc Networks Based on Multiple Metrics: Bandwidth and Delay. *Advances in Wireless Ad Hoc and Sensor Networks, Signals and Communication Technology* (pp. 35-60). Springer Science + Business Media , LLC .

Hagen, S. (2006). *IPv6 Essentials - Integrating IPv6 Into Your IPv4 Networks 2nd Edition.* O'Reilly.

Hu, Y.-C., Johnson, D. B., & Maltz, D. A. (2001). *Flow State in the Dynamic Source Routing Protocol for Mobile Ad Hoc Networks.* Draft-IEFT-MANET-dsrflow-00.

Hu, Y.-C., Johnson, D. B., & Maltz, D. A. (2001). Flow State in the Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. *IETF Manet Working Group Internet Draft <draft-ieft-manet-dsrflow-00.txt>* .

Inc OPNET Technologies. (2006). *OPNET Wireless network modeling and simulation*. Retrieved from Software Package: http://www.opnet.com

Issariyakul, T., & Hossain, E. (2009). *Introduction to Network Simulator NS2*. New York, USA: Springer Science.

ITU-T Recommendations. (2008). E.800: Definitions of terms related to Quality of Services. *International Telecommunication Union* .

Jagadeesan, H., & Singh, T. (2002). *A Radical Approach in providing Quality-of-Service over the Internet using the 20-bit IPv6 Flow Label field.* Internet draft-jagadeesan-rad-approach-service-01.txt, IETF.

Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., & Degermark, M. (1999). Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks. *Proceedings of the 5th annual ACM/IEEE Internation Conference on Mobile Computing and Networking*, (pp. 195-206). Seattle, Washington .

Johnson, D. B., & Hu, Y.-C. (2001). Implicit Source Routing for On Demand Ad Hoc Networks. *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networkin & Computing (MobiHoc 2001)*, (pp. 1-10). Long Beach, CA.

Johnson, D. B., & Hu, Y.-C. (2001). Implicit Source Routing for On Demand Ad Hoc Networks. *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001)* , pp. 1-10, ACM, Long Beach, CA.

Johnson, D. B., Maltz, D., & Broch, J. (2001). *DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad hoc Networking.* Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Johnson, D., & Maltz, D. A. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks. In H. T.Imielinski, *Mobile Computing* (pp. 153-181). Kluwer Academic Publishers.

Kasera, S. (2006). *ATM Networks: Concepts and Protocols Second Edition.* Tata McGraw-Hill Education .

Lee, S.-B., & Campbell, A. (1998). INSIGNIA: In-Band Signalling Support for QoS In Mobile Ad Hoc Networks. *Proceedings of 5th Intl. Workshop on Mobile Multimedia Communication MoMuc '98* .

Liu, C., & Kaiser, J. (2005). *A Survey of Mobile Ad Hoc Network Routing Protocols.* Ulm, Germany: University of ULM Tech Report Series Nr. 2003-08.

Maltz, D. A., Johnson, D. B., & Hu, Y.-C. (2004). The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). *Draft-IEFT-MANET-DSR-10* .

McTasney, R., Grunwald, D., & Sicker, D. (2009). Low Latency in Wireless Mesh Network. *Guide To Wireless Mesh Networks* , 379-424.

Murthy, S., & Garcia-Luna-Aceves, J. (1996). An Efficient Routing Protocol for Wireless Networks. *Mobile Networks and Applications - Special issue: Routing in mobile communications networks* , 183-197.

Nichols, K., Blake, S., Baker, F., & Black, D. (1998). *RFC 2474: Definition of the Differentitated Services Field (DS Field) in the IPv4 and IPv6 Headers.* Network Working Group.

Nichols, K., Jacobson, V., & Zhang, L. (1999). *RFC 2638: A Two-bit Differentiated Service Architecture for the Internet.* Network Working Group.

Park, V. D., & Corson, M. S. (1997). A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. *Proc. INFOCOM '97* .

Park, V., & Corson, M. (1997). A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. *INFOCOM '97 Sixteeth Annual Joint Conference of IEEE Computer and Communications Societies* (pp. 1405-1413). Kobe, Japan: IEEE Communications Society.

Partridge, C. (1992). *RFC 1363: A Proposed Flow Specifications.* Network Working Group.

Perkins, C. E., & Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *SIGCOMM ' 94 Proceedings of the conference on Communication Architectures, Protocols and Applications*, (pp. 234-244). New York.

Perkins, C., & Royer, E. (1999). Ad-Hoc on-demand distance vector routing. *WMCSA '99 Second IEEE Workshop on Mobile Computing Systems and Applications* (pp. 90-100). New Orleans, LA: IEEE Communication Society.

Perkins, C., Belding-Royer, E., & Das, S. (2003). *RFC - Ad Hoc On-Demand Distance Vector (AODV) Routing* . IETF.

Prakash, B. (2004). *Using The 20 Bit Flow Label Field In The IPv6 header To Indicate Desirable Quality of Service on The Internet.* M.S.Thesis, University of Colorado, Boulder, United States.

R.Branden. (1989). *RFC 1122: Requirements for Internet Host- Communication Layers* . Network Working Group.

R.Dube, C.D.Rais, K.Wang, & S.K.Tripathi. (1997). Signal Stability Based Adaptive Routing for ad hoc mobile networks. *IEEE Pers. Comm* , 36-45.

Rajahalme, J., Conta, A., Carpenter, B., & Deering, S. (2004). *RFC 3697 - IPv6 Flow Label Specification* . Network Working Group.

Rosen, E., Viswanathan, A., & Callon, R. (2001). *RFC 3031: Multiprotocol Label Switching Architecture.* Network Working Group.

Royer, E. M., & Toh, C.-K. (1999). A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications* (pp. 46-55). IEEE Communications Society.

Sarangapani, J. (2007). *Wireless Ad Hoc and Sensor Networks: Protocols, Performance and Control.* CRC Press.

Scalable Network Technologies. (2006). *QualNet Network Simulator*. Retrieved from http://www.scalable-networks.com

Studi, P. (2003). *Quality of Service for Mobile Ad Hoc Networks.* Zurich: Swiss Federal Institute of Technology.

Toh, C.-K. (2002). *Ah Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall PTR.

Toh, C.-K. (1997). Associativity Based Routing For Ad Hoc Mobile Networks. *Wireless Personal Communication Journal, Special Issue on Mobile Networking and Computing Systems* , 103-139.

Vivek Alwayn. (2002). In *Advanced MPLS Design and Iimplementation* (pp. 4-5). Indianapolis, USA: Cisco Press.

Wroclawski, J. (1997). *RFC 2210: The Use of RSVP with IETF Integrated Services.* Network Working Group.

Xue, J., Stuedi, P., & Alonso, G. (2003). *ASAP: An Adaptive QoS Protocol for Mobile Ad Hoc Networks.* Zurich: Swiss Federal Institute of Technology.

Xue, Q., & Ganz, A. (2003). Ad hoc QoS on-demand routing (AQOR) in Mobile ad hoc networks. *Journal Parallel and Distributed Computing* , 154-165.

Zygmunt, J. H. (1997). A new routing protocol for the reconfigurable wireless networks. *Proc. of IEEE 6th International Conference on Universal Personal Communications (ICUPC '97)* , 562–566.

APPENDIX A

```
# 1 connecting to 2 at time
    2.5568388786897245
#
setudp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
setcbr_(0) [new
    Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0)
    start"

#
# 4 connecting to 5 at time
    56.333118917575632
#
setudp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
setcbr_(1) [new
    Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1)
    start"

#
# 4 connecting to 6 at time
    146.96568928983328
#
setudp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
setcbr_(2) [new
    Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 146.96568928983328 "$cbr_(2)
    start"

#
# 6 connecting to 7 at time
    55.634230382570173
#
setudp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
setcbr_(3) [new
    Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.25
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3)
    start"
```

```
#
# 7 connecting to 8 at time
    29.546173154165118
#
setudp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(4)
setcbr_(4) [new
    Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set interval_ 0.25
$cbr_(4) set random_ 1
$cbr_(4) set maxpkts_ 10000
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 29.546173154165118 "$cbr_(4)
    start"

#
# 7 connecting to 9 at time
    7.7030203154790309
#
setudp_(5) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(5)
set null_(5) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(5)
setcbr_(5) [new
    Application/Traffic/CBR]
$cbr_(5) set packetSize_ 512
$cbr_(5) set interval_ 0.25
$cbr_(5) set random_ 1
$cbr_(5) set maxpkts_ 10000
$cbr_(5) attach-agent $udp_(5)
$ns_ connect $udp_(5) $null_(5)
$ns_ at 7.7030203154790309 "$cbr_(5)
    start"

#
# 8 connecting to 9 at time
    20.48548468411224
#
setudp_(6) [new Agent/UDP]
$ns_ attach-agent $node_(8) $udp_(6)
set null_(6) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(6)
setcbr_(6) [new
    Application/Traffic/CBR]
$cbr_(6) set packetSize_ 512
$cbr_(6) set interval_ 0.25
$cbr_(6) set random_ 1
$cbr_(6) set maxpkts_ 10000
$cbr_(6) attach-agent $udp_(6)
$ns_ connect $udp_(6) $null_(6)
$ns_ at 20.48548468411224 "$cbr_(6)
    start"

#
# 9 connecting to 10 at time
    76.258212521792487
#
setudp_(7) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(7)
set null_(7) [new Agent/Null]
$ns_ attach-agent $node_(10) $null_(7)
setcbr_(7) [new
    Application/Traffic/CBR]
$cbr_(7) set packetSize_ 512
$cbr_(7) set interval_ 0.25
$cbr_(7) set random_ 1
$cbr_(7) set maxpkts_ 10000
$cbr_(7) attach-agent $udp_(7)
$ns_ connect $udp_(7) $null_(7)
$ns_ at 76.258212521792487 "$cbr_(7)
    start"
```

```
#
# 9 connecting to 11 at time
    31.464945688594575
#
setudp_(8) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(8)
set null_(8) [new Agent/Null]
$ns_ attach-agent $node_(11) $null_(8)
setcbr_(8) [new
    Application/Traffic/CBR]
$cbr_(8) set packetSize_ 512
$cbr_(8) set interval_ 0.25
$cbr_(8) set random_ 1
$cbr_(8) set maxpkts_ 10000
$cbr_(8) attach-agent $udp_(8)
$ns_ connect $udp_(8) $null_(8)
$ns_ at 31.464945688594575 "$cbr_(8)
    start"

#
# 11 connecting to 12 at time
    62.77338456491632
#
setudp_(9) [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp_(9)
set null_(9) [new Agent/Null]
$ns_ attach-agent $node_(12) $null_(9)
setcbr_(9) [new
    Application/Traffic/CBR]
$cbr_(9) set packetSize_ 512
$cbr_(9) set interval_ 0.25
$cbr_(9) set random_ 1
$cbr_(9) set maxpkts_ 10000
$cbr_(9) attach-agent $udp_(9)
$ns_ connect $udp_(9) $null_(9)
$ns_ at 62.77338456491632 "$cbr_(9)
    start"

#
# 11 connecting to 13 at time
    46.455830739092008
#
setudp_(10) [new Agent/UDP]
$ns_ attach-agent $node_(11)
    $udp_(10)
set null_(10) [new Agent/Null]
$ns_ attach-agent $node_(13)
    $null_(10)
setcbr_(10) [new
    Application/Traffic/CBR]
$cbr_(10) set packetSize_ 512
$cbr_(10) set interval_ 0.25
$cbr_(10) set random_ 1
$cbr_(10) set maxpkts_ 10000
$cbr_(10) attach-agent $udp_(10)
$ns_ connect $udp_(10) $null_(10)
$ns_ at 46.455830739092008
    "$cbr_(10) start"

#
# 13 connecting to 14 at time
    83.900868549896813
#
setudp_(11) [new Agent/UDP]
$ns_ attach-agent $node_(13)
    $udp_(11)
set null_(11) [new Agent/Null]
$ns_ attach-agent $node_(14)
    $null_(11)
setcbr_(11) [new
    Application/Traffic/CBR]
$cbr_(11) set packetSize_ 512
$cbr_(11) set interval_ 0.25
$cbr_(11) set random_ 1
$cbr_(11) set maxpkts_ 10000
$cbr_(11) attach-agent $udp_(11)
$ns_ connect $udp_(11) $null_(11)
$ns_ at 83.900868549896813
    "$cbr_(11) start"
```

```
#
# 14 connecting to 15 at time
    155.17211061677529
#
setudp_(12) [new Agent/UDP]
$ns_ attach-agent $node_(14)
    $udp_(12)
set null_(12) [new Agent/Null]
$ns_ attach-agent $node_(15)
    $null_(12)
setcbr_(12) [new
    Application/Traffic/CBR]
$cbr_(12) set packetSize_ 512
$cbr_(12) set interval_ 0.25
$cbr_(12) set random_ 1
$cbr_(12) set maxpkts_ 10000
$cbr_(12) attach-agent $udp_(12)
$ns_ connect $udp_(12) $null_(12)
$ns_ at 155.17211061677529
    "$cbr_(12) start"

#
# 15 connecting to 16 at time
    39.088702704333095
#
setudp_(13) [new Agent/UDP]
$ns_ attach-agent $node_(15)
    $udp_(13)
set null_(13) [new Agent/Null]
$ns_ attach-agent $node_(16)
    $null_(13)
setcbr_(13) [new
    Application/Traffic/CBR]
$cbr_(13) set packetSize_ 512
$cbr_(13) set interval_ 0.25
$cbr_(13) set random_ 1
$cbr_(13) set maxpkts_ 10000
$cbr_(13) attach-agent $udp_(13)
$ns_ connect $udp_(13) $null_(13)
$ns_ at 39.088702704333095
    "$cbr_(13) start"

#
# 15 connecting to 17 at time
    43.420613009212822
#
setudp_(14) [new Agent/UDP]
$ns_ attach-agent $node_(15)
    $udp_(14)
set null_(14) [new Agent/Null]
$ns_ attach-agent $node_(17)
    $null_(14)
setcbr_(14) [new
    Application/Traffic/CBR]
$cbr_(14) set packetSize_ 512
$cbr_(14) set interval_ 0.25
$cbr_(14) set random_ 1
$cbr_(14) set maxpkts_ 10000
$cbr_(14) attach-agent $udp_(14)
$ns_ connect $udp_(14) $null_(14)
$ns_ at 43.420613009212822
    "$cbr_(14) start"

#
# 16 connecting to 17 at time
    121.92280978985261
#
setudp_(15) [new Agent/UDP]
$ns_ attach-agent $node_(16)
    $udp_(15)
set null_(15) [new Agent/Null]
$ns_ attach-agent $node_(17)
    $null_(15)
setcbr_(15) [new
    Application/Traffic/CBR]
$cbr_(15) set packetSize_ 512
$cbr_(15) set interval_ 0.25
$cbr_(15) set random_ 1
$cbr_(15) set maxpkts_ 10000
$cbr_(15) attach-agent $udp_(15)
$ns_ connect $udp_(15) $null_(15)
$ns_ at 121.92280978985261
    "$cbr_(15) start"
```

```
#
# 16 connecting to 18 at time
    137.20174070317378
#
setudp_(16) [new Agent/UDP]
$ns_ attach-agent $node_(16)
    $udp_(16)
set null_(16) [new Agent/Null]
$ns_ attach-agent $node_(18)
    $null_(16)
setcbr_(16) [new
    Application/Traffic/CBR]
$cbr_(16) set packetSize_ 512
$cbr_(16) set interval_ 0.25
$cbr_(16) set random_ 1
$cbr_(16) set maxpkts_ 10000
$cbr_(16) attach-agent $udp_(16)
$ns_ connect $udp_(16) $null_(16)
$ns_ at 137.20174070317378
    "$cbr_(16) start"

#
# 17 connecting to 18 at time
    72.99343390995331
#
setudp_(17) [new Agent/UDP]
$ns_ attach-agent $node_(17)
    $udp_(17)
set null_(17) [new Agent/Null]
$ns_ attach-agent $node_(18)
    $null_(17)
setcbr_(17) [new
    Application/Traffic/CBR]
$cbr_(17) set packetSize_ 512
$cbr_(17) set interval_ 0.25
$cbr_(17) set random_ 1
$cbr_(17) set maxpkts_ 10000
$cbr_(17) attach-agent $udp_(17)
$ns_ connect $udp_(17) $null_(17)
$ns_ at 72.99343390995331 "$cbr_(17)
    start"

#
# 17 connecting to 19 at time
    19.655724884781858
#
setudp_(18) [new Agent/UDP]
$ns_ attach-agent $node_(17)
    $udp_(18)
set null_(18) [new Agent/Null]
$ns_ attach-agent $node_(19)
    $null_(18)
setcbr_(18) [new
    Application/Traffic/CBR]
$cbr_(18) set packetSize_ 512
$cbr_(18) set interval_ 0.25
$cbr_(18) set random_ 1
$cbr_(18) set maxpkts_ 10000
$cbr_(18) attach-agent $udp_(18)
$ns_ connect $udp_(18) $null_(18)
$ns_ at 19.655724884781858
    "$cbr_(18) start"

#
# 20 connecting to 21 at time
    170.32769159894795
#
setudp_(19) [new Agent/UDP]
$ns_ attach-agent $node_(20)
    $udp_(19)
set null_(19) [new Agent/Null]
$ns_ attach-agent $node_(21)
    $null_(19)
setcbr_(19) [new
    Application/Traffic/CBR]
$cbr_(19) set packetSize_ 512
$cbr_(19) set interval_ 0.25
$cbr_(19) set random_ 1
$cbr_(19) set maxpkts_ 10000
$cbr_(19) attach-agent $udp_(19)
$ns_ connect $udp_(19) $null_(19)
$ns_ at 170.32769159894795
    "$cbr_(19) start"
```

```
#
# 20 connecting to 22 at time
    160.44260791523504
#
setudp_(20) [new Agent/UDP]
$ns_ attach-agent $node_(20)
    $udp_(20)
set null_(20) [new Agent/Null]
$ns_ attach-agent $node_(22)
    $null_(20)
setcbr_(20) [new
    Application/Traffic/CBR]
$cbr_(20) set packetSize_ 512
$cbr_(20) set interval_ 0.25
$cbr_(20) set random_ 1
$cbr_(20) set maxpkts_ 10000
$cbr_(20) attach-agent $udp_(20)
$ns_ connect $udp_(20) $null_(20)
$ns_ at 160.44260791523504
    "$cbr_(20) start"

#
# 24 connecting to 25 at time
    60.419296464146719
#
setudp_(21) [new Agent/UDP]
$ns_ attach-agent $node_(24)
    $udp_(21)
set null_(21) [new Agent/Null]
$ns_ attach-agent $node_(25)
    $null_(21)
setcbr_(21) [new
    Application/Traffic/CBR]
$cbr_(21) set packetSize_ 512
$cbr_(21) set interval_ 0.25
$cbr_(21) set random_ 1
$cbr_(21) set maxpkts_ 10000
$cbr_(21) attach-agent $udp_(21)
$ns_ connect $udp_(21) $null_(21)
$ns_ at 60.419296464146719
    "$cbr_(21) start"

#
# 26 connecting to 27 at time
    46.258873029732555
#
setudp_(22) [new Agent/UDP]
$ns_ attach-agent $node_(26)
    $udp_(22)
set null_(22) [new Agent/Null]
$ns_ attach-agent $node_(27)
    $null_(22)
setcbr_(22) [new
    Application/Traffic/CBR]
$cbr_(22) set packetSize_ 512
$cbr_(22) set interval_ 0.25
$cbr_(22) set random_ 1
$cbr_(22) set maxpkts_ 10000
$cbr_(22) attach-agent $udp_(22)
$ns_ connect $udp_(22) $null_(22)
$ns_ at 46.258873029732555
    "$cbr_(22) start"

#
# 27 connecting to 28 at time
    98.067954088592884
#
setudp_(23) [new Agent/UDP]
$ns_ attach-agent $node_(27)
    $udp_(23)
set null_(23) [new Agent/Null]
$ns_ attach-agent $node_(28)
    $null_(23)
setcbr_(23) [new
    Application/Traffic/CBR]
$cbr_(23) set packetSize_ 512
$cbr_(23) set interval_ 0.25
$cbr_(23) set random_ 1
$cbr_(23) set maxpkts_ 10000
$cbr_(23) attach-agent $udp_(23)
$ns_ connect $udp_(23) $null_(23)
$ns_ at 98.067954088592884
    "$cbr_(23) start"
```

```
#
# 28 connecting to 29 at time
    47.128346453946243
#
setudp_(24) [new Agent/UDP]
$ns_ attach-agent $node_(28)
    $udp_(24)
set null_(24) [new Agent/Null]
$ns_ attach-agent $node_(29)
    $null_(24)
setcbr_(24) [new
    Application/Traffic/CBR]
$cbr_(24) set packetSize_ 512
$cbr_(24) set interval_ 0.25
$cbr_(24) set random_ 1
$cbr_(24) set maxpkts_ 10000
$cbr_(24) attach-agent $udp_(24)
$ns_ connect $udp_(24) $null_(24)
$ns_ at 47.128346453946243
    "$cbr_(24) start"

#
# 28 connecting to 30 at time
    99.87114126788039
#
setudp_(25) [new Agent/UDP]
$ns_ attach-agent $node_(28)
    $udp_(25)
set null_(25) [new Agent/Null]
$ns_ attach-agent $node_(30)
    $null_(25)
setcbr_(25) [new
    Application/Traffic/CBR]
$cbr_(25) set packetSize_ 512
$cbr_(25) set interval_ 0.25
$cbr_(25) set random_ 1
$cbr_(25) set maxpkts_ 10000
$cbr_(25) attach-agent $udp_(25)
$ns_ connect $udp_(25) $null_(25)
$ns_ at 99.87114126788039 "$cbr_(25)
    start"

#
# 33 connecting to 34 at time
    170.4269140541679
#
setudp_(26) [new Agent/UDP]
$ns_ attach-agent $node_(33)
    $udp_(26)
set null_(26) [new Agent/Null]
$ns_ attach-agent $node_(34)
    $null_(26)
setcbr_(26) [new
    Application/Traffic/CBR]
$cbr_(26) set packetSize_ 512
$cbr_(26) set interval_ 0.25
$cbr_(26) set random_ 1
$cbr_(26) set maxpkts_ 10000
$cbr_(26) attach-agent $udp_(26)
$ns_ connect $udp_(26) $null_(26)
$ns_ at 170.4269140541679 "$cbr_(26)
    start"

#
# 33 connecting to 35 at time
    139.08523271749041
#
setudp_(27) [new Agent/UDP]
$ns_ attach-agent $node_(33)
    $udp_(27)
set null_(27) [new Agent/Null]
$ns_ attach-agent $node_(35)
    $null_(27)
setcbr_(27) [new
    Application/Traffic/CBR]
$cbr_(27) set packetSize_ 512
$cbr_(27) set interval_ 0.25
$cbr_(27) set random_ 1
$cbr_(27) set maxpkts_ 10000
$cbr_(27) attach-agent $udp_(27)
$ns_ connect $udp_(27) $null_(27)
$ns_ at 139.08523271749041
    "$cbr_(27) start"
```

```
#
# 35 connecting to 36 at time
    32.17101298839367
#
setudp_(28) [new Agent/UDP]
$ns_ attach-agent $node_(35)
    $udp_(28)
set null_(28) [new Agent/Null]
$ns_ attach-agent $node_(36)
    $null_(28)
setcbr_(28) [new
    Application/Traffic/CBR]
$cbr_(28) set packetSize_ 512
$cbr_(28) set interval_ 0.25
$cbr_(28) set random_ 1
$cbr_(28) set maxpkts_ 10000
$cbr_(28) attach-agent $udp_(28)
$ns_ connect $udp_(28) $null_(28)
$ns_ at 32.17101298839367 "$cbr_(28)
    start"

#
# 36 connecting to 37 at time
    16.444244867397586
#
setudp_(29) [new Agent/UDP]
$ns_ attach-agent $node_(36)
    $udp_(29)
set null_(29) [new Agent/Null]
$ns_ attach-agent $node_(37)
    $null_(29)
setcbr_(29) [new
    Application/Traffic/CBR]
$cbr_(29) set packetSize_ 512
$cbr_(29) set interval_ 0.25
$cbr_(29) set random_ 1
$cbr_(29) set maxpkts_ 10000
$cbr_(29) attach-agent $udp_(29)
$ns_ connect $udp_(29) $null_(29)
$ns_ at 16.444244867397586
    "$cbr_(29) start"

#
# 36 connecting to 38 at time
    88.50990661816202
#
setudp_(30) [new Agent/UDP]
$ns_ attach-agent $node_(36)
    $udp_(30)
set null_(30) [new Agent/Null]
$ns_ attach-agent $node_(38)
    $null_(30)
setcbr_(30) [new
    Application/Traffic/CBR]
$cbr_(30) set packetSize_ 512
$cbr_(30) set interval_ 0.25
$cbr_(30) set random_ 1
$cbr_(30) set maxpkts_ 10000
$cbr_(30) attach-agent $udp_(30)
$ns_ connect $udp_(30) $null_(30)
$ns_ at 88.50990661816202 "$cbr_(30)
    start"

#
# 38 connecting to 39 at time
    95.921739365869087
#
setudp_(31) [new Agent/UDP]
$ns_ attach-agent $node_(38)
    $udp_(31)
set null_(31) [new Agent/Null]
$ns_ attach-agent $node_(39)
    $null_(31)
setcbr_(31) [new
    Application/Traffic/CBR]
$cbr_(31) set packetSize_ 512
$cbr_(31) set interval_ 0.25
$cbr_(31) set random_ 1
$cbr_(31) set maxpkts_ 10000
$cbr_(31) attach-agent $udp_(31)
$ns_ connect $udp_(31) $null_(31)
$ns_ at 95.921739365869087
    "$cbr_(31) start"
```

```
#
# 38 connecting to 40 at time
    132.31640266828072
#
setudp_(32) [new Agent/UDP]
$ns_ attach-agent $node_(38)
    $udp_(32)
set null_(32) [new Agent/Null]
$ns_ attach-agent $node_(40)
    $null_(32)
setcbr_(32) [new
    Application/Traffic/CBR]
$cbr_(32) set packetSize_ 512
$cbr_(32) set interval_ 0.25
$cbr_(32) set random_ 1
$cbr_(32) set maxpkts_ 10000
$cbr_(32) attach-agent $udp_(32)
$ns_ connect $udp_(32) $null_(32)
$ns_ at 132.31640266828072
    "$cbr_(32) start"

#
# 41 connecting to 42 at time
    162.51257573371407
#
setudp_(33) [new Agent/UDP]
$ns_ attach-agent $node_(41)
    $udp_(33)
set null_(33) [new Agent/Null]
$ns_ attach-agent $node_(42)
    $null_(33)
setcbr_(33) [new
    Application/Traffic/CBR]
$cbr_(33) set packetSize_ 512
$cbr_(33) set interval_ 0.25
$cbr_(33) set random_ 1
$cbr_(33) set maxpkts_ 10000
$cbr_(33) attach-agent $udp_(33)
$ns_ connect $udp_(33) $null_(33)
$ns_ at 162.51257573371407
    "$cbr_(33) start"

# 41 connecting to 43 at time
    131.04751890108341
#
setudp_(34) [new Agent/UDP]
$ns_ attach-agent $node_(41)
    $udp_(34)
set null_(34) [new Agent/Null]
$ns_ attach-agent $node_(43)
    $null_(34)
setcbr_(34) [new
    Application/Traffic/CBR]
$cbr_(34) set packetSize_ 512
$cbr_(34) set interval_ 0.25
$cbr_(34) set random_ 1
$cbr_(34) set maxpkts_ 10000
$cbr_(34) attach-agent $udp_(34)
$ns_ connect $udp_(34) $null_(34)
$ns_ at 131.04751890108341
    "$cbr_(34) start"

#
# 42 connecting to 43 at time
    105.50024639139895
#
setudp_(35) [new Agent/UDP]
$ns_ attach-agent $node_(42)
    $udp_(35)
set null_(35) [new Agent/Null]
$ns_ attach-agent $node_(43)
    $null_(35)
setcbr_(35) [new
    Application/Traffic/CBR]
$cbr_(35) set packetSize_ 512
$cbr_(35) set interval_ 0.25
$cbr_(35) set random_ 1
$cbr_(35) set maxpkts_ 10000
$cbr_(35) attach-agent $udp_(35)
$ns_ connect $udp_(35) $null_(35)
$ns_ at 105.50024639139895
    "$cbr_(35) start"
```

```
#
# 42 connecting to 44 at time
    132.45129531829213
#
setudp_(36) [new Agent/UDP]
$ns_ attach-agent $node_(42)
    $udp_(36)
set null_(36) [new Agent/Null]
$ns_ attach-agent $node_(44)
    $null_(36)
setcbr_(36) [new
    Application/Traffic/CBR]
$cbr_(36) set packetSize_ 512
$cbr_(36) set interval_ 0.25
$cbr_(36) set random_ 1
$cbr_(36) set maxpkts_ 10000
$cbr_(36) attach-agent $udp_(36)
$ns_ connect $udp_(36) $null_(36)
$ns_ at 132.45129531829213
    "$cbr_(36) start"

#
# 43 connecting to 44 at time
    64.09054449018582
#
setudp_(37) [new Agent/UDP]
$ns_ attach-agent $node_(43)
    $udp_(37)
set null_(37) [new Agent/Null]
$ns_ attach-agent $node_(44)
    $null_(37)
setcbr_(37) [new
    Application/Traffic/CBR]
$cbr_(37) set packetSize_ 512
$cbr_(37) set interval_ 0.25
$cbr_(37) set random_ 1
$cbr_(37) set maxpkts_ 10000
$cbr_(37) attach-agent $udp_(37)
$ns_ connect $udp_(37) $null_(37)
$ns_ at 64.09054449018582 "$cbr_(37)
    start"

#
# 43 connecting to 45 at time
    100.5135456661291
#
setudp_(38) [new Agent/UDP]
$ns_ attach-agent $node_(43)
    $udp_(38)
set null_(38) [new Agent/Null]
$ns_ attach-agent $node_(45)
    $null_(38)
setcbr_(38) [new
    Application/Traffic/CBR]
$cbr_(38) set packetSize_ 512
$cbr_(38) set interval_ 0.25
$cbr_(38) set random_ 1
$cbr_(38) set maxpkts_ 10000
$cbr_(38) attach-agent $udp_(38)
$ns_ connect $udp_(38) $null_(38)
$ns_ at 100.5135456661291 "$cbr_(38)
    start"

#
# 44 connecting to 45 at time
    141.0795085137149
#
setudp_(39) [new Agent/UDP]
$ns_ attach-agent $node_(44)
    $udp_(39)
set null_(39) [new Agent/Null]
$ns_ attach-agent $node_(45)
    $null_(39)
setcbr_(39) [new
    Application/Traffic/CBR]
$cbr_(39) set packetSize_ 512
$cbr_(39) set interval_ 0.25
$cbr_(39) set random_ 1
$cbr_(39) set maxpkts_ 10000
$cbr_(39) attach-agent $udp_(39)
$ns_ connect $udp_(39) $null_(39)
$ns_ at 141.0795085137149 "$cbr_(39)
    start"
#Total sources/connections: 25/40
```