Design and Investigation of Scalable Multicast Recursive Protocols for Wired and Wireless Ad hoc Networks

By
Firas Al-Balas

A Thesis Submitted in Partial Fulfillment for the degree of doctor of philosophy

in the
INTEGRATED COMMUNICATIONS RESEARCH CENTRE
FACULTY OF ADVANCED TECHNOLOGY
UNIVERSITY OF GLAMORGAN
Prifysgol Morgannwg

January 2009
Declaration of Authorship

This is to clarify that, this thesis titled ‘Design and investigation of scalable multicast recursive protocols for wired and wireless ad hoc networks’ and the work presented in it are my own. I confirm that:

✓ Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;

✓ Where I have consulted the published work of others, this is always clearly attributed;

✓ Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;

✓ I have acknowledged all main sources of help;

✓ Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

Signed: …………………………………………………………………………………

Date: …………………………………………………………………………………

Director of studies:

Prof. Khalid Al-Begain.

Director of Integrated Communications Research Centre

Faculty of Advanced Technology

University of Glamorgan, UK
ABSTRACT

The ever-increasing demand on content distribution and media streaming over the Internet has created the need for efficient methods of delivering information. One of the most promising approaches is based on multicasting. However, multicast solutions have to cope with several constraints as well as being able to perform in different environments such as wired, wireless, and ad hoc environments. Additionally, the scale and size of the Internet introduces another dimension of difficulty. Providing scalable multicast for mobile hosts in wireless environment and in mobile ad hoc networks (MANETs) is a challenging problem. In the past few years, several protocols have been proposed to efficient multicast solutions over the Internet, but these protocols did not give efficient solution for the scalability issue. In this thesis, scalable multicast protocols for wired, wireless and wireless ad hoc networks are proposed and evaluated. These protocols share the idea of building up a multicast tree gradually and recursively as join/leave of the multicast group members using a dynamic branching node-based tree (DBT) concept. The DBT uses a pair of branching node messages (BNMs). These messages traverse between a set of dynamically assigned branching node routers (BNRs) to build the multicast tree. In the proposed protocols only the branching node routers (BNRs) carry the state information about their next BNRs rather than the multicast group members, which gives a fixed size of control packet header size as the multicast group size increases, i.e. a good solution to the problem of scalability. Also the process of join/leave of multicast group members is carried out locally which gives low join/leave latency.

The proposed protocols include: Scalable Recursive Multicast protocol (SReM) which is proposed using the DBT concepts mentioned above, Mobile Scalable Recursive Multicast protocol (MoSReM) which is an extension for SReM by taking into consideration the mobility feature in the end hosts and performing an efficient roaming process, and finally, a Scalable Ad hoc Recursive Multicast protocol (SARM) to achieve the mobility feature for all nodes and performing an efficient solution for link recovery because of node movement. By cost analysis and an extensive simulation, the proposed protocols show many positive features like fixed size control messages, being scalable, low end to end delay, high packet rate delivery and low normalized routing overhead. The thesis concludes by discussing the contributions of the proposed protocols on scalable multicast in the Internet society.
ACKNOWLEDGMENT

This thesis would not have reached successful completion without the help, support and guidance of many people whom I am sincerely thankful. Deepest gratitude is to my supervisor Prof. Khalid Al-Begain. It has been an honor to be one of his PhD students. I appreciate all his contributions of time, ideas, and continuous guidance to make my PhD. Also I would like to thank my second supervisor of the supervisory committee, Dr. Gaius Mulley for his assistance in this study.

I am as ever, especially indebted to my parents, for their love and support throughout my life. Moreover, I also wish to thank my wife and kids, for their support and understanding during my study. Finally, my sincere thanks go to my friends, who shared their love and experiences with me.

Firas
CONTENTS

Design and Investigation of Scalable Multicast Recursive Protocols for Wired and Wireless Ad hoc Networks

Declaration of Authorship.................................................................................................. ii

ABSTRACT...................................................................................................................... iii

ACKNOWLEDGMENT................................................................................................... iv

LIST OF TABLES............................................................................................................ ix

LIST OF FIGURES ........................................................................................................... x

ABBREVIATIONS ........................................................................................................xiii

1. INTRODUCTION ..................................................................................1
   1.1 Multicast Applications ................................................................................. 1
   1.2 Multicast routing in MANETs ................................................................. 2
   1.3 Motivation ................................................................................................. 5
   1.4 Summary of the contributions ................................................................. 6
   1.5 Publications from this thesis ................................................................. 7
   1.6 Structure of thesis ................................................................................... 7

2. RESEARCH BACKGROUND .................................................................9
   2.1 Introduction ............................................................................................. 9
   2.2 Definition of Wireless Mobile Ad Hoc Network .................................... 10
   2.3 Features and Applications of Wireless Ad Hoc Network .................... 11
      2.3.1 Military operations ........................................................................ 11
      2.3.2 Search and rescue operation ....................................................... 11
      2.3.3 Bluetooth ................................................................................. 12
2.3.4 Conference room ................................................................. 12
2.3.5 Personal area networks (PANs) ........................................ 12

2.4 Routing in wireless Ad hoc network ......................................... 12
2.4.1 Unicast routing protocol ...................................................... 12
2.4.2 Proactive routing protocol .................................................... 13
2.4.3 Reactive routing protocols ............................................... 15
2.4.4 Hybrid routing protocol .................................................. 23
2.4.5 Multicast ad hoc Routing protocol ................................... 24
2.4.6 Tree based routing protocol .............................................. 25
2.4.7 Mesh based routing protocol ............................................. 27

2.5 Discussion ............................................................................... 30
2.6 Summary ................................................................................. 33

3. RELATED WORK ........................................................................ 34

3.1 Introduction ............................................................................. 34
3.2 Explicit multicast protocols .................................................. 35
3.2.1 Explicit multicast routing protocol (Xcast) .................. 35
3.2.2 Explicit Multicast Extension (Xcast+) ......................... 37
3.2.3 A REcursive UNicast TreE (REUNITE) Protocol ........ 39
3.2.4 Hop By Hop Multicasting routing protocol (HBH) ... 40
3.2.5 Simple Explicit Multicast (SEM) Protocol ................. 42

3.3 Explicit Multicast in MANETs .............................................. 44
3.3.1 Differential Destination Multicast (DDM) Protocol .......... 45
3.3.2 A Scalable Multicast protocol for MANETs .............. 47
3.3.3 An Explicit Multicast protocol for MANETs .......... 50

3.4 Discussion ............................................................................ 52
3.5 Summary .............................................................................. 54
5.3 MoSReM Protocol Description .............................................................. 101
5.4 MoSReM Details .................................................................................. 102
  5.4.1 Roaming messages in MoSREM .................................................... 102
  5.4.2 Roaming messages roles in MoSReM ............................................ 104
5.5 Scalable Ad hoc Recursive Multicast (SARM) protocol ...................... 108
  5.5.1 Mobility messages in SARM .......................................................... 109
5.6 Handling node Movement ................................................................. 111
5.7 Performance evaluation ..................................................................... 113
  5.7.1 Simulation .................................................................................... 113
  5.7.2 Results and discussion ................................................................. 115
5.8 Conclusions ...................................................................................... 120

6. CONCLUSIONS AND FUTURE WORK ........................................... 122
  6.1 General Evaluation .......................................................................... 122
  6.2 Contributions of the thesis .............................................................. 123
  6.3 Future Work ................................................................................... 125

7. REFERENCES .................................................................................... 127
### LIST OF TABLES

Table 2.1 Comparison between different kinds of unicast ad-hoc protocols.......................... 30

Table 2.2 Comparison between Multicast ad-hoc routing protocols ..................................... 32

Table 4.1 Terminologies used in SReM ................................................................................. 58

Table 4.2 Messages used in SReM ......................................................................................... 59

Table 4.3 Cost analysis of SReM, SEM, HBH, REUNITE, Xcast and Xcast+...................... 81

Table 5.1 Simulation Parameters .......................................................................................... 114
LIST OF FIGURES

Figure 2.1 An Ad Hoc network [64]................................................................. 10
Figure 2.2 Unicast-based multicast packet delivery ........................................... 13
Figure 2.3 The Route Discovery implemented at AODV [16]............................ 20
Figure 2.4 Route request propagation in DSR [58].......................................... 21
Figure 2.5 Route reply in DSR [58]................................................................. 22
Figure 2.6 Routing zone for node A with distance P=2 [60]............................. 24
Figure 2.7 Multicast packet delivery .............................................................. 24
Figure 2.8 Node Join operation in MAODV [64]............................................. 26
Figure 2.9 Data Forwarding in ODMRP[64].................................................. 29
Figure 3.1 Xcast packet delivery [10]............................................................ 36
Figure 3.2 Xcast+ data packet delivery [10]................................................... 38
Figure 3.3 E2M packet delivery [7]............................................................... 49
Figure 3.4 Packet Delivery at EM2NET [7].................................................... 51
Figure 4.1 IGMPv3 message format............................................................... 61
Figure 4.2 RqM / RpM message format.......................................................... 62
Figure 4.3 eBNM message format................................................................. 64
Figure 4.4 rBNM message format............................................................... 64
Figure 4.5 MFT entry format....................................................................... 65
Figure 4.6 Joining process in SReM (Case 1)............................................... 67
Figure 4.7 Joining process in SReM (Case 2)............................................... 68
Figure 4.8 Joining process in SReM (Case 3)............................................... 69
Figure 4.9 Leaving process in SReM (Case 1)............................................. 70
Figure 5.5 BNR_Down message format ................................................................. 110
Figure 5.6 Link_failure message format ............................................................ 110
Figure 5.7 Packet header size for each data packet sent ...................................... 116
Figure 5.8 Packet delivery ratio as a function of pause time ................................. 117
Figure 5.9 Packet delivery ratio .......................................................................... 118
Figure 5.10 End to end delay as a function of pause time .................................... 119
Figure 5.11 End to end delay as a function of group size ...................................... 120
ABBREVIATIONS

ACK  Acknowledgment message
AMR  Active Multicast Routers
BNM  Branching Node Messages
BNR_DOWN  Branching Down message
BNR_UP  Branching Up message
BP  Basic Packet
CM  Control Message
DBT  Dynamic Branching node-based Tree
DR  Designated Router
eBNM  Branching Node enquiry Message
FG  Forwarding Group node
FS  Forwarding Set
IGMP  Internet Group Multicast Protocol
IMR  Intermediate Multicast Router
IN  Intercepting Node
JoinM  Join Message
LeaveM  Leave Message
Link_Failure  Link Failure message
LMR  Local Multicast Router
MANETs  Mobile Ad hoc Networks
MCM  Multicast in Confirmed Mode
MCT  Multicast Control Table
MDT  Multicast Destination Table
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFIT</td>
<td>Multicast Forwarding Intercepting</td>
</tr>
<tr>
<td>MFT</td>
<td>Multicast Forwarding Table</td>
</tr>
<tr>
<td>ML</td>
<td>Multicast List</td>
</tr>
<tr>
<td>MP</td>
<td>Multicast Packets</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi Point Reply</td>
</tr>
<tr>
<td>MTI</td>
<td>Multicast Tree Identity</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transfer Unit</td>
</tr>
<tr>
<td>MUCM</td>
<td>Multicasting in Unconfirmed Mode</td>
</tr>
<tr>
<td>NACK</td>
<td>Negative Acknowledgment</td>
</tr>
<tr>
<td>nLMR</td>
<td>new Local Multicast Router</td>
</tr>
<tr>
<td>oLMR</td>
<td>old Local Multicast Router</td>
</tr>
<tr>
<td>rBNM</td>
<td>Branching Node reply Message</td>
</tr>
<tr>
<td>Rm_in</td>
<td>Roaming in message</td>
</tr>
<tr>
<td>Rm_out</td>
<td>Roaming out message</td>
</tr>
<tr>
<td>RpM</td>
<td>Registration reply Message</td>
</tr>
<tr>
<td>RqM</td>
<td>Registration request Message</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply message</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request message</td>
</tr>
<tr>
<td>SFT</td>
<td>Session Forwarding Table</td>
</tr>
<tr>
<td>SMT</td>
<td>Session Membership Table</td>
</tr>
<tr>
<td>TRM</td>
<td>Branching Routers Table</td>
</tr>
<tr>
<td>TTL</td>
<td>Time To Live</td>
</tr>
<tr>
<td>X2M</td>
<td>Xcast to Multicast</td>
</tr>
<tr>
<td>XF</td>
<td>Xcast Forwarder</td>
</tr>
<tr>
<td>XFT</td>
<td>eXplicit Forwarding Table</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Mobile devices, such as laptop computers, Pocket PCs, cellular phones, etc., are now easily affordable, and are becoming more popular in everyday life [42]. As they all become communication devices; which need an infrastructure to play their role. The infrastructures that can be used are wired, wireless cellular or wireless LAN. However, such infrastructure may not be available in all situations. Therefore, there is a need for infrastructureless communication.

A network of mobile hosts without an infrastructure is known as an ad hoc network [12]. According to [42], a mobile ad hoc network (MANET) is defined as follows:

"An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any centralized administration or standard support services regularly available on the wide-area network to which the hosts may normally be connected."

Ad hoc networks were first mainly used for military applications. Since then, they have become increasingly more popular within the computing industry. Applications include emergency search-and-rescue operations, deployment of sensors, conferences, exhibitions, virtual classrooms and operations in environments where construction of infrastructure is difficult or expensive.

1.1 Multicast Applications

As computer and network technologies have developed, conditions have been created by which the transfer of multimedia data in real time can be supported. This means that the development of advanced applications, such as television and videoconferencing via the Internet, has become possible.
Internet videoconferencing and TV belong to the category of group communication unlike others that are consisting of point-to-point communication or file transfers. These new application often have a source and large number of receivers. An efficient multicast service is required [22], since the use of multiple unicast routes is unfeasible in terms of bandwidth, and server processing, since the media, which are mainly audio and video, demand a variety of resources from the communications system.

In traditional group communication, data travels over a data link once, and is replicated by intermediate routers in order to serve those clients who wish to receive the content. This approach is not scalable as it frequently floods the network with data in the case of new receivers.

To coup with the increase in the users, scalable multicast protocols is needed to perform the data transfer to the users, these protocols should dynamically add the new users without a huge traffic or reconstruction of the tree. For example, in videoconferencing application, if a new person wants to share this application, the protocol should be able to start forwarding data to this new user without interrupting the other users and at the same time if any of the users wants to leave this application, the protocol must support a smooth leave to this user and reconfigure the group locally and dynamically.

Applications like videoconferencing, TV and software update is considered as a group communication applications and can benefit from the idea of scalable multicast routing protocols concepts to deliver data from source to the set of receivers and to satisfy the increase in the number of users for these applications. These protocols will dynamically add and remove the receivers without any reconstruction of the multicast tree.

1.2 Multicast routing in MANETs

Routing protocol can be defined as a set of rules that specify how nodes communicate with each other for information exchange. In general, a multi-hop routing protocol is needed in a mobile ad hoc network, because two hosts wishing to exchange packets may not be able to
communicate directly with each other if they are out of radio range [42]. Routing in ad hoc networks has certain challenges for example; the nodes in the ad hoc network can move around with any velocity or direction, this implies that the nodes can move out of and into the range of different nodes. Therefore, the routing protocols in ad hoc network must be able to be adapted to the dynamic changes in the topology of the wireless network. Some routing protocols can keep up to date state information about the topology with periodic update messages. Other routing protocols reduce or remove the state information kept in nodes.

The basic communication mechanism is the transmittal of data between a pair of hosts or devices, one side sending, the other, receiving. This type of communication mechanism is called point to point communication. A telephone call is an example of point to point link between two phones. However, telephone calls are multiplexed into a company trunk; the individual call is part of point to point session. End to end connection refers to a connection between two systems or devices. For example, where the packets are transferred between two routers represents point to point link. Therefore, the Internet connection can be considered of multi point to point links. The other type of communication mechanisms is the point to multipoint mechanism. At this kind of communication, one sender is sending the same information to a set of receivers; these receivers have a common feature(s). For example, a mainframe and its terminals that perform a specific application. The device that provides the multipoint connection is usually an intelligent controller that manages the flow of information from the multiple devices attached to it. One more type of communication mechanisms is multipoint to multipoint, at this type of communication each device is a sender and sending the information to all other devices connected. Each of these devices can be sender and receiver at the same time.

The routing paths between any pair of hosts can be symmetric or asymmetric. For a pair of hosts A and B, if the path from A to B (forward direction) is different from the path from B to A (reverse direction), we say that the pair of paths between A and B exhibit routing asymmetry [33]. On the other hand if the forward path is the similar to the reverse path, we say that the pair of paths is symmetric.
The majority of multicast routing protocol constructs a reverse shortest path tree from the source to the group members [22]. The data packets from the source to the receiver follow the unicast path used to go from the receiver to the source. If these paths have different characteristics, e.g., different delays, the use of symmetric paths may discard the QoS deployment. So the ability to use asymmetric routes will give an improvement to the multicast routing protocols.

Routing protocols for wireless ad hoc networks are classified into: unicast, multicast or broadcast routing protocols [71]. In unicast the source node sends a separate copy of each packet for each receiver, while multicast routing protocols process is to send one copy from the source to a group of receivers. Those receivers share the same group number and run the same application. Broadcasting is the process of sending the packets without considering the number or the nature of receivers. Upon receiving packets, the intermediate nodes just rebroadcast the packet until hopefully all the receivers will receive this packet.

In typical ad hoc networks, hosts work in groups to carry out a given task. Hence, multicast plays an important role in ad hoc networks. In the past few years, a number of multicast ad hoc routing protocols have been proposed. Depending on the method of data propagation, these routing protocols are classified into two types:

- **Tree-Based multicast routing protocols**, a spanning tree is built to connect the group nodes with other intermediate nodes to help in forwarding packets between group members. This type of routing protocols guarantees the loop free feature. Examples regarding this type are Multicast Ad hoc On-demand Distance Vector protocol (MAODV) [57], Ad-hoc Multicast routing protocol using incrementing Id-numbers (AMIRS) [68] and Ad-hoc Multicast Routing protocol (AMRoute).

- **Mesh-based multicast routing protocols**, a mesh is built to connect the group members. These protocols will give redundant routes between group members. These redundant routes will reduce the packet delivery loss because of link failure.
Examples on this kind of routing protocols are On-demand Multicast Routing Protocol (ODMRP) [47] and Core Assisted Mesh Protocol (CAMP) [14].

These protocols follow the concepts of the traditional multicast protocols; the multicast routing state maintenance and shared group membership management are stored in the intermediate group members and updated periodically. If applied for use with large number of small groups within the network these protocols may become more complex and less efficient.

To solve this issue, recently there has been a shift toward the stateless multicast routing protocols for small group networks and for MANETs. Some of these protocols were developed to avoid or reduce any multicast routing state to be maintained by routers. This is done by using explicit messages that encode receiver’s addresses or a set of group members addresses in a header and the later is included in an ordinary unicast routing packet.

Examples of this type of multicast routing protocols are Differential Destination Multicast (DDM) [20] Protocol, A Scalable Multicast protocol for MANETs (E2M) [38] and An Explicit Multicast protocol for MANETs (EM2NET) [7].

A detailed discussion for the routing protocols previously mentioned above will be included in the next two chapters of literature review and related work.

1.3 Motivation

Wireless ad-hoc networks have gained considerable importance in wireless communications. Wireless communication is established by nodes acting as routers and transferring packets from one to another in ad-hoc networks. Routing in these networks is highly complex due to the moving nodes and the difficulty of keeping state information about the routers; hence in the past few years a shift towards the stateless multicast routing protocols has happened.
The purposes of this thesis are:-

- To investigate and survey different solutions providing scalable multicast routing protocols for wired and wireless ad hoc networks.

- To design a scalable multicast routing protocol to be used for wired networks for Internet applications. The proposed protocol aims to improve the scalability feature in the wired networks.

- To perform an extensive performance evaluation including a cost analysis and an implementation study for the proposed protocol. Furthermore, an investigation and an extensive analysis for the proposed protocol performing different scenarios to test the scalability feature and comparing the results with previously proposed protocols.

- To design and investigate the mobility feature when applied for the receivers to adapt the roaming process.

- To design scalable multicast ad hoc protocol to be used in ad hoc environment Then implement and analyze the proposed protocol by running different scenarios, analyzing the obtained results and comparing these results with the most recent proposed protocols in this area of study.

1.4 Summary of the contributions

The contributions of this thesis are as follows:

- A detailed discussion and investigation of previously proposed scalable protocols for wired and wireless ad hoc networks. This includes a comparison between these protocols taking into consideration the scalability feature.

- Investigate and introduce the idea of scalable multicast wired protocol started by members in our group of research. The completion includes a detailed investigation of the protocol in addition to detailed description of data structure of this protocol.

- A detailed cost performance and simulation evaluations are done with comparison to other protocols in the same area of research.

- Propose a new scalable protocol to support the mobility feature in receivers. This protocol guarantees the roaming process of receivers.

- As many users are carrying various types of mobile devices, such as laptops, Pocket PCs, cell phones, etc., there are increasing needs to communicate with each other
without any aid of infrastructure. This thesis designed and described a new scalable ad hoc recursive multicast protocol, which supports the scalability feature in ad hoc networks efficiently.

1.5 Publications from this thesis


1.6 Structure of thesis

Further to this chapter, there are five chapters that are organized as follows.

Chapter 2 describes the background related to this thesis study area, In particular, knowledge background for readers to facilitate understanding the ad hoc environment and its protocols, including the main features and applications for ad hoc (Section 2.1), the different types of ad routing protocols (unicast, multicast) are defined with subtypes for each with examples (Section 2.4). A discussion and comparison of the previous explained examples of routing protocols (Section 2.5), Section 2.6 whilst summarizes Chapter 2.

Chapter 3 introduces the related work to our proposed algorithms including explicit multicast protocols for wired networks (Section 3.2) and explicit multicast protocols for MANETs (Section 3.3). (Section 3.4) introduces a discussion of the previously proposed protocols in explicit multicast, followed by a summary for the chapter found in (Section 3.5).

Chapter 4 discusses the scalable recursive multicast protocol, including an overview of the proposed protocol (Section 4.2), building the multicast tree by discussing the membership
management (Section 4.3), joining and leaving of members (Section 4.4). To deliver data packet a header should be added to the data packet header, this new header is discussed in (Section 4.5). A performance evaluation including cost analysis and simulator based simulation is presented with discussion and evaluation in (Section 4.6), while (Section 4.7) summarizes this chapter with its results.

Chapter 5 presents the new protocols proposed for mobility feature over scalable multicast protocol starting by proposing and describing new protocol for mobility feature applied for receivers (Section 5.2), the proposed protocol details are widely discussed in (Section 5.3). In (Section 5.4) a description of new scalable multicast ad hoc protocol is introduced. The data structure details and different messages structure is discussed in (Section 5.5). Then discussing how the proposed protocol handles the node mobility feature in (Section 5.6). The performance evaluation with results, comparison and analysis are discussed in (Section 5.7). The last section summarizes this chapter.

In chapter 6, the thesis is concluded by briefly revising the main arguments that influenced the design of scalable multicast protocols for wired and wireless ad hoc protocols. Results are summarized and their improvements in the wired and wireless ad hoc networks are presented. Finally the thesis discusses future work orientations.
Chapter 2

RESEARCH BACKGROUND

In this chapter, the different types of routing protocols (Unicast and Multicast) proposed and developed for Ad hoc networks are discussed. At the beginning of this chapter a general description of ad hoc networks, their applications and special features over traditional wireless network are discussed. The different types of routing protocols for Mobile Ad hoc Networks (MANETs) with examples of each type are then surveyed. Some examples will be discussed in details as they will be used as a part of our research work.

2.1 Introduction

As computer applications have become massively embedded in our daily tasks we therefore want these applications to be available “all the time everywhere”. A number of reasons have helped in supporting the previous fact: the rapid decrease in the size of mobile devices, the rise in mobile computer laptops specifications, the huge availability of applications on mobile devices, and the growth in the markets for wireless telephones and mobile devices.

As wireless devices become widely used and the applications using the Internet on these devices are also abundantly available, customers still require using these network applications at any place and at any time even when neither the Internet nor the infrastructure is available. For example, during military operations military units in a battlefield using wireless devices, require keep contact with each other to exchange information forming a network in the absence of base stations [72]. In the case of infrastructure absence, the only way is to establish a wireless network is by distributing routing function of base station (routing) to the wireless nodes and establish a temporary network- in other words Mobile Ad hoc networks (MANETs) and sometimes called ad-hoc networks or multihop wireless networks.
This chapter gives an overview of ad hoc networks, and different types of ad hoc routing protocols.

2.2 Definition of Wireless Mobile Ad Hoc Network

Mobile Ad-hoc network (MANET) is a collection of wireless mobile nodes forming temporary network without the aid of any stand-alone infrastructure or centralized administration [43]. Each node has wireless interface to establish connection and information exchange with other nodes. Examples of possible Ad-Hoc nodes are laptops and Personal Digital Assistants (PDA). Figure 2.1 shows a simple ad hoc network, which consists of mobile devices forming a network without any infrastructure..

![Figure 2.1 An Ad Hoc network [64].](image)

In ad-hoc networks, each wireless node has a transmission range. If all the nodes in the ad hoc network are in the range of each other there is no need of routing protocol to exchange data in between. However, if some of the ad hoc nodes are not within each others range then
an intermediate router or routers are needed to help in forwarding the data among nodes. In the absence of infrastructure in the ad hoc network, the router will be one of the nodes in the network, thus every node in the network will act as host and as router. As a router, the node should be able to forward data and run routing protocol. At the same time the node acts as a host with an IP address in a traditional sense.

The wireless ad hoc nodes are often mobile. Two nodes at any time may be within each other range, however due to their mobility the two nodes might be outside its counterpart range. Change in topology related to node mobility may break the links between nodes and therefore other links should be established. The paths between two nodes may consist of many links passing over several nodes in the middle: a case that is called “multihop routing”.

2.3 Features and Applications of Wireless Ad Hoc Network

Wireless ad hoc networks are suitable for applications where there is a lack of infrastructure or establishment of an infrastructural network is cost effective. Examples of wireless ad hoc networks applications: Military operations, search and rescue operations, Bluetooth, conference room and Personal Area Networks (PAN). The purpose of this section is to highlight these examples.

2.3.1 Military operations

These kinds of operations are considered the primary application for ad hoc networks. The military operations require high survivability, reliability and fast communication recovery, but in an environment that does not possess infrastructure. Ad hoc network is the best solution for this application considering these essential features.

2.3.2 Search and rescue operation

Ad hoc network is very beneficial when the infrastructure is damaged because of natural disasters. Medical and other supporting teams working on disaster recovery need fast and effective communications where they do not have time to wait establishment wired network to start working in the area of disaster, and the nature of work during the disaster recovery requires a flexible communication network. Ad hoc network can establish communication network among disaster recovery teams very quickly and with flexible network topology.
2.3.3 Bluetooth

Bluetooth is one of the commercial applications that utilized the concepts of ad hoc network [31]. It can enable low-cost short-time links [9] between wireless devices without depending on any infrastructure service for communication; hence, ad hoc network technology is the best solution [9].

2.3.4 Conference room

Conferences or meetings are usually held outside the ordinary offices, consequently there is a need to connect the mobile devices for effective data and information exchange. Thus, the infrastructure is missing and a temporal network is required. In such situation, it is clear that you need to set up an ad hoc network to fit.

2.3.5 Personal area networks (PANs)

Personal Area networks (PANS) are interconnection of technological devices close to human body (e.g PDAs, Cellular phones, and pagers) that can exchange information in between. Methods for connecting and exchanging data among these nodes for PANs can exploit the benefits of ad hoc networks technologies.

2.4 Routing in wireless Ad hoc network

Given the special characteristics of ad hoc networks, routing protocols that adapt to these characteristics are essential. The dynamic topology is the most important characteristic which comes because of node mobility. Nodes may change their positions quickly and frequently, which means that routing protocols are required to adapt these changes. Routing protocols for wireless ad hoc networks are classified into: unicast, multicast or broadcast routing protocols [71]. Each group has similar and different routing features. In this section these routing protocol classifications will be discussed in more details.

2.4.1 Unicast routing protocol

The procedure of delivering information (data or control packets) from one node (Source node) to another node (Destination node) is called Unicast routing. When there is more than one destination the source needs to send a number of copies equal to the number of
destinations. Figure 2.2 shows the unicast operation where the source needs to send three copies of the same packet to be received by the three destinations.

![Figure 2.2 Unicast-based multicast packet delivery](image)

The source node knows, before sending the packets (data or control), the final destination node address which should receive these packets. These protocols can be classified into: proactive, reactive and hybrid routing protocols, which they differ in the way that the intermediate nodes -located between the source and the destination- obtain the routing information.

### 2.4.2 Proactive routing protocol

In proactive routing protocols, each node maintains routing information about all other nodes (or nodes in an exact division) in the network. This information is kept in tables, which is updated periodically. Different routing protocols have different tables according to the way each of these protocols works. The advantage of proactive routing protocols is that the source needs the route to be ready; the routes are ready and stored so that there will not be any latency. The problem with this kind of protocols, however, is that some of the stored routes may never be used. Another problem comes when the network topology is changing very fast leading to consuming a scarce wireless network bandwidth to keep the route information up to date. Examples of this kind of ad hoc routing protocols: Destination Sequenced Distance
Vector (DSDV) [53], Optimized Link State Routing (OLSR) [40], and Source Tree Adaptive Routing Protocol (STAR) [29]. This section describes a number of proactive routing protocols.

### 2.4.2.1 Destination Sequence Distance Vector (DSDV) protocol

The Destination Sequence Distance Vector (DSDV) [53] is a hop-by-hop proactive unicast routing protocol where each node maintains a routing table. Like distance vector, DSDV keep the entries of the routing table fresh by periodically broadcast routing updates. The main advantage of DSDV over traditional distance vector protocols is being loop free protocol.

In each node the routing table in DSDV protocol stores entries for all reachable destinations for the designated node, each entry consists of next hop, a number of hops for that destination and a sequence number created by the destination. Sequence numbers in DSDV protocol, used to show the freshness of routes in the routing table for the destinations and guarantee the loop free feature in DSDV.

The route updates of DSDV can be either time-driven or event-driven. In time-driven every node periodically transmits updates including its routing information to its immediate neighbours. Whereas in event-driven updates, a node detects significant changes in its route information from the last update it sends an update for the routing information in an event-triggered style.

As previously explained, DSDV is a distance vector with some adjustments to be suitable for ad hoc use. These adjustments are the use of event-triggered update among the broadcast (time-event) updates. Moreover, DSDV uses two types of update messages: full dump and incremental dump. A full dump message carries all information available for update, where incremental dump message carries only the information about routes which their metrics changed from the last update.
2.4.2.2 Optimized Link State Routing (OLSR)

OLSR [18] is a proactive unicast ad hoc protocol, based on traditional link state algorithm. At this kind of protocols each node keeps and updates information about the link with the other nodes stored in a table. The main contribution of OLSR [18] is that it reduces both the number of re-broadcasting update messages and the size of these messages by setting a multi point reply (MPR) node strategy. Each node during the update strategy selects a set of one hop neighbouring nodes as MPRs. The main responsibility of these MPRs is to re-broadcast the update messages from the node. Other nodes that are not MPRs can read and process messages without retransmitting.

To select the MPRs, each node periodically sends a list of its one hop neighbours using a Hello messages, and from the list included in Hello messages, each node selects a set of one hop neighbours to cover its two hop neighbours in the network.

2.4.3 Reactive routing protocols

Reactive (On-demand) routing protocols attempt to reduce the overhead in proactive routing protocols, in which a node starts searching for route information only when this node has a packet to be sent to a specific destination [3]. These kinds of protocols usually consist of two types of processes: route discovery and route maintenance. This means that a node has packet(s) to send for a specific destination, it starts a route discovery by flooding a route request over the network. After a period of time the source node will receive one or more route reply with information about the routes available to the destination node. The source node upon receiving these routes will select the one which is suitable for the data to send according to number of predefined features. The chosen route will be used to transmit the packets and this route will be maintained by the source during the transmission to ensure that the packets will reach the destination. Route maintenance occurs after the establishment of routes, this process continually update the routes until either the destination node becomes unreachable from any path from the source or the route becomes undesired [58].

According to [3], On-Demand routing protocols can be classified into two types: source routing [10] and hop-by-hop routing protocols. In the first type of reactive routing protocol,
the header of the data packet will carry the whole address information from the source to the destination. The intermediate nodes do not require maintaining any up-to-date routing information about the neighbourhood nodes to forward the data packet, but they just use the information kept in the header of the data packet to forward the data packet. The main disadvantage of this type of reactive routing protocols is that these protocols did not perform well on large networks [3]. This drawback will happen because at large networks the header of data packet needs to maintain routing information for a large number of intermediate nodes which will cause a huge overhead at the data packet delivery. Another reason for this drawback over large networks is that the data packet will be forwarded to a large number of intermediate nodes which means high probability of link failure at these nodes, in other words high probability of active route failure. For that reason, in large networks the hop-by-hop reactive routing protocol is more commonly used. The header of the data packet carries the destination and the next hop address, as soon as the data packet reaches the intermediate node this node update the header of the data packet by inserting the address of the new next hop and forwards data packets. All the intermediate nodes perform similarly until the data packet reaches its destination node.

This type of reactive routing protocols has the advantage of low volume of overhead information at the data packet header and it will use better and fresh route to the destination. On the other hand the disadvantage of this type of routing protocols is that each node in the network should keep fresh information about active routes and fresh information about its neighbours which means that each node will send periodical messages to keep fresh information which will cause high overhead over the network.

Examples of these kind of protocols: Ad hoc on demand Distance Vector routing protocol (AODV) [54], Dynamic Source Routing Protocol (DSR) [43] and Temporally-Ordered Routing Algorithm (TORA) [51]. This section describes a number of reactive routing protocols.
2.4.3.1  Ad hoc on demand distance vector routing protocol (AODV)

AODV [16, 56, 54] is a reactive ad hoc protocol, that can be considered as a combination of DSDV [53] and DSR [43]. It takes the on-demand mechanism for route discovery and route maintenance from DSR [43], while the hop-by-hop and sequence number is taken from DSDV [53].

AODV uses different set of messages for route discovery and route maintenance. Whenever a node has data packets and needs to send these packets to another node it broadcasts a Route Request message (RREQ) to all its neighbours. This RREQ message propagates through the network until reaches the destination node or an intermediate node that posses a fresh route to the destination. The route will be ready by unicasting a Route Reply message (RREP) to the source from the destination.

Another kind of messages used by this algorithm is Hello messages. These messages are sent periodically in order to keep each node and its neighbours connected and maintain the routes valid. If one node does not receive a Hello message from its neighbour node, the later assumes that its neighbour has moved away and marks that link to the node as broken then notifies the affected nodes by sending a link failure notification to them.

The updated information about routes is kept in a table, this table called a route table. The structure of this table and the operation implemented in AODV are as follows:-

- **Route table management.**

AODV maintains a route table at each node; the following information should remain tracked for each route at this table:

- Destination sequence number: Sequence number for the destination known to this node.
- Destination IP address: the destination node IP address.
- Next Hop: the neighbour node for this route toward the destination.
- Hop Count: number of hops to reach the destination.
- Life Time: the amount of time this route is valid.
- Request Buffer (flag): a flag to show if this node passes on the request message or not.

- Route discovery
When a node needs to transmit data packets to a destination, it broadcasts a RREQ message if the route to the destination is unknown or the previous route expires. After RREQ broadcast, the node waits a period of time in order to receive a RREP for its request. If the node does not receive a reply it may rebroadcast the RREQ again or assumes that the destination is no longer exists.

If there is no valid route for the destination carried by the RREQ at the intermediate nodes, the intermediate nodes forward the RREQ by broadcasting. At the same time these nodes keep a reverse route which contains the source IP address and the IP address from the RREQ received. This reverse route is a temporary route in order to be used later to form a RREP to the source. Figure 2.3 (a) shows the propagation of RREQ from source node H to find out a route to the destination C, the message will pass through the intermediate nodes (F, G and D).

A node generates RREP if it is either the destination node or a node with valid route to the destination. While RREP is forwarded back, a route for the source node is created to the destination and the route is ready for sending data packets. Figure 2.3 (b) shows the RREP message sent by the destination (D) upon receiving the RREQ message from the source H.

- Route maintenance
A node will detect a link failure to one of its neighbours by link layer messages or by periodical Hello messages. As soon as the node detects this failure it will remove the route entry for that neighbour and creates a route failure message, then it sends this message to all neighbouring nodes that are actively using this route, informing them that this route is no longer valid. Each node in AODV protocol keeps a list of active neighbours in order to keep track of neighbours that are using particular list of neighbours.
The AODV protocol is one of the most well-known unicast ad hoc protocols. The advantage of this protocol in comparison to the classical distance vector and link state routing protocols is that it uses less number of messages because of its reactive nature. Another advantage is that it is loop free protocol because it uses the idea of sequence numbers for messages. On the other hand the use of sequence number may cause some difficulties; suppose this sequence number becomes no longer synchronized or wrapped around which will cause a lot of conflicts.

(a) Propagation of the route_request message
2.4.3.2 Dynamic Source Routing Protocol (DSR)

DSR [43] is a reactive ad hoc routing protocol based on the source routing concepts. Each node maintains a route cache to keep previously known routes to this node, where the information kept in each node will expire if it is not used for a predefined period of time.

When a node needs to send data to a destination, it searches in its cache for a route to that destination, if the route exists, then the node starts sending data to that destination. Otherwise, a route request message as part of route discovery procedure will be created and propagated in the network. The intermediate node upon receiving the route request message searches in its cache for a route to this destination. If the route to the destination not found, it appends its address to the route request message and propagates the message to its neighbours.

Figure 2.3 The Route Discovery implemented at AODV [16].

(b) Sending the route_reply message backward to the source
Figure 2.4 describes an example of route request in DSR. The source N1 needs to find a route for the destination N8. This route is not available in the source cache, so it propagates a route request message in the network. The intermediate nodes (N2 to N7) rebroadcast the message, during the rebroadcast each message adapt its address in the path toward the destination. Finally, the destination will receive the route request message from its neighbours; each route request contains a list of intermediate nodes which represent the path from the source. The destination N8 will select the shortest paths (minimum number of hops). The route request message contains the source address, the destination address and a unique identification number. The later found at route request message limits the number of message propagation in the intermediate nodes, which processes the route request if it has not been seen before, which at the end will reduce the rebroadcasting of route request message.

Whenever the route discovery is successful (Reached the destination or an intermediate node has a valid route to the destination), the source will receive a route reply message, which contains a list of intermediate nodes addresses by which the destination is reachable.

The route reply is generated by an intermediate node which has a valid route to the destination or by the destination itself. If the route reply is generated by the destination then the contents of route request message is placed in the route reply message. On the other hand,
if the intermediate node is generating the route reply message then it appends its destination route cache into the route request message and inserts this information into the route reply message.

![Figure 2.5 Route reply in DSR](image)

Figure 2.5 shows an example of route reply message generated by the destination N8 in response of a route request generated at the source N1 explained at Figure 2.5. The route reply message contains the addresses of the intermediate nodes (N2, N5) in addition to the source address (N1) and the destination address itself (N8).

The advantages of DSR are

- The implemented cache in this protocol can keep more than one route to the destination, the cache contents are ready to be used as soon as the source has packets to send, without latency time.
- The broadcasting of route request in this protocol is limited, because the intermediate nodes will not retransmit the same route request.

The main disadvantage of DSR is that it does not scale well with large and dense networks because the route request message should carry the information about the intermediate node, which may become a burden for large networks.
The route maintenance process at DSR is divided into two messages: route error and acknowledgment. Route error message is generated when a node encounters fatal transmission problem, while the acknowledgment message is created when correct operation of link route is achieved.

2.4.4 Hybrid routing protocol

Many challenges face the routing protocols proposed for mobile ad hoc networks such as frequently changed topology, low transmission power and asymmetric link. Proactive and reactive routing protocols were proven to be insufficient to overcome these challenges when operate separately. Hybrid ad hoc routing protocols are proactive and reactive routing protocols at the same time, which are designed to work as proactive routing protocols for the close destinations from the source and act as reactive routing protocols for far destinations. These types of routing protocols are designed to reduce the overhead caused by route discovery through allowing the close nodes to work together. These kinds of routing protocols are also called Zone-Based routing protocols as they divide the network into zones and nodes in each zone work together and communicate with other zones on demand. Zone Routing Protocol (ZRP) [32] is an example of this kind of unicast ad hoc routing protocols. This next section will describe ZRP as an example of routing protocols related to this type.

2.4.4.1 Zone routing protocol (ZRP)

ZRP [32] is developed to combine the advantages of reactive and proactive routing protocols into one type of routing protocols i.e hybrid routing protocols.

In ZRP, a routing zone parameter (of radius P) is defined for each node; this parameter defines the distance in number of hops from each node where within this zone the routing information is available. For destinations outside this zone the ZRP performs a route discovery using any of on-demand unicast routing protocols exploiting the information available from the local zone. Figure 2.6 shows the node A with routing zone P=2.
2.4.5 Multicast ad hoc Routing protocol

Multicast routing is the task of sending one copy of data packet from single source to multiple receivers (source specific) or from multi senders to multiple receivers (Group shared) [59]. Figure 2.7 shows multicast packet delivery where the source needs only to send one copy of the packet to be received by all destinations.
The nature of ad hoc applications discussed in Section 2.3, requires hosts to work in groups to carry out a specific task. Therefore multicast networks are the best solution for this kind of ad hoc application. Traditional wireless multicast protocols can not usually be used for ad hoc application due to the frequent change of topology and the unstable wireless environment.

In recent years, many different multicast ad hoc protocols have been proposed. All these protocols are classified according to the way they propagate data into tree-based or mesh-based ad hoc routing protocols. While mesh-based protocols send data through the network and a subset of nodes are responsible of forwarding these data for all the group members, tree-based protocols propagates data over an established tree holding all the multicast group members.

Multicast Ad hoc On-demand Distance Vector protocol (MAODV) [57], Ad-hoc Multicast Routing protocol using incrementing Id-numbers (AMRIS) [68], Preferred Link Based Multicast Protocol (PLBM) [62], Reservation-Based multicast routing Protocol (RBM) [21] and Ad-hoc Multicast Routing protocol (AMRoute) [69] are examples of tree based ad hoc routing protocols.

Protocols like On-demand Multicast Routing Protocol (ODMRP) [47], Forwarding Group Mesh Protocol (FGMP) [17], Core Assisted Mesh Protocol (CAMP) [6, 23] and Dynamic Core-Based Multicast Routing Protocol (DCMP) [23] are examples of mesh-based ad hoc routing protocol.

The previously mentioned types of multicast ad hoc protocols (i.e Tree-based, Mesh-based) will be discussed in detail in the next two sections.

2.4.6 Tree based routing protocols

Using this type of multicast ad hoc protocols, a spanning tree is built to connect the multicast group members (nodes). The spanning tree algorithm prevents loops in a tree. For ad hoc environment after building the tree to connect the group nodes the data packets can flood
through the tree without any loops or duplicates transmission. The subsections will provide brief description of the most popular tree-based multicast ad hoc routing protocols.

2.4.6.1 Multicast Ad hoc On-demand Distance Vector protocol (MAODV)

MAODV [57] is a multicast extension of the AODV [54] routing protocol. MAODV follows the way AODV establishes the unicast route with multicast capabilities. MAODV is a tree-based algorithm, so it establishes a shared tree to connect the group members. Each multicast group has a group leader taking responsibility for establishing and maintaining a destination sequence number and broadcasting a (Grp-Hello) message periodically to maintain the group members connected.

Building a tree at MAODV is done incrementally and in a receiver-initiated way. If a node needs to join a multicast group, it initiates and propagates RREQ message with joined flag set. A node within the same group that has a valid route to the destination will send back a RREP message and inform the group leader with the information about the new node. The prospective node that sent the RREQ may receive more than one RREP, so the RREP message with the highest sequence number and minimum number of hops will be selected. This process will ensure that the new member is added without any loop. Figure 2.8 shows the operations of adding a prospective node to a group using MAODV protocol.

![Figure 2.8 Node Join operation in MAODV](image)

Figure 2.8 Node Join operation in MAODV [64].
2.4.6.2 Ad-hoc Multicast routing protocol using incrementing Id-numbers

Ad-hoc Multicast routing protocol using incrementing Id-numbers AMRIS [68] is a tree-based multicast ad hoc routing protocol, designed to run independently of underlying routing protocol. It builds a group-shared tree for multicast packet delivery. The key idea for AMRIS is the use of multicast session member id msm-id. The msm-id is used to direct the flow of data and indicate the logical level of nodes in the multicast shared tree. The root will get the smallest msm-id within the members of the group; this special rooted node is called s-id. To construct a group-shared tree, the s-id broadcast a new-session packet contains its msm-id. Neighbouring node receiving this packet will create there own msm-id which should be greater than the received msm-id but not consecutive. This propagation process is repeated by the neighbour nodes so that the msm-id’s get greater for the nodes far from the s-id.

Like other multicast tree-based routing algorithms, when a node needs to join the group, it sends Join-Req to its potential parent which has smaller msm-id. If the parent node is a group member, it will respond by a Join-Ack message, otherwise it will forward the message to its potential parents. As soon as the Join-Req message initiator receives a Join-Ack this means it is now part of the multicast shared tree. Otherwise if this node received a Join-NAck or did not receive any messages within a predefined period of time it initiates a Branch-Reconstruction phase until being able to join the multicast group.

2.4.7 Mesh based routing protocol

In the presence of high mobility in the network, the tree-based multicast structure will not perform optimally because it is fragile and requires to be reconfigured frequently. Reconfiguration will incur large control traffic. Furthermore, the traffic will be enforced to stop until the reconfiguration is finished and which may cause a delay and a loss of data packets.

Mesh-based multicast routing protocols have overcome these drawbacks. At mesh-based a construction of mesh for each group offers redundant paths among the group members. The subsection will give brief description of the most common mesh-based multicast ad hoc protocols.
2.4.7.1 On-demand Multicast Routing Protocol (ODMRP)

ODMRP [47] is a source initiated on-demand mesh-based multicast ad hoc protocol. It is on-demand and source-initiated because the source establishes and updates the group membership and multicast routes whenever it has data to send. This protocol builds a mesh providing multiple routes, so multicast data packets will be delivered in case of node movement and topology changes. The concept of forwarding groups [17] is used to establish the mesh.

When a multicast sender wants to send data packets and no route information is available, it generates a JOIN DATA packet with data payload attached and then broadcasts it. When a neighbour node receives the JOIN DATA packet it stores the upstream node ID into its routing table for reverse path construction and rebroadcast the packet. When the JOIN DATA packet reaches the multicast receiver, it generates a JOIN TABLE packet and broadcast it to its neighbours. The JOIN TABLE packet contains a multicast group address, list of all senders whose JOIN DATA were received and the neighbour nodes which were used as a hop toward each sender. When a node receives the JOIN TABLE it checks if one of the next nodes ID matches its own ID, if it does, then this node is a forwarding group (FG) for the source. Then this node broadcast its own JOIN TABLE which contains matched entries. This process builds (or updates) the route from source to the receivers and builds a mesh of nodes and forwarding group nodes. The main advantage of using forwarding group at ODMRP is that it is less affected by link failure due to node movement. Figure 2.9 shows the forwarding data flow in ODMRP.
2.4.7.2 Core Assisted Mesh Protocol (CAMP)

CAMP [27] extends the receiver-initiated approach for Core Based Tree (CBT) [6] to form a multicast mesh. It also builds and maintains a multicast mesh for distributing information within each multicast group. A multicast mesh is a set of nodes within the network topology that insures at least a path from the source to each receiver. Within the multicast mesh of a group, CAMP guarantee forwarding data from any source in the group along the reverse shortest path to the source. One or more cores can be defined for each mesh. CAMP uses cores to limit the traffic required for a node needs to join the multicast group. CAMP provides loop-free packet forwarded over meshes.

CAMP [27] is considered the first multicast protocol that uses routing structure other than tree structure [28]. One possible disadvantage of CAMP is that it needs an underlying unicast protocol like WRP [50] for providing unicast routes to all destinations in the network.
2.5 Discussion

This chapter has surveyed the concepts of wireless ad hoc networks and the different types of routing protocols related to it, Unicast and Multicast. This section aims to perform a comparative discussion on those types of ad hoc routing protocols.

Table 2.1 Comparison between different kinds of unicast ad-hoc protocols.

<table>
<thead>
<tr>
<th>Routing Class</th>
<th>Proactive</th>
<th>Reactive</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing structure</td>
<td>Flat and hierarchical are available</td>
<td>Mostly flat</td>
<td>Mostly hierarchical</td>
</tr>
<tr>
<td>Availability of routes</td>
<td>Always available</td>
<td>On-demand</td>
<td>Depends on node location</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(in zone or out of zone)</td>
</tr>
<tr>
<td>Periodic route update</td>
<td>Mostly used</td>
<td>Rarely used, sometimes for local repair</td>
<td>Usually</td>
</tr>
<tr>
<td>Handling mobility effect</td>
<td>Using periodical messages</td>
<td>Different scenarios, AODV uses local route discovery</td>
<td>More than one path especially between zones. No single point of failure.</td>
</tr>
<tr>
<td>Storage</td>
<td>High requirements</td>
<td>Lower than proactive, depends on the number of routes required stored in cache or small tables.</td>
<td>Depends on the size of the zone, big zones means large number of nodes at the end it is as proactive; high storage</td>
</tr>
<tr>
<td>Delay</td>
<td>Small as the routes are available (small response time)</td>
<td>Bigger than proactive, need time to establish the route (response time)</td>
<td>Between proactive and reactive. Small for nodes in the zone and become large for nodes outside the zone</td>
</tr>
</tbody>
</table>

Table 2.1 summarizes a comparison of the three categories of unicast protocols discussed in this chapter, proactive, reactive and hybrid routing protocols. The tradeoffs between proactive and reactive is quite complex. To decide which approach are better many factors should be considered, such as the size of the network, the mobility, the traffic load and so on. DSDV and OLSR discussed before are proactive routing protocols; which is suitable for networks that topology change is small. This kind of protocols has an advantage that the routes are immediately available and stored in tables which will reduce the response time. On the other hand, they cause large amount of storage and high routing overhead as they use periodical messages to keep information stored in nodes up to date. OLSR surpasses DSDV.
by using the Multi Point Rely (MPR) to reduce the routing overhead. At the same time, DSDV uses an event-driven style to reduce the number of periodical messages.

AODV and DSR are reactive routing protocols; which is suitable for networks with light traffic load. An advantage of this kind of routing protocols is the small amount of routing overhead compared to proactive routing protocols. This is due to that reactive routing protocols are on-demand routing protocols; the route established when needed. On the other hand, the protocols have large response time as they usually are not ready when needed. DSR overcomes AODV by using fast storage mechanism to store the most recent routes used leading to more than one route that may be available to the destination. However, AODV supports local connectivity among neighbouring nodes by using periodical small sized hello messages, this issue is important for MANETs environments.

Taking the advantages of the previous kinds, a third type of unicast routing protocol is proposed called hybrid routing protocols. ZRP is an example of this kind of routing protocols; it proposed to work as a proactive for the nodes in a zone with radius P where P the number of nodes far from the source, and works as reactive for outside the zone. It provides higher scalability than pure proactive and pure reactive routing protocols. The scalability is achieved by this kind of routing protocols by grouping the related nodes into zones and each zone communicates with other zones using border nodes. Another advantage of ZRP as a hybrid routing protocol is that there is no one point of failure between zones, as any of the border nodes can establish a connection with other zones on-demand. However, hybrid protocols require complex management in the case of high node mobility, which will cause frequent changes for the zone members causing a high overhead. This overhead because this type of protocols needs to keep information about the nodes in zone every time the zone members changed.

The basic idea behind designing multicast routing protocols for ad-hoc networks is to establish and maintain a connection between group members efficiently and with less effort. Table 2.2 summarizes features of protocols belonging to this type of ad-hoc routing
protocols. Depending on the way the group members are connected, multicast ad-hoc routing protocols can be categorized into tree-based and mesh-based routing protocols.

**Table 2.2 Comparison between Multicast ad-hoc routing protocols**

<table>
<thead>
<tr>
<th>Routing Protocol</th>
<th>MAODV</th>
<th>AMRIS</th>
<th>ODMRP</th>
<th>CAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast structure</td>
<td>Core based tree</td>
<td>Tree</td>
<td>Mesh</td>
<td>Mesh</td>
</tr>
<tr>
<td>Routing information exchange &amp; maintenance</td>
<td>Reactive</td>
<td>Proactive</td>
<td>Reactive</td>
<td>Proactive</td>
</tr>
<tr>
<td>Periodical messages required</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dependency on unicast routing protocol</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Routing hierarchy</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Scalability</td>
<td>Fair</td>
<td>Fair</td>
<td>Fair</td>
<td>Good</td>
</tr>
</tbody>
</table>

MAODV and AMRIS discussed in more details earlier in this chapter are examples of tree-based multicast ad-hoc routing protocols. MAODV is an extension for AODV, which is considered as a reactive multicast routing protocol, it uses small size periodical (Grp-Hello) messages to keep routes fresh. In AMRIS, there is no dependency on unicast routing protocol, consequently, it can be considered as a proactive multicast routing protocol which causes high overhead in comparison with MAODV. Both protocols give a fair solution to scalability issue.

ODMRP and CAMP are considered as mesh-based multicast ad-hoc routing protocols. The mesh-based provides redundant paths from source to destinations among the group. ODMRP can be considered as reactive multicast routing protocol that uses periodical messages to keep group members connected and information updated and improves the scalability feature by using the Forwarding Group nodes (FG). CAMP can be considered as a proactive multicast routing protocol, which requires running an underlying unicast routing protocol and is using a periodical messages to keep routes up to date. This protocol distinguished by scalability issue because it constructs a mesh for each group and each mesh can have more than one core.
2.6 Summary

This chapter has described the different types of ad hoc routing protocols. With examples of each type as well as mentioning the main concepts, advantages and disadvantages of each.

The proposed work in this thesis uses the recursive unicast method to implement the multicast routing, so there is a need for underlying unicast routing protocol to be used. The selection of AODV to be the underlying routing protocol is done as this protocol gives a reliable and efficient tree based routing.

These mentioned protocols are considered as a state scheme protocols, because these protocols keep routing information inside the routers. The following chapter will discuss different type of wired and ad hoc protocols, explicit multicast protocols that were proposed to reduce the state routing information kept and move towards reducing the state information.
Chapter 3

RELATED WORK

This chapter explores the idea of explicit multicast in MANETs, by discussing the most important explicit routing algorithms for MANETS found in the literature to date. This chapter will start with the general idea about explicit multicast in MANETs then explains thoroughly these routing algorithms in terms of their strength and their drawbacks.

3.1 Introduction

In previous chapter, multicast protocols designed for MANETs were discussed. These protocols follow the concepts of the traditional multicast protocols; i.e distributed multicast routing state maintenance and shared group membership management. These protocols if applied for use with small groups within the network may become more complex and less efficient.

To solve this issue, recently there has been a shift towards the stateless multicast routing protocols for small group networks and for MANETs. Some of these protocols were developed to avoid any multicast routing state to be maintained by routers. The multicast data is encapsulated into a unicast packet and transmitted among end receivers. In other protocols a destination list is placed in the packet header and transmitted toward the destinations using the underlying unicast routing protocol.

The next sections will describe some protocols with details that fall into the scope of explicit multicast for static networks and for ad hoc networks used for small groups.
3.2 Explicit multicast protocols

3.2.1 Explicit multicast routing protocol (Xcast)

Xcast or Explicit Multicast [10] is a multicast protocol designed to support large number of small multicast groups over a network. This concept is achieved by explicitly encoding all destinations in the Xcast packet header, instead of using the conventional multicasting concept which use a multicast group address for the group members in the network. Xcast can be considered as a source-based protocol, as the source node is responsible for encoding the entire destination in the packet header, assuming that the underlying routing protocol will deliver this packet to all destinations.

In usual multicast protocols, packet carries the multicast group address as a multicast group identifier. In Xcast, the source node knows the destinations of the packets to be sent. As mentioned above, the source node will group the list of destination addresses according to their next hops on its packet header and will send the packet to the Xcast routers; as soon as the router receives this packet it parses the header and updates the packet header again by grouping the destinations according to their next hops. This procedure is repeated until the packet reaches the destinations.

Figure 3.1 illustrates the Xcast process in details. Node A wants to send data packet to the receivers (B,C,D) by the following sequence. First it encodes all the destinations according to the next hop (R1) into its packet header and forwards the packet to R1. By receiving the packet, R1 determines that R2 is the next hop so R1 modifies the header to be (R2:B,C,D) and forwards it to R3. Upon receiving the packet by R3, it determines that R4, R5 are the next hops, so the header will be modified to be (R4:B; R5:C,D). This process will continue and the header will be modified at each intermediate node, at R7 the header will be modified to be (R8:C;R9:D), until finally each destination receives the packet.
In general, the process that a router will do upon receiving an Xcast packet is as follows:

- Determine the next hop for each set of destinations by performing a route table lookup.
- Based on the next hop, the destinations will be divided into groups.
- Create a number of packets according to the number of groups from the previous step.
- Send each packet to its next hop with the modified packet header.
- If there is one destination for any of the next hops, this packet can be sent unicast by using the underlying routing protocol.

Compared to traditional multicast protocols, Xcast has the following advantages:

- Because it is a stateless protocol, routers need not keep any state information about the multicast group members.
- No need for multicast routing protocol, Xcast achieved the multicast concepts explicitly and the path is taken from the underlying unicast routing protocol.
- No single point of failure happens, as there is not core node required in Xcast.
On the other hand, Xcast has the following disadvantage:-

- Overhead. All the nodes in the neighbourhood that receive the Xcast packet need to process the header to check if it is one of the next hops included in the packet. This is a major overhead in terms of processing and delay.
- The huge increase in the packet header when the number of receivers increases. This increase will cause consumes in bandwidth and delay in delivering packets.

As discussed above, Xcast is suitable for small groups because there is no need to maintain information at intermediate nodes. Some modifications have been made to overcome Xcast drawbacks; Xcast+ is one of these.

### 3.2.2 Explicit Multicast Extension (Xcast+)

Xcast+ [61] is a multicast routing protocol, that is an extension of Explicit Multicast Xcast to achieve an efficient packet delivery of multicast packets. This is achieved by adding an Internet Group Multicast Protocol (IGMP) [36] join at the receiver’s side and sending a registration request message from the Designated Router (DR) received IGMP(S,G) join to the sender, and by encoding the (DRs) instead of encoding the receivers themselves as happened at Xcast.

In Xcast+, the receiver sends an IGMP(S,G) join, when the adjacent router -Designated Router (DR) - receives this join message it initiates a registration request message to the source. The source upon receiving the registration message replies by a registration reply message to the DRs to acknowledge the receiving of the registration request message. The source will keep track of the DRs addresses involved in the multicast session (S,G) in its cache table. When the source needs to sent Xcast+ packets, the sender and all other intermediate nodes follows the same process implemented in Xcast, except for encoding the addresses of Designated Routers DRs in the data packet instead of the addresses of the receivers. At the receiver’s side the Xcast+ packet is multicast to the receivers by DRs.

Figure 3.2 illustrates the Xcast+ process of delivering data packets. Receivers (A,B,C,D,E,F) send a IGMP(S,G) join to the adjacent router; (A,B) to M1, (C) to M2 and (D,E,F) to M3.
M1, M2 and M3 are called Designated Routers (DRs), and they will send a registration message to the sender S. The registration message consists of Source address S, Group address G and its own address. When sender (S) wants to send a data packet to the receivers, the sender encodes the DRs known to the sender (M1, M2 and M3) into the Xcast+ data packet. The process in the intermediate nodes is the same as in Xcast discussed before. When the packet is received by the DRs, they send the data packets to the receivers by multicasting.

![Figure 3.2 Xcast+ data packet delivery](image)

As discussed before, Xcast+ develops a few extensions to Xcast, and encodes the DRs in the data packet instead of receivers, and then it uses Xcast to Multicast (X2M) at the DRs. These extensions bring some benefits:

- Xcast+ is suitable for increasing the number of receivers in the subnet. This happens because the data packet encodes the DRs instead of receivers, and by default the number of DRs are less than or equal to receivers.
- Receivers at Xcast+ do not need any additional control messages to join the session. The receiver is capable of running IGMP, which means that it is capable of joining the Xcast+ session.
On the other hand, this protocol works under an assumption that receivers are working in groups, an idea that is not correct in real Internet environment. Xcast+ will work as Xcast because each DR will carry small number of receivers and the Xcast message will carry huge number of DR(s) which will decrease the scalability issue in this protocol. This issue is the main disadvantage that may be considered in this protocol.

3.2.3 A REcursive UNicast TreE (REUNITE) Protocol

REUNITE [65] is a multicast routing protocol which uses recursive unicast tree idea to implement the multicast service. REUNITE uses two types of messages, Join message and Tree message, in addition of implementing two tables in the routers: - Multicast Forwarding Table (MFT) and Multicast Control Table (MCT). This protocol was developed to improve the scalability over networks by reducing the amount of information stored in the multicast group members.

REUNITE protocol implements multicast service by using a recursive unicast scheme. Each REUNITE group builds a tree rooted by a designated node called a root. Every branching node (node with two or more different next hops) maintains a list of receivers in its MFT, where each receiver is maintained by one branching node only. When the root needs to send (multicast) packet it sends a copy to each receiver in its MFT list. When this packet arrives at branching node the branching node does the same by sending a copy of this packet to each receiver in its MFT list. This process is repeated until each receiver gets the packet.

As mentioned above, each router at REUNITE maintains a Multicast Forwarding Table (MFT) which contains entries for receivers where data delivery packet should branch at this router. Another table maintained by this protocol is Multicast Control Table (MCT) which contains an entry for receivers that the tree does not branch in this router. This protocol uses two types of control messages: JOIN message, which is sent (unicastlly) from the receiver periodically towards the root, and TREE message, which is sent (multicast) by the root along the multicast delivery tree. The TREE message is used to create and refresh the entries at MCT and the group entries at MFT, while the JOIN message is used to create and refresh the
receivers in the MFT, for more detail about the joining/leaving process related to this protocol, these details can be found in [65].

In [65] some benefits are mentioned of this protocol as follows:

- **Enhanced scalability by reducing forwarding state.** Only branching tree nodes keep information about the state of the group which allow groups to be bigger without high storage space to keep state information about.

- **Load distribution.** At conventional multicast protocols, if a receiver is unable to join the group for any reason, such as overload at the routers, the tree will become partitioned. In REUNITE if this case happens then the router just needs to ignore the JOIN message and lets an upstream router process this message and share the load. This is because at REUNITE not all routers should keep information about the multicast group members.

On the other hand, REUNITE has some drawbacks, such as:

- This protocol may face the scalability problem if the network is large, a problem that is created because in this protocol each node whether it is a branching node or not has to maintain a table to carry information about the network.

- The Join and Tree messages are sent periodically, which means an extra overhead is happened.

- The protocol decides that a node leaves the group as soon as it does not receive a Join message for a period of time. This assumption is not correct because there are other reasons that force the receiver to stop sending Join messages other than the receiver want to leave. This will cause loss of data for that receiver until it reconnected to the group again.

### 3.2.4 Hop By Hop Multicasting routing protocol (HBH)

HBH [22] is a multicast tree management routing protocol which implements a multicast distribution through a recursive unicast tree. It uses IP class D address for multicast group addressing.
HBH tree construction has the ability to solve the asymmetric problem found in multicast routing protocols. HBH uses two tables; MCT and MFT, these two table are the same tables used at REUNITE [65] protocol with the difference that each table entry at HBH contains the address of the next branching node instead of the address of the receiver itself. The data packet received by the branching node has a unicast destination address equals to the HBH branching node address. Each multicast group is identified by a unique pair of addresses (S,G), where S is the source node address and G is the group class-D IP address allocated by the source.

3.2.4.1 Tree construction in HBH

HBH uses three types of messages: Join Tree and Fusion. Join message is sent towards the source by a receiver which prefers to join the multicast group, this message is sent periodically in order to refresh the MFT entry in the branching node where this receiver is connected. Tree message is sent periodically by the source to finish and refresh the construction of multicast tree and to find possibly different path to the receiver which sent the join message. Fusion message is sent by a potential branching node to construct the distribution of the tree along with tree message.

Each router at HBH constructed tree maintains either MCT or MFT tables. Non-branching routers maintain a single MCT entry which contains the source and the receiver addresses which the later sends the join message through this router. Each branching router maintains a MFT table; which contains entries to the addresses of the receivers and/or the addresses of the next branching router side by side to the source address that is the first branching node in the network. The first join message send by the receiver goes directly to the source and does not intercepted by any intermediate router. Each entry on both tables is associated by two timers, (t1,t2). With expiry of t1 the table entry becomes stale, i.e used for data forwarding but no tree messages is sent to downstream nodes; and when t2 expires the entry is considered as destroyed. The basic ideas are: the join message is sent to the source and never intercepted by any intermediate nodes; the tree message is periodically send by the source; these are combined with fusion messages sent by potentials nodes to construct and refine the tree structure.
This protocol provides the best routes in asymmetric networks and enhanced tree stability in presence of group dynamics as mentioned at [22]. In addition, this protocol constructs the shortest path tree between the source and the receivers. On the other hand this protocol may share the same drawbacks that of REUNITE protocol.

3.2.5 Simple Explicit Multicast (SEM) Protocol

Simple Explicit Multicast (SEM) [11] protocol is one of the recent proposed protocols that adapt the explicit messages method in order to support multicast routing over networks. SEM [11] builds a multicast tree by using two types of messages: Branch and previous_branch. Moreover, SEM uses a periodical alive message to maintain the multicast tree. Joining and leaving of receivers is also done by Join and Leave messages sent by the receiver toward the source. The routing state for the multicast channel (S,G), where S is the source unicast address and G is the standard group multicast address, is only kept at branching routers table (TRM). Entries in TRM(S,G) are S, source address; G, multicast group address; Pb previous branching router address and the next branching router address.

3.2.5.1 Receivers and tree construction

A receiver wanting to join the multicast group (S,G) sends an IGMP join message addressed to this group. When receiving the join message by designated router (DR), this DR creates a join(S,G) message to the source. When the source receives this message this DR is added to a list kept in a Multicast Control Table (MCT), which contains all the DRs that have receiver(s) belonging to this multicast group (S,G). Each entry at MCT are S, Source node address, G, Multicast group address and the list of DRs.

Intermediate nodes do not need to maintain any state information for the multicast session. Thus, it is necessary for the DR routers to know the source address, the previous and the next DR.
SEM uses a periodical *alive* message between branching nodes to maintain the tree. The downstream DR sends *alive* message to the previous DR periodically. This message upon being received by the previous DR, it updates the contents of the TRM related to the sender. This message will keep the DRs connected and with up-to-date information.

In SEM, tree constructing uses *branch* and *previous_branch* messages. The source keeps track of the DRs that send join message on behalf of the receivers join messages, also the source encodes the list of DRs kept in its MCT table into the header of *branch* message. In addition, the source parses the header and portions the list into sub lists (Li) according to the next hop router and forwards a copy to each next hop router with the appropriate SEM header. Each intermediate router repeats the same process upon receiving the *branch* message, which has the role to discover the branching router in the multicast tree.

In intermediate nodes that are non-branching router, they just forward the *branch* message to the uniquely next hop without any changes. If the intermediate node becomes a branching router (two or more next hops to DRs), it checks whether there is an entry at its TRM corresponding to this multicast group. If yes, this entry is updated. Otherwise a new entry is created in the TRM table at this branching router. The TRM entry consists of the source address, the multicast group address, the previous branching router address (it is the source address at the first branching node receiving the *branch* message) and the list of next hop branching routers (initially empty). The branching router updates the branch message by replacing the value of the previous branching node with its own address. Then this branching router forwards the same message. The branching router also creates and sends the *previous_branch* message which contains its own address as a source address and a destination address as the address of the branching router sent the *branch* message. This message will inform the previous branching router about its next branching router. At the end of this process the multicast tree will be readily constructed with paths from source towards each of the DRs through the next branching routers. So the source can send data packets to the receivers within this group.
3.2.5.2 Receivers leave and tree maintenance

Because DRs run IGMP protocol, they automatically discover if any of the receivers connected or if all have gone. When DR discovers that there are no receivers connected directly to it, it creates an *alive* message and sends it towards the previous branching router. The branching router upon receiving this message eliminates this DR from its TRM table and creates a *leave* message and sends it directly to the source. When receiving the leave message, the source eliminates the corresponding state and sends a new branch message to all the lists (Li) to reconstruct the tree.

Moreover, if one of the branching routers or DRs breaks down, the upstream branching router will not receive an *alive* message from this broken node. Consequently, the upstream branching router will eliminate the state and sends a leave message to the source. The source at the same time will eliminate the corresponding state and re-build the tree by sending a new branch message. The source upon receiving the leave messages will not start directly the tree reconstruction, rather it will wait for a period of time and then start reconstruction in order to gather as much as possible of the *leave* messages and rebuild the tree again.

SEM has the advantage of reducing the state information in the multicast tree nodes. The drawback of this protocol in the case of node leaving is that it needs to reconstruct the whole multicast tree which will cause an extra routing overhead and will cause loss of time during the reconstruction.

3.3 Explicit Multicast in MANETs

In the previous section, a number of explicit multicast protocols for fixed networks were discussed. These protocols cannot be used for MANETs networks due to the nature of these kind of networks. Node mobility and wireless environment in combination can result in rapid and dynamic topology changes in MANETs.

Recently, explicit multicast ad hoc protocols were proposed to achieve the idea of stateless multicast for MANETs. In the next subsections, the most recently proposed and most popular protocols will be discussed.
3.3.1 Differential Destination Multicast (DDM) Protocol

DDM [20] is a source control based multicast routing protocol for MANETs. The source controls the multicast group to achieve the security and admission aspects. DDM encapsulates the destinations addresses into the data packet. At DDM when an intermediate node receives data packet including the DDM header, it just looks in the DDM header of the data packet to decide where to forward this data packet. When changes occur in the destination list or the underlining unicast routing protocol, an upstream node needs only to inform its downstream nodes (next hops) about the difference in the destination list for forwarding since the last successfully transmitted packet, hence the “Differential Destination Multicast” name comes from.

3.3.1.1 Membership Management

As mentioned above, this protocol is a source based multicast protocol. The source is acting as an admission controller. When a receiver (Joined node) needs to join the multicast group, it sends a JOIN message toward the source, this message includes the group ID to join in addition to the source ID and the receiver (Joining) node ID which already exist in the unicast IP header as source and destination fields.

When the JOIN message is received and accepted by the source, the source will add this receiver address to its Multicast List (ML). Then the source sends a unicast acknowledgment [48] message to the receiver.

The newly joining node, after sending the JOIN message, must wait for a period of time JOIN-WAITING-PERIOD. If it has not received any ACK or NACK (for secured session) by the end of this period it will retransmit the JOIN message again and reset the JOIN-WATING-PERIOD. The value of this period is reduced using exponential backed off algorithm after each every retransmitting until its value comes to zero. The joining node will stop sending JOIN message either if the value JOIN-WATING-PERIOD comes to zero or the joining receives an ACK message.
The ML kept at source should be refreshed to keep it updated, which is the responsibility of source. Once at every MEMBER-REFRESH-PERIOD, the source sets a POLL flag in the next data packet. A multicast session member receiving this data packet will resend a unicast JOIN message to the source to renew its membership. If after sending MAX-REFRESH-TIMEOUT polling data packets, and there is still no JOIN messages received from particular member, the source believes that these members are no more interested in being a member of this multicast group.

The member itself is responsible for taking the decision about leaving the session. The member which wants to leave the session sends a unicast LEAVE message to the source, upon receiving this message, the source will remove this member from its ML and no further data packets will be sent to this node.

### 3.3.1.2 Types and structure of packets

DDM uses two types of packets: control packets and multicast data packets. There are four control messages JOIN, LEAVE, ACK, and RSYNC. The first three types of control messages are used for members’ membership. The structure of these messages is: a field for message type, field for the group ID, source and destination ID fields. The last two fields are already in the unicast IP header for the packet. The RSYNC message is used to request upstream node to synchronize the stored list on both nodes.

Multicast data packets contain a DDM header and a payload. The first consists of summary section and DDM block, where the summary session contains flags fields, TTL field, and sender address. The DDM block contains the intended receiver, the type of DDM block, DDM sequence number and any other field required for some types. The DDM block types are: Empty (E), Refresh (R) and Difference (D) blocks.

The E block does not need any more fields, where the other two blocks have a list. The R block has a destination list to refresh, whilst the Difference (D) block is used to describe the difference in the receiver list. This D block is either incremental Di or decrement Dd. Both Di and Dd are included in the header when needed. Each type of D block has a list; Di has a
list (Li) which contains a list of nodes reaccepted also needs to forward the data to the nodes in this list in addition to whom it is already forwarded. A Dd block has a list (Ld) which informs the node received router to stop forwarding data packets to the nodes included in the list.

Each node for each active multicast session maintains a Forwarding Set (FS), where this set is a set of destinations that this node should forward the multicast data packet to. The FS at the source node is ML itself.

### 3.3.2 A Scalable Multicast protocol for MANETs

A Scalable Multicast protocol for MANETs (E2M) [38] is an explicit multicast protocol built on the top of Xcast. It overcomes some of the limitations found in Xcast schemes, which introduces a level of scalability. By dynamic selection of a node to be an Xcast Forwarder (XF), the XF selection is made according to the number of group members and the traffic load in the network. This protocol is designed to take into consideration the mobility feature for MANETs.

#### 3.3.2.1 Membership management

The membership management in this protocol is similar to the one used in [10] with an addition of two new messages, XF_JOIN and XF_ACK. The Designated Router (DR) which is eligible to become an XF sends an XF_JOIN towards the source with the list of destinations served by this XF. The source upon receiving this message sends back an XF_ACK to the XF router. Each member which wants to join the multicast group G sends a MEMBER_JOIN directly towards the source S. Each multicast session is uniquely identified by a pair of (S,G). The source upon receiving the MEMBER_JOIN message from the members, it adds this node to its Session Membership Table (SMT). The E2M protocol initially put all destinations in its SMT table in the Xcast header of the packet and forwards it. This initial packet forwarding is similar to the Xcast scheme for packet forwarding.

#### 3.3.2.2 XF Selection

An intermediate node may decide to become an XF. In [30] the authors mentioned that the traffic load and the number of downstream group members are considered the criteria for
selecting the XF. If the intermediate node X decided to become an XF, it generates a XF_JOIN message carrying the list of group members served by this node. During the XF_JOIN journey to the source, a Time To Live (TTL) field will be incremented when passing through the intermediate node. At the end TTL will carry the number of nodes between this XF node and the source. The source waits for a period of time before sending an acknowledgment for this XF_JOIN message. This delay period is to give time for any more XF_JOIN to arrive. If only one XF_JOIN is received by the source, it updates its eXplicit Forwarding Table (XFT) by inserting this node address and the corresponding list of nodes served by this XF node. This XF node should send periodical XF_JOIN message to the source making sure the stored information in the XFT table are up-to-date.

When the source node receives more than one XF_JOIN message, the TTL value will help the source to determine which XF is more suitable. If two XF_JOIN messages came from the same next hop then the source will select the XF with the larger TTL. In case the TTL values are equal, the source randomly selects one of these XFs to be an XF. Finally, the source will have the XFs and a list of destinations associated to each in its XFT table.

3.3.2.3 Node Mobility Management

Node mobility is an important feature in MANETs. Any proposed protocol for MANETs should be able to handle the node movement. In E2M, if the source comes to his knowledge that one of the XFs moved (e.g. route error packet generated by AODV) the source extracts the list of members served by this XF and insert these members in the Xcast extended header. In case a non-XF group member moves the periodic MEMBER_JOIN message will stop arriving to the corresponding XF, by that the designated XF will send an XF_JOIN with the difference in his state.

3.3.2.4 Data Structure

Data structures in E2M are implemented in XFs and in source node. For each session, the following tables are maintained by the source:-
• SMT (Session Membership Table): - This table maintains the membership for a given session.
• SFT (Session Forwarding Table):- This table maps the destinations to their next hop. Nodes with the same next hop are grouped together.
• XFT (eXplicit Forwarder Table):- Nodes working as a XFs and their downstream session members.

The source node maintains SMT for the members of each session and also for each session the XFT stores all XFs to their corresponding list of destinations addresses. The source maintains the SFT by grouping the group members to their respective next hop. This SFT is further divided into subsets according to their next hops. Each XF maintains both SFT and SMT tables. XF does not need to maintain a XFT table. Figure 3.3 shows an example of E2M packet delivery. Node A is the source node and nodes (B to G) are receivers, where N6 is an XF. The MFT source contains the XFs and receivers to its next hop, in our example, MFT at A contains N6 as an XF and B,C as receivers grouped to the next hop N1. Upon receiving the data packet, N1 forward the packet to its next hop without any changes, N2 will act similarly as the source did and sends two packets to its next hops. N6 will receive this packet and depending in its XFT, it generates a number of packets and forwards these packets to its next hops, the procedure continues until all receivers get the data packet.

![Figure 3.3 E2M packet delivery](image)

Figure 3.3 E2M packet delivery [7]
This protocol improves the scalability issue in multicast ad hoc networks. In [7] some drawbacks are mentioned for this protocol:

- All nodes on the path from root to XF should know the total number of receivers in its downstream network.
- Any node can play the role of XF without doing the real thing.
- This protocol did not effectively take into consideration the node mobility.

3.3.3 An Explicit Multicast protocol for MANETs

An Explicit Multicast protocol for MANETs (EM2NET) [7] has been proposed to support the scalability issue in MANETs. This protocol constructs a multicast tree consisting of set of Intercepting Nodes (IN), defined as a node that is part of the multicast group and serves two disjoint next hops in the group.

3.3.3.1 Messages and Data Structure

Each IN node should maintain a Multicast forwarding IN Table (MFIT). Each entry in this table consists of a unique pair of (S,G), where S is the session source and G is the group address, previous IN node address and a list of next INs.

EM2NET implements two types of messages: - control messages and data messages. Control messages are used to build the multicast tree for the group members and to keep this tree connected in case of node mobility. When a node need to join multicast group it sends JOIN message towards the source, which upon receiving this message reply by a BRANCH _ACK message to the JOIN message sender. During the BRANCH_ACK message the IN discovery will be started. If an intermediate node is a group member and received this message, it just needs to forward the message. Otherwise, if the node is an IN node then an entry in its MFIT table is created to the receiver node. The IN node send a BRANCH_UP message to the previous node to inform it about its new situation and continue forwarding the BRANCH_ACK message to the downstream nodes. At the end, the multicast tree will be constructed, consisting of a set of IN nodes with other group nodes. BRANCH_UP and
BRANCH_DOWN messages are sent periodically to keep the tree connected and solve any failure that could happen. These two messages will be discussed in the next section.

3.3.3.2 Node Mobility Management

A BRANCH messages (BRANCH_UP, BRANCH_DOWN) are sent periodically to make sure that the INs are still connected. If an X node did not receive a BRANCH_UP message in a certain period of time, the X relies that the downstream IN is no more available. Node X removes the entry for the downstream IN from its MFIT, and from now no packet will be forwarded downstream to that IN. In case an X node did not receive a BRANCH_DOWN message, this means that the upstream IN is no longer available. Node X will be a source for the subtree created because of this link failure. Node X starts a repair procedure as follows; firstly node X notifies the downstream node that there is link failure to the source. Secondly, it starts a local search for the new node to connect with a known TTL. As soon node X found new node to connect it sends a JOIN message to the source and sends a BRANCH_DOWN to the downstream nodes to inform it that the link is back. Figure 3.4 shows the packet delivery for EM2NET.

Figure 3.4 Packet Delivery at EM2NET [7].
EM2NET improved the scalability in ad hoc environment, but one can notice that the protocol did not mention the method of constructing the connections between the group members and there are no details about the joining/leaving procedure for receivers. These drawbacks may cause data loss or fragmentation in the multicast tree.

3.4 Discussion

As has been discussed in the previous sections of this chapter, Xcast, Xcast+, REUNITE, HBH and SEM are examples of explicit multicast protocols. These protocols are sharing in one thing by trying to improve the scalability issue in multicast networks, and this is done by trying to remove or reduce the state information stored in the nodes participating in the multicast tree in order to perform multicast routing.

Xcast can be considered as pure stateless multicast protocol, this protocol encodes all the receivers’ addresses in an Xcast header, and the intermediate nodes just forward the packet without any processing. There is no information to be stored in the intermediate nodes. This protocol works very well for large number of small multicast groups. However, when the groups become bigger, the efficiency of this protocol decreases because the Xcast header becomes bigger. Xcast+ protocol came as an improvement to Xcast, at this protocol the Xcast header encodes the Designated Routers instead of the receivers themselves; hence, this improves on the scalability issue. However, This protocol faces the problem of real application in Internet, where end users in general will not work in groups so the number of designated routers will be large and Xcast+ will not take advantage of its improvement and works as Xcast thus facing the same problem of scalability.

REUNITE and HBH are explicit protocols that are positioned between the state and stateless protocols. These protocols maintain tables in the group nodes to carry out information about the network. The main feature in these protocols is that they are using a recursive unicast method to construct the multicast tree. These protocols improve the scalability issue by allowing part of the group node to carry information about the group members’ receivers. On the other hand, these protocols cause high overhead because of the periodical messages implemented to keep information in the MCT and MFT tables’ updated.
To the best of our knowledge, SEM is the most recent and efficient explicit multicast protocol proposed. It adopted the idea of keeping the state information only in the branching node routers instead of keeping this information in all group members as in traditional multicast protocols. However, it still has a number of limitations that discussed before.

To overcome these limitations found in SEM, this thesis proposes new protocol called SReM, which has the same objectives as SEM with major enhancements. The basic idea behind SReM is to forward data between dynamically selected nodes called Branching Node Routers (BNRs). Its goal is to reduce the state information kept for the multicast group members and to reduce the routing overhead by providing a local join/leave and tree maintenance procedures using fixed size messages. Hence, it is achieving higher degree of scalability. The main difference between SReM and SEM is that in SReM there is no reconstruction of tree when group member leaves as in SEM.

The second part of the discussion is concerned with the explicit multicast protocols for MANETs. DDM, E2M and EM2NET are the most recent and efficient protocols proposed in this area of research. DDM is a source based explicit multicast protocol, which keeps the receivers addresses as list in the source and the list is kept up to date by periodical Join messages. These messages will increment or decrement the contents of the multicast list in the source node. In this protocol if the mobility is high then the changes in the list will be too much which will cause a huge overhead. Other problem may be faced by this protocol is the huge overhead in the source because the source is fully responsible for joining and leaving of members and forwarding data packets.

E2M employs Xcast forwarders (XFs) to take part of the job taken in the previous protocol by the source. The Xcast forwarder(s) keep information about a set of receivers and at the same time this XF is known to the source. This idea will reduce the number of control packets arrived to the source. This protocol did not give an efficient solution for mobility issue especially when the XF moves. The way proposed to solve the movement of XF is to give the source the responsibility to take XF job by again encoding the receivers that was
served by this XF in the source Xcast header. By using this method, in high mobility scenarios E2M will work like DDM and an extra overhead and delay will happen.

To the best of our knowledge, EM2NET is the most recently proposed explicit multicast protocol for MANETs. The basic idea of this protocol as claimed by authors is to dynamically forward data between intercepting nodes in the network. This protocol has a set of limitations discussed before.

In our research work, we proposed the Scalable Ad-hoc Recursive Multicast protocol (SARM) to overcome these limitations. Our proposed protocol has the same objectives as EM2NET with major developments. SARM differs from EM2NET in tree construction which done recursively and locally. Other improvement is the low latency of join/leave process.

3.5 Summary

This chapter discusses one type of multicast routing protocols proposed for wired and wireless ad hoc networks. The main feature of protocols related to that type is to reduce the routing information stored in the nodes participating in multicast tree. This feature is to improve the scalability in multicast routing protocols by swapping part of the state information to an explicit messages used during routing process.

In the proposed work in this thesis the scalability feature is considered for wired and ad-hoc multicast protocol. The proposed work started from where the others have reached and developed new scalable multicast routing protocols.

In the next chapter, a new scalable multicast routing protocol is proposed (SReM). This protocol proposed a new idea for improving the scalability feature of routing protocols. This chapter will give a full description and evaluation of the proposed protocol.
Chapter 4

SCALABLE RECURSIVE MULTICAST PROTOCOL

Scalable Recursive Multicast protocol (SReM) is a novel scalable multicast protocol for fixed network which will be discussed in details at this chapter. This discussion will include a detailed description of this protocol as well as the simulation environment, scenarios and metrics used to evaluate its performance for this protocol. Finally the output results will be presented and evaluated.

4.1 Introduction

Due to the scarce resources in the Internet, multicast provides an efficient solution to use these recourses fully and efficiently. Because of the explosive increasing in traffic over the Internet, multicasting became an important issue in routing protocols [66, 6]. Some issues are still open such as scalability [26], billing [25], address allocation [8] and security [44], where the scalability has drawn much attention for Internet application.

The main issue that causes the scalability problem is the increasing demand on the Multicast Forwarding Table (MFT) because of the increase in multicast group members or the increase in the number of multicast groups. So there are two main aspects that can be used to evaluate the scalability in multicast protocols: scalability with regard to the number of group members and to the number of multicast groups in the network. Recently, there are a number of proposed mechanisms to solve the scalability issue in multicast protocols, which can be categorized into tunnelling techniques, forwarding state reduction and explicit multicast. Our research falls in the explicit multicast mechanism; a detailed discussion for this mechanism was performed in the previous chapter.
Explicit Multicast (Xcast)[10] has been discussed in more details in the previous chapter, which was proposed to support a large number of small size multicast groups in the network. Xcast encodes the multicast receivers into a list and attaches it to the data packet header. Xcast+ is an extension for Xcast, by adding the IGMP-enabled joining receiver at the receiver side and sending the join request through a source specific join message to the sender. The source encodes the list of multicast routers instead of receivers address as in Xcast. Xcast+ supports a very large number of medium size multicast groups. However, both Xcast and Xcast+ still face the scalability problem, in case of large number of individual receivers which is found in the Internet.

The scalability problem facing Xcast and Xcast+ is considered by REUNITE [65]. It uses a recursive unicast protocol to implement the multicast scheme, and also it uses a source address and a port number to identify each multicast group rather that the class-D IP address. REUNITE maintains two tables, MFT and MCT. The branching node routers maintain the MFT table where the non-branching node routers implement the MCT table. REUNITE supports node balancing and graceful degradation such that the branching can be automatically migrated to other less loaded router if the designated router does not have enough resources to support additional multicast group members. Similar to REUNITE, HBH [22] uses unicast infrastructure to do packet forwarding, but uses EXPRESS’s [37] channel abstraction to identify a group. The main advantage of HBH is that it takes into consideration the asymmetric paths into the network. Both REUNITE and HBH suffer from massive control messages because the use of extra multicast control table MCT besides using the MFT table. Reducing the number of overhead messages found in the previous proposed protocols is considered by Simple Explicit Multicast (SEM) [11]. This protocol uses a similar mechanism used in Xcast+ [61] to construct the multicast tree. For delivering packets it uses a similar mechanism used at REUNITE [65]. In (SEM) [11], the source uses unicast encoding for multicast packets and sends them to its next hop branching node routers. Each branching node router acts as a source where packets travel from a branching node router to another. In SEM, the mechanism of using explicit messages is introduced to build a whole multicast tree in advance to deliver multicast packets among branching nodes. However,
SEM has a major problem in the join/leave latency because of reconstruction of tree in case of member leaving.

4.2 Overview of proposed protocol

The proposed Scalable Recursive Multicast protocol (SReM) uses a method that builds the multicast tree dynamically and gradually as group members’ join/leave.

The proposed protocol introduces the idea of using a dynamic branching node-based multicast tree (DBT) to deliver packets. In SReM, there are two types of signalling messages, i.e., join/leave signalling messages (JS/JL) and branching node messages (BNMs). Each receiver that wants to join/leave a multicast group (G) sends a join/leave message (JoinM/LeavM) to its local multicast router (LMR) using IGMPv.3 [36]. This LMR then sends a registration request message (R_qM) to the multicast source (S) on behalf of this receiver. In response to the receiving of the R_qM, the source will send a registration replying message (R_pM) to confirm the registration. In the initial stage of the multicast tree setup, the multicast source will be responsible of searching for the first branching node router (BNR) by using a pair of BNMs upon receiving these registration messages. Following that both the multicast source and BNRs will perform this function (searching for BNRs).

As a result, a dynamic branching node-based multicast tree (DBT) will then be created. Each of BNRs in DBT has a multicast forward table (MFT) with the multicast tree identity (MTI) of (S,G). MFT includes the address of its previous BNR as well as a list of its next branching nodes’ addresses. Using the DBT, SReM delivers multicast via BNRs by unicast.

As mentioned, SReM uses the mechanism of building the tree in response of join/leave of receivers. This mechanism is similar to SEM [11], but in SEM the tree is build in advance. SReM uses unicast mechanism to forward multicast data packets similar to REUNITE [65] and HBH [22] but using only one multicast table in the branching nodes (MFT) without using the multicast control table (MCT). SReM adapts a pair (S,G) to identify each group similar to EXPRESS [37], but keeping the multicast state about their next BNRs rather than the whole multicast group members.
The basic idea of DBT [5] was proposed without any actual implementation and simulation study. In our first part of research a modification was made to the previous proposed protocol and implemented in addition to comprehensive simulation study.

The remainder of this chapter is organized as follows; Sections 4.3 to 4.5 will explain SReM protocol and its details while at Section 4.6 the implementation and results will be discussed in Section 4.7.

4.2.1 Definitions

In this section, definitions of terminologies and messages used in the proposed protocol are explained. These messages and terminologies will be discussed in details during the explanation of the proposed protocol in this chapter, which are summarized in Table 4.1.

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMR</td>
<td>Local Multicast-enabled Router, a multicast group member connecting the receivers with multicast enabled feature.</td>
</tr>
<tr>
<td>IMR</td>
<td>Intermediate Multicast-enabled Router, a multicast enabled router connecting different multicast group members.</td>
</tr>
<tr>
<td>BNR</td>
<td>Branching Node Router, an IMR with the ability to serve as a branching node in the multicast tree.</td>
</tr>
<tr>
<td>MFT</td>
<td>Multicast Forwarding Table, a table maintained by each BNR to store information about the previous BNR and next BNRs/LMRs.</td>
</tr>
<tr>
<td>DBT</td>
<td>Dynamic Branching node-based Tree, a kind of tree-based method to form a network dynamically.</td>
</tr>
</tbody>
</table>

The proposed protocol uses a set of messages for constructing and maintaining the multicast tree. Table 4.2 summarizes these messages used in the proposed protocol.
### Table 4.2 Messages used in SReM

<table>
<thead>
<tr>
<th>Message</th>
<th>Function</th>
<th>Originator</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>JoinM</td>
<td>IGMP v3 message</td>
<td>Multicast receiver wants to join the multicast group</td>
<td>A multicast-enabled Local Router (LMR).</td>
</tr>
<tr>
<td>LeaveM</td>
<td>IGMP v3 message</td>
<td>Multicast member wants to leave the multicast group</td>
<td>A multicast-enabled Local Router (LMR).</td>
</tr>
<tr>
<td>RqMs(type)</td>
<td>Request message on behalf of new receiver joining/leaving a multicast group</td>
<td>Local Multicast Router (LMR) on behalf of in response to JoinM/LeaveM messages.</td>
<td>The multicast group source node.</td>
</tr>
<tr>
<td>RpMs(type)</td>
<td>Reply message on behalf of RqM message.</td>
<td>The multicast group source node.</td>
<td>The designated LMR organizing the RqM for new receiver joining the multicast group.</td>
</tr>
<tr>
<td>eBNM</td>
<td>Branching node enquiry message</td>
<td>Any Branching Node Receiver (BNR) in case of new branching node creation.</td>
<td>Downstream Intermediate Multicast Router (IMR)</td>
</tr>
<tr>
<td>rBNM</td>
<td>Branching node reply message</td>
<td>Any IMR eligible to be a new BNR</td>
<td>The upstream BNR that sent the eBNM</td>
</tr>
</tbody>
</table>

### 4.3 Membership Management

#### 4.3.1 Dynamic Branching node-based Tree (DBT)

In SReM, the interaction between branching nodes due to multicast group member joins and/or leaves, leads the setup of the multicast tree dynamically and gradually, rather than in advance. Therefore, the multicast tree is referred to as a dynamic branching node-based tree (DBT). The two types of branching node messages used to set up the DBT are enquiring BNM (eBNM) and replying BNM (rBNM). These two messages carry information like previous branching node address, next branching node address and new Local Multicast-enabled Router (LMR).

The key point in setting up the DBT is that each Intermediate Multicast Router (IMR) is aware whether it is a BNR or not. So, IMRs have three functions:

- Know its state as a BNR or not.
- Investigate BNRs in the route to the destination.
- Inform the upstream BNR of its state.

More details on DBT construction and the format of messages will be discussed later.

4.3.2 Messages and tables structure at SReM

In SReM a set of messages and one table are used to build and maintain the multicast tree. These messages are categorized into three main types;

- The joining and leaving messages send by new or existing members to join or leave.
- The messages sent by LMRs on behalf of joining or leaving members
- The messages used for updating or creating BNRs in the multicast tree.

The only table created and maintained is the Multicast Forwarding table (MFT) at the branching nodes. In the next section a detailed description and explanation of these messages and the MFT table.

Messages structure and signaling

As mentioned in the previous section, messages used in SReM can be divided into three categories: - joining/leaving messages between LMRs and receivers, joining/leaving messages on behalf of the first type from LMRs to the source and branching node messages among BNRs include the source.

In SReM, the join/leave message is the first message for building the multicast tree, which initiates the procedure, when a receiver wants to join the multicast group and become a member of this group it sends a JoinM message to its LMR which is the local or closest multicast enabled router. On the other hand, a group member wants to leave the multicast group needs to send a LeavM message to the LMR. These messages are sent using IGMPv3 protocol. The format of these messages is discussed in [36] and explained in Figure 4.1.
Where

- **Type**: The message type (0x11 JoinM, 0x17 LeaveM)
- **Max. Resp. Code**: specifies the maximum time allowed before sending a responding message, (Used only for JoinM message).
- **Checksum**: The checksum for messages errors.
- **Group Address**: The group address the receiver is joining or leaving for JoinM or LeaveM messages.
- **Resv**: Reserved. Set to 0 on transmission and ignored when reception.
- **Number of Sources (N)**: Number of source addresses in this message.
- **Source Address [1..N]**: The IP address(s) the message is sending to.

The IGMPv3 messages are restricted for the communication between the receivers and the LMRs. To transmit the information included in these two previous messages to the source, new pair of messages are created at LMRs; Registration request Message (RqM) and Registration reply Message (RpM).

These messages may be updated at BNRs. RqM messages are created on behalf of joining a new group member in the multicast group and sent to the source. The source upon receiving this message will reply by RpM to the message sender. A detailed explanation about the role of these messages in joining and leaving process will be discussed later in this chapter.
In SReM, the formats of RqM and RpM messages are similar to the format of IGMP messages with some changes as follows:-

- The Max Resp. Time field in IGMP message is changed to a one bit flag field used by SReM registration messages, called S flag.
- An address field is added to the payload of the message. This field carries the receiver or the existing BNR address.

RqM and RpM are encapsulated in IP datagram, where the source is one of the LMRs and the destination is always the multicast source. Figure 4.2 shows the format for RqM and RpM messages.

```
<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>15</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>S</td>
<td>reserved</td>
<td>Checksum</td>
</tr>
<tr>
<td>Source-specific class D address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver's address / an existing BNR's address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Figure 4.2  RqM / RpM message format**

Where

- **Type**: This field for RqM or RpM message type :
  - 11: specifies a member joining packet (RqM). This packet sent by the LMR.
  - 00: specifies a membership confirmation packet (RpM). This is sent by the source or BNR.
  - 01: specifies a member leaving the multicast group (RqM). This packet is sent by the LMR.
  - 10: specifies the update of branching nodes (RqM). This is sent by the BNRs.

- **S bit**: This one bit field is set to 1 when the packet type is (11). For other types of it is unset to 0. This field is sent with (11) packet with value 1 to indicate that any BNR or IMR in the route can process this message. As soon as this packet processed by one of BNRs or IMRs it is unset to the value of 0 which means that no more processing is required from any of BNRs or IMRs.

- **Reserved bits**: for any future use.
- **Checksum**: This field contains 16-bit checksum.
- **Source-Specific class D Address**: this field contains a source-specific multicast group address.
- **Receiver’s address / an existing BNR’s address**: This field specifies an address of a multicast member. For the first three types of packets this field contains the receiver address joining or leaving the multicast group. For the fourth type of packets this field contains the address of the existing BNR.

In SReM the (“10”) message type is important in maintaining and updating the multicast tree. With this type (i.e. “10”), the field of receiver’s address in the message (RqM) should be the address of new branching node (not like other three types), which is stored in the existing BNR’s MFT. This message is triggered if a RqM message with the type=(“10”) (i.e. a leaving message) is arrived at an existing BNR which has only one forwarding branch in its MFT, sent by this existing BNR (no longer being BNR now), and will be sent to and terminated at its upstream BNR.

In the rest of the chapter the previous messages will be specified as RqM_s(Type) and RpM_s(Type), where (s) is the state bit and can be either 0 or 1, and (Type) describes the message type discussed before. For example, RqM_1(11) is a non interrupt (s =1) request message sent by one LMR towards the source on behalf of a new receiver join.

The third type of messages is the branching node messages, which consists of two branching node messages, enquiry Branching node Message (eBNM) and replying Branching Node Message (rBNM). The eBNM format shown in Figure 4.3 is as follows:

((src= the IP_branching node address; group address = (S,G); dest = IP_new LMR router, IP_next branching node; previous branching node)).
The rBNM message format is similar to the eBNM but simpler. rBNM format shown in Figure 4.4 is as follows

([src= the IP_new branching node address; group address = (S,G); dest = previous branching node]).

In SReM, the procedure of sending BNM messages is performed by one of the BNRs to search for new BNR. This procedure starts by a new receiver joining the multicast group. So BNRs are responsible for building the multicast tree which happens gradually and locally. The only case that the source will participate in building the multicast tree is when a new receiver is joining the multicast tree through an IMR located between the source and its next BNRs. This kind of building the multicast tree is another remarkable advantage for SReM over other related protocols like SEM [9], where the multicast tree is built in advance and needs to be rebuilt whenever new joining or leaving process happens.
The Multicast Forwarding Table (MFT) is created and updated in BNRs (in this sense, the source is one of the BNRs) to carry an entries to the other BNRs and LMRs connected to it. It is worth to notice that only branching nodes maintain such table and other multicast tree members do not have to maintain any tables. This is also an advantage for SReM over related protocols like REUNITE and HBH where the non-branching nodes should also maintain MCT tables. The structure of MFT entry, shown at Figure 4.5, is as follows: (The session address (S,G), IP_previous branching node, IP_next Branching node and/or IP_LMRs). This table is created and updated in response to branching messages. The issue of creating and updating the MFT will be discussed next section.

<table>
<thead>
<tr>
<th></th>
<th>Session Address (S,G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP (previous BNR)</td>
<td></td>
</tr>
<tr>
<td>IP (next BNR or LMR[1])</td>
<td>.</td>
</tr>
<tr>
<td>IP (next BNR or LMR[2])</td>
<td>.</td>
</tr>
<tr>
<td>IP (next BNR or LMR[N])</td>
<td>.</td>
</tr>
</tbody>
</table>

Figure 4.5 MFT entry format

4.4 Joining / leaving and BNR Selection Procedure

4.4.1 Service Discovery

Although the service discovery methods are out the scope of this thesis, it is worth mentioning a brief discussion of them. Service discovery protocols allow devices to discover automatically network services thus making the task of network administration and configuration easy. In fixed and wired networks service discovery protocols simplify the interaction among users, devices and services [73].

The service discovery protocols for wired networks have been divided into three categories. Directory based, set of nodes that store information about providers. Clients then query these
nodes to discover providers. Some are directory-less like UPnP [2] that has a peer-to-peer (P2P) architecture and the service information is stored on each device. The third category is hybrid that can work in both modes, that is with or without a directory depending on the situation. Examples of these service provide protocols are Jini [39], Universal Plug and Play (UPnP) [2] and Salutation [19].

In this thesis, it is assumed that the receiver has discovered the service by any of the above methods and is ready to initiate the request for joining. As part of that, it also assumed that the receiver has obtained the necessary information required to join the multicast group by sending the appropriate join message using IGMPv3.

As such, the scope of SReM stands as it receives the join message from the receiver and deals with routing and joining/leaving of receivers.

In SReM, joining/leaving and BNR selection is done by using the above mentioned signalling messages i.e JoinM, LeavM, R_qM, R_pM, eBNM and rBNM. These messages work together and individually to build the multicast tree and perform the process of multicasting. The BNR selection is part of joining and leaving process so the description of BNR selection will be described during joining and leaving processes. In the next subsection will describe the role of these messages.

4.4.2 Joining process in SReM

The joining process in SReM is initiated at the receivers, by a procedure that differs depending on the structure of the existing multicast tree and from where the new receiver is joining the session. In conclusion, there are three cases for joining process in SReM as follows:-

- Case 1: the new receiver is joining the multicast group through an LMR with already joined receiver(s).
- Case 2: the new receiver is the first receiver joining through an LMR and this LMR is directly connected to a BNR.
- **Case 3:** The new receiver is the first receiver joining through an LMR and this LMR is not directly connected to a BNR.

In all these mentioned cases which will be discussed in detail in this section, it is assumed that S is the multicast group source node, N1 to N3 are IMRs or BNRs, M1 to M5 are LMRs and (a, b) and (x) are receivers already joined or need to join the multicast tree respectively.

- **Case 1:**

In this case, there are previous members joined to the LMR where the new receiver member will join. Figure 4.6 shows this case. Node x (the new receiver member) sends a JoinM to its designated LMR M3, upon receiving this message, M3 will initiate a RqM0(11). Because the value of S bit in the message is zero, this message will go directly to the source and it will not be accessed by any of the IMRs or BNRs in the path to the source. Upon receiving this message, the source S initiates and sends a RqM1(00) toward the new receiver(x). When this message received by the BNR serving this new receiver, it unset the S bit and forward the message to the LMR connecting this new receiver. It worth to notice that in this case there is no changes in the multicast tree construction.

![Figure 4.6 Joining process in SReM (Case 1)](image)

- **Case 2:**

In this case, the new receiver (x) is joining a multicast group through an existing BNR, i.e the new receiver’s LMR is attached to the first BNR directly. Figure 4.7 shows this case. The new receiver node (x) will send a JoinM to the designated LMR (M3). Upon receiving this message, M3 initiates and sends a RqM1(11) toward the source. The first IMR in the path to
the source is N2 where it is already a BNR, and will update its MFT table entry to add x’LMR (M3), and unset the S bit in the request message. The request message RQM0(11) sent to the source to complete the joining process. As mentioned before this message will arrive in short time to the source because the S bit is set to 0 on the first BNR (N2). The source upon receiving this registration message sends back a confirmation message RpM1(00) towards the first BNR(N2) on behalf of x’LMR(M3) request message. At the end of this procedure the new receiver (x) will be considered as a member in the multicast group.

Again, it is worth to notice that there are no changes in the multicast tree construction because of this new joining receiver.

*Case 3:*

In this case, the new receiver (x) is the first receiver joining LMR and there is intermediate IMR(s) in the path to the first BNR towards the source. Figure 4.8 shows this case, where the new receiver (x) that wants to join the multicast group sends a JoinM message to the designated LMR (M3). Upon receiving this message, M3 will create and send a RQM1(11) towards the source. The IMRs along the route to the source just forward this message without any changes. Upon receiving this message, N2 as the first BNR along the route to the source updates its MFT entry by adding the x’LMR. Moreover, N1 will update the request message

---

**Figure 4.7 Joining process in SReM (Case 2)**

<table>
<thead>
<tr>
<th>Nodes</th>
<th>MFTs Before x join</th>
<th>MFTs After x join</th>
</tr>
</thead>
<tbody>
<tr>
<td>S (Source)</td>
<td>MTI</td>
<td>unchanged</td>
</tr>
<tr>
<td>N1</td>
<td>MTI &amp; IP_M1 &amp; IP_M0</td>
<td>unchanged</td>
</tr>
<tr>
<td>N2</td>
<td>MTI &amp; IP_M0</td>
<td>MTI &amp; IP_M0 &amp; IP_M1</td>
</tr>
<tr>
<td>N3</td>
<td>unchanged</td>
<td></td>
</tr>
</tbody>
</table>

---

Join message

---

RQM or RpM
by changing the value of $S$ bit to zero and forward the updated message ($R_qM_0(11)$) to the source.

The source upon receiving this message will create and send a reply message ($R_pM_1(00)$) towards the $x'LMR$ to confirm that the joining of the new receiver ($x$) is accepted.

Figure 4.8 Joining process in SReM (Case 3)

In this case, $N_1$ will discover that the new receiver ($x$) is connecting through an LMR where this LMR is not in $N_1$ MFT entry. In this case, changes in the multicast tree should happen; a new branching node should be added to the multicast tree. In other words, tree maintenance should take place because of this new receiver joining case. Because the tree maintenance process is an important issue in SReM, it will be discussed later in a separate section in this chapter.

4.4.3 Leaving process in SReM

The receiver that wants to leave the multicast group starts by sending a leaveM messages to the LMR connecting this receiver. Different cases of leaving process may happen depending on the tree structure and whether or not the leaving member is the last member attached to the LMR. These expected cases are:-

- Case 1: the leaving member is not the last member connected to the LMR.
- Case 2: the leaving member is the last member connected to the designated LMR and the BNR connecting this LMR will stay a BNR after member leaving the multicast group.

- Case 3: the leaving member is the last member connected to the designated LMR and the BNR connecting this LMR will not be a BNR any more.

In this section these cases will be discussed in detail assuming that S is the multicast group source node, N1 to N3 are IMRs or BNRs, M1 to M5 are LMRs and (a, b) and (x) are receivers already joined or need to leave the multicast tree respectively.

- **Case 1:**
In this case, Figure 4.9, the leaving member (x) sends a leaveM message to its designated LMR (M3), M3 will know that the leaving member (x) is not the last member connected to it. This is done by checking the entries in the Multicast Destination Table (MDT) maintained each LMR.

![Diagram of Case 1](Figure 4.9 Leaving process in SReM (Case 1))

The designated LMR (M3) will initiate a RqM0(01) message and forwards it directly to the source. Because the S bit flag is 0, the intermediate IMRs or BNRs will not process this message and it will take a short period of time to arrive the source. It worth noting that there is no changing in the multicast tree because member (x) leaving the group.
Case 2:
In this case, where receiver (x) is the last group member attached to the designated LMR, and the first BNR connecting the designated LMR will stay a BNR after member (x) leaving the multicast group. Figure 4.10 shows this case, where the receiver (x) will send a leaveM message to the designated LMR (M3), upon receiving this message, M3 will create and forward RqM1(01) message toward the source. The first BNR (N2) upon receiving this message from (M3) it will check its MFT deciding whether it will stay a BNR or not. In this case N2 will stay as a BNR, and will update the message by changing the S bit to zero and forward the updated RqM0(01) message the source. At the same time N2 will update its MFT by removing the entry for the M3. The RqM0(01) sent by N2 will go directly to the source and no process will take place on this message in the intermediate IMRs or BNRs. Again, in this case the multicast tree structure will not be changed.

Figure 4.10 Leaving process in SReM (Case 2)

Case 3:
In this case, shown in Figure 4.11, the BNR(N2) will not stay a BNR after receiver x leaves the multicast group session. The existing LMR (M3) will send RqM1(01) message to the first BNR(N2). Upon receiving this message N2 will check its MFT table, it realizes that M3 is the only branch and when receiver (x) leaves, this BNR will not be a BNR any more. N2 will send the RqM1(10) message towards the source, and will remove the LMR entry from its
MFT. The previous (upstream) BNR (N1) upon receiving the R_qM_1(10) will update its MFT by removing the first BNR (N2) from its MFT and will check out that instead of (N2), N3 will be the new downstream BNR. Meanwhile (N1) will send R_qM_0(10) message towards the source to complete the de-registration process. In leaving process, only in this case, changes occurred to the DBT.

**Figure 4.11 Leaving process in SReM (Case 3)**

In the previous section and subsections a discussion of building and upgrading the DBT for a multicast group session was performed. Next section will discuss about the delivering of data in SReM.

### 4.4.4 Tree maintenance

As discussed in the pervious sections, some cases in joining and leaving of receivers, changes will occur in the DBT. The procedure of applying these changes to reconstruct the DBT and continue the data flow is called tree maintenance.

In case 3 of joining a new receiver discussed before, changes will occur in the DBT. The changes include a new branching node added to the DBT. The process of finding the new BNR and updating the previous BNRs because of the new receiver joining is the scope of discussion in this section.
Figure 4.12 shows the case when a new receiver (x) is the first receiver wants to join the multicast tree through M3. In this case tree maintenance should occur, as mentioned previously in this chapter.

![Diagram showing tree maintenance in SReM]

N1 is the first BNR receiving the join message; N1 will realize that this joining message is coming from new branch that is not found in its MFT table. Upon receiving the join message, N1 initiates an eBNM message with fields values shown in Figure 4.13 and forwards the message downstream towards the x’LMR.

![Table showing MFT values]

Upon receiving this message, the IMR (N2) will discover that the list of destinations in this message is connected to this IMR from different next hops. At this stage, N2 realizes that it is a new branching node for the tree, and then N2 creates an rBNM message with field’s values shown in Figure 4.14 and sends it upstream towards the previous BNR. This new BNR (N2) will create an MFT table with one entry for M3 and N3 as branching for it as shown in Figure 4.5. Meanwhile, N2 will create a rBNM message to its previous BNR (N1), N1 address extracted from eBNM. Upon receiving the rBNM message, N1 updates the entry in its MFT table by removing the N3 address entry and insert N2 address as a next BNR.
4.5 Delivery of Packets

4.5.1 SReM header and packet delivery

Data packets delivery happens by using unicast packet delivery from the source to the next BNR. These packets travel between BNRs until they reach the LMRs, from there they are forwarded using the standard multicast. The data packet format used in SReM is similar to the packet format used for unicast data packet with some differences. At SReM data packet, the IP destination address is always the next BNR hop. Also the pair of (S,G) that distinguish each multicast session is added to the header. Figure 4.15 shows the data packet format with SReM header.

To multicast a data packets, a SReM header is added to each packet by BNRs (including the source itself). Each BNR encodes the address of one next BNR in data packet, which leads to a fixed size of packets towards the next BNR.

At the start of multicasting, the source will send a copy of data packet with respective SReM headers to all BNRs in its MFT. Upon receiving this packet, each BNR needs to properly process the SReM header and performs the following steps for processing the data packet:

- Checks its MFT and replicates the packet so that there is only one copy of the packet for each of its next BNRs.
- Modifies the SReM header in each data packet copy by changing the destination IP address to the IP address of the next BNR.
- Sends the modified copies of the data packets to their destinations unicastly.
For non-branching nodes, packets are forwarded just like normal unicast. Finally, when the data packets arrive to the (LMRs), these packets will be delivered to the receivers by using the standard multicast method.

4.5.2 Multicasting in confirmed mode

In SReM, the access control of multicast packets needs to be authenticated via the registration process before being sent to the receivers. For convenience, in this thesis it’s referred to this multicast as a Multicast in Confirmed mode (MCM). The first case of this multicasting mode is shown in Figure 4.16 shows the data packet delivery in MCM in case a new BNR created and explained as follows

Figure 4.16 Data packet delivery in MCM in case a new BNR created

In this case a new BNR is created because of new member joining, discussed in Section 4.4.2 case 3, therefore in SReM, the multicast source is always in charge of the registration process of members of a multicast group, but the delivery of multicast packets is fulfilled through BNRs, exactly a ‘local’ BNR for a new member. Although the part of registration process (initiated from BNRs) is always towards and through the multicast source, this process is fast and expected to be finished almost at the time when BNRs are in place to deliver multicast packets. This leads to efficient and less delay in the delivery of multicast packets. The other
case shown in Figure 4.17 is similar to the first case but there is not any creation for new BNR, the data packet delivery is similar to case1.

Therefore, SReM is effective and of less latency in delivering of multicast packets.

![Diagram](image)

**Figure 4.17 Data packet delivery in MCM in case no new BNR created**

### 4.5.3 Multicasting in unconfirmed mode

In some circumstance, the multicast source doesn’t need to be aware of which are receiving the data packets sent, but it needs to know the receivers in a statistical point of view. In this case, BNRs and LMRs do not need to wait for registration reply messages (RpMs) from the source and can immediately multicast packets to receivers as soon as they are in place to deliver multicast packets. Clearly, this process of delivery of multicast packets is faster than that in MCM, referred as multicasting in unconfirmed mode (MUCM) in this thesis. This process of delivering of multicast packets has two cases, depending on the presence of new BNR formed, which are similar to the previous multicasting mode cases and shown in Figure 4.18 shows the data packet delivery in MUCM in case a new BNR created and Figure 4.19 shows the data packet delivery in MUCM in case no new BNR created.
To support MUCM, branching node messages (BNMs) should contain a field to show the type of multicast source, say, confirmed or unconfirmed. If this field marked to be confirmed, it means multicast packets can not be forwarded to receivers by BNRs or LMRs until RpMs being received (this is the case of MCM), otherwise, it is the case of MUCM.
This mode is called unconfirmed multicast data packet forwarding, which means a receiver can receive its packets without any permission from the source. Meanwhile, another entry in BNRs and LMRs needed to be created to keep the information of multicast delivery mode. BNRs and LMRs, which keep this information, can decide whether or not they send and forward multicast packets immediately or until receiving an RqM. Obviously, in MUCM, no RpMs messages are needed to be generated.

As it can be noticed, the joining process in MUCM is similar to that in MCM, except for that BNRs or LMRs starts to send multicast packets for a receiver as soon as receiving an RqM. On the other hand, the leaving process in MUCM is totally the same as that in MCM. The BNR or LMR for a receiver will stop forwarding multicast packets immediately when they receive RqMs.

4.6 Performance Evaluation

Performance evolution is an idea used to measure and evaluate the performance of routing protocols. In this thesis, to evaluate the proposed protocol and compare it to some other protocols, a detailed performance evaluation is applied. This performance evaluation divided into two parts:

- **Cost analysis**, in this evaluation part, different cost measurements are investigated and evaluated. These measurements are also applied on other protocols in the same area of our proposed protocol. At the end, a discussion for the results is obtained.

- **Simulation analysis**, at this evaluation part the proposed protocol is implemented using a well-known simulator. The simulation run on different scenarios and evaluating different metrics. The results obtained from this part of evaluation are also compared with other protocols in the same area of SReM. At the end of this part a discussion about the results is provided.
4.6.1 Comparison and Cost analysis

A. Comparison between SReM, Xcast, Xcast+ and SEM

Xcast encodes the list of the addresses of all receivers in each packet, while Xcast+ encodes the list of designated routers in each packet. However, both SReM and SEM use this mechanism only in the branching message. In Xcast and Xcast+, the packet will follow the unicast path between the source and the destination, but in SEM and SReM the packet will follow a unicast path among branching nodes. Furthermore, there are two major differences between SEM and SReM as follows.

In SEM, all of the destinations of a multicast session are encoded in branching messages to build a multicast tree in advance and the source or sender is always responsible of the set-up and maintenance of the whole multicast tree, and the multicast tree needs to be rebuilt whenever the joining or leaving of one of destinations is happened. This leads to two disadvantages.

First, SEM has the scalability problem, like the Xcast and Xcast+. This is due to the fact that as the size of a multicast group increases, both the size of delivering packets in Xcast and Xcast+ and the size of BNMs in SEM will increase correspondingly.

Secondly, SEM has the problem of a big join/leave latency, which is occurred by the set-up of the whole multicast tree in advance or rebuilt the whole multicast tree whenever a new member is joining or an old member is leaving.

B. Comparison between SReM, HBH and REUNITE

The main similarity between our proposed SReM, HBH and REUNITE is that all of them use unicast infrastructure to perform packet forwarding, and only the branching node routers (BNRs) are required to maintain a multicast forward tables (MFTs). This feature of our proposed SReM, like HBH and REUNITE, enable it to be deployed under existing Internet which is based on unicast infrastructure.

The main difference between SReM and HBH and REUNITE is that SReM uses the so-called DBT mechanism to maintain and update the multicast tree, while HBH and REUNITE
use the multicast tree control plane based on Multicast Control Table(s) (MCTs) in no-branching node routers (non-BNRs). Due to this fact, REUNITE and HBH suffer from the massive control messages traffic due to the use of this extra multicast control table (MCT) at all of non-branching nodes in addition to MFT in BNRs. This is because the number of BNRs usually is far less than the number of non-BNRs, given a multicast group, only less part of multicast routers takes part in the multicast session in SReM, while all multicast routers are involved in the multicast session in HBH or REUNITE. Therefore, from this scalability point of view, the advantage of our proposed scheme SReM is remarkable compared with HBH and REUNITE.

An additional disadvantage, REUNITE introduces dynamic behaviours such as tree restructuring, race condition of joins and duplicates packets during tree restructuring. However, these problem will not be found in SReM due to the use of DBT multicast mechanism.

Another significant difference between our proposed SReM, HBH and REUNITE is that in SReM the join/leave message is sent only when a new member of multicast group join or an existing member leave, while both REUNITE and HBH needs periodically sending JOIN and TREE messages, using as a timer for each entry in MFTs. Furthermore, HBH need to introduce new FUSION message. Obviously, these operations result in remarkable control messages traffic across networks. However, in SReM there is no remarkable control message traffic like REUNITE and HBH.

SReM, however, aims to overcome these disadvantages in SEM. In SReM, the format of BNMs is changed. Instead of all of destinations of a multicast session, only one BNR and one newly-joining destination are encoded in the BNMs, as discussed previously in this chapter. As a result, the BNMs have a flat size regardless of the size of the multicast group. Therefore, SReM is scalable.

On the other hand, in SReM, the multicast source is in charge of the registration process but rarely takes part in the building of the multicast tree. In fact, the source takes part in the set-
up of multicast tree only at the start stage (exactly, the searching for first level BNRs from the source), and BNRs will take the responsibility of the update and maintenance of the remained part of the multicast tree. This means the operations of joining or leaving are always implemented locally, i.e., being done between the destination triggering the join/leave and its closest BNRs in existing multicast tree. Therefore, no remarkable join/leave latency will be incurred in SReM.

Table 4.3 summarises the cost analysis of SReM, Xcast, Xcast+, SEM, HBH and REUNITE schemes. SReM, SEM, HBH and REUNITE have control plane against the Xcast and Xcast+, their cost of packet header processing is minimised. It is clear that SReM is the one with the most advantages. Furthermore, compared to SEM, SReM has less control overhead and lower join and leave latency. This will be confirmed by simulation in the following section.

Table 4.3 Cost analysis of SReM, SEM, HBH, REUNITE, Xcast and Xcast+

<table>
<thead>
<tr>
<th></th>
<th>Xcast</th>
<th>Xcast+</th>
<th>SEM</th>
<th>SReM</th>
<th>HBH</th>
<th>REUNITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast address allocation</td>
<td>none</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>none</td>
</tr>
<tr>
<td>Multicast routing state management</td>
<td>none</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Control overhead</td>
<td>none</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Overhead by increase of receivers</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>high</td>
</tr>
<tr>
<td>Extra header processing overhead</td>
<td>High</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Deployment</td>
<td>Low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Join and leave latency</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
</tbody>
</table>
4.6.1.1 Simulation environment and results

In this simulation, SReM is compared with Xcast, Xcast+ and SEM where comparison with HBH and REUNITE is excluded. This is because of the following reasons:

- Xcast, Xcast+, SEM, HBH and REUNITE is considered the most efficient and recent protocols in the area of scalable multicast routing.

- Xcast, Xcast+, SEM and SReM share the same objective in providing scalable multicast routing explicitly.

- HBH and REUNITE have not been chosen to compare to our proposed protocol SReM. This is because in both HBH and REUNITE the join/leave messages are sent periodically, which results in some unfairness and inconvenience to compare with the Xcast, Xcast+, SEM and our proposed SReM.

In this simulation, the Waxman’s probability model [67] is used to produce random network topology. The average connection degree at nodes is between 3 and 4. The link costs are chosen randomly and uniformly distributed in [1, 10]. We assume that there is one multicast source, which is randomly chosen from the routers, and some of nodes are randomly selected as LMRs, where a router becomes a LMR as long as there is one multicast receiver attached to it. The link costs between receivers and LMRs are set a fixed value of 0.1, based on the consideration of the local link cost is usually lower than that between routers. The source and routers (BNRs) which are taking part in multicasting are called as active multicast routers (AMR). The total number of AMR is changing with the joining/leaving of the member of multicast group.

In our simulation model, to deliver multicast packets, the Dijkstra algorithm [4] is used to find the shortest paths between the source and the receivers. This algorithm is often used in
routing for finding the shortest path between two nodes, such as Intermediate System to Intermediate System (IS-IS) protocol [63] and Open Shortest Path First (OSPF) protocol [49]. The tree information such as forming BNRs, down-BNRs, up-BNR, nodes cost are kept at each node dynamically. At each BNR and LMR, the possible processing includes:

i) Perform a route table look up.
ii) Partition the set of destinations based on their next hops.
iii) Replicate the packets for each of next hops.
iv) Modify the list of destinations in each of copies.

For Xcast and Xcast+, the processing will be made for multicast packets. For SEM and SReM, the processing will be made for multicast packets and control messages\(^1\). The performance of these protocols will highly depend on their respective processing.

To evaluate and compare the four algorithms considered in this thesis, three metrics are considered, which are:

- The first one is processing cost (at nodes), including costs due to header encoding / decoding, exchanging of control messages. The processing cost is occurred at nodes when processing being made for multicast packets / control messages.
- The second one is delivering cost (over links), including costs in delivery of multicast packets and control messages. The delivering cost is the one occurred at delivering multicast packets / exchanging control messages.
- The third metrics is join/leave cost. The join/leave cost is an alternative measure of latency to reflect the real delay in join/leave processing.

These metrics are given in more details as follows.

The following assumptions and definitions are used to calculate the total processing / delivering cost in a multicast tree.

\(^1\) In THIS simulation, HBH and REUNITE have not been chosen to compare to our proposed protocol SReM. This is because in both HBH and REUNITE the join/leave messages are sent periodically, which results in some unfairness and inconvenience to compare with the Xcast, Xcast+, SEM and our proposed SReM.
First, a basic packet (BP) with a size of maximum transfer unit (MTU) is used as a benchmark. The BP is a packet which has an IP header, with one source and one destination, and its payload.

Secondly, Multicast packets (MPs) are always of a bigger size than a BP due to the fact that there might be a multicast group address or a protocol header in the multicast packets. However, the control messages (CMs) (in SEM and SReM) are usually of a size of less than MTU (for SEM) or far less than MTU (for SReM). For simplicity, we treat them as multicast packets with a size of MTU.

Thirdly, assuming that the processing cost to be 1 (unit) if a packet with a size of BP is needed to be processed at a node and the delivering cost of this packet to be 1 as well if it is delivered via a link with a cost of 1.

Depending on the previous assumptions, the processing cost at node \( k \) is given by

\[
P_{\text{cos} \, k}(N_k) = M_{k,\text{MP}} \times \frac{S_{\text{MP}}}{S_{\text{BP}}} + M_{k,\text{CM}},
\]

(1)

where \( M_{k,\text{MP}} \) and \( M_{k,\text{CM}} \) are the number of destinations in multicast packets and that in control messages at node \( k \), \( S_{\text{MP}} \) and \( S_{\text{ar}} \) are the size of a multicast packet and the size of a BP (i.e. MTU), respectively. In this simulation, the values of MTU fixed at 576 octets, i.e., the minimum size of packets required for routers. When the size of multicast packets is over MTU, a fragment processing for packets is needed, which might happen for Xcast multicast packets, but not being considered in our simulation.

Correspondingly, the delivering cost for one multicast packet over a link \( i \) is given by

\[
D_{\text{cos} \, i}(L_i) = L_i \times \left( \frac{S_{\text{MP}}}{S_{\text{BP}}} + J_{i,\text{CM}} \right),
\]

(2)

Where \( J_{i,\text{CM}} \) and \( L_i \) are the number of destinations in control messages and the cost over a link \( I \), respectively.

Based on previous discussion, the average processing cost (\( \bar{P}_{\text{cos}} \)) at a node and average delivering cost (\( \bar{D}_{\text{cos} \, i} \)) over a link are given by,

\[
\bar{P}_{\text{cos}} = \frac{1}{N} \sum_{i=1}^{N} P_{\text{cos} \, i}(N_i),
\]

(3)
\[
\overline{D_{\text{cost}}} = \frac{1}{L} \sum_{i \in T} D_{\text{cost}}(L_i),
\]

Where \( N \) and \( L \) are the total number of routers and the total number of links in the network, and \( N_i \) and \( L_i \) are the number of active routers and the number of active links in the multicast tree \( i \), respectively. \( P_{\text{cost}} \) and \( D_{\text{cost}} \) denote processing cost at a node and delivering cost over a link respectively.

Finding the join/leave delay cost should be able to reflect the real network as closely as possible. In multicast, there are many factors to contribute the join/leave latency, such as how the registration (towards the source) is carried out, the quantity of multicast traffic needed to be processed at nodes and over links, the time to build the multicast tree, the size of queue etc. However, their contributions in occurred latency are varied for different multicast protocols. For Xcast and Xcast+, no processing of building the multicast tree is needed but more processing cost at nodes and delivering cost over links, while for SEM the whole multicast tree is needed to be set up in advance, compared to that for SReM where the multicast tree is built gradually and could be updated ‘locally’, and less processing and delivering cost are required for both SEM and SReM. In this thesis for the consideration of join/leave latency, we assume that the delay cost occurred in the registration processes are identical for all four protocols. This assumption is reasonable because for multicast protocol considered, the source is always in charge of tracking the membership of multicast group. Furthermore, for simplicity, the average processing cost at nodes and average delivering cost over links are further used to reflect the delay cost, but their contributions to delay cost are added together. Based on these considerations, join/leave delay cost for a given receiver \( i \) can be defined as follows:

\[
T_{\text{cost}}(i) = K_{i,\text{node}} \overline{P_{\text{cost}}} + \gamma K_{i,\text{link}} \overline{D_{\text{cost}}},
\]

where \( K_{i,\text{node}} \) and \( K_{i,\text{link}} \) are the number of nodes (routers) and the number of links involved in the receiver \( i \)’s join/leave processing, \( \overline{P_{\text{cost}}} \) and \( \overline{D_{\text{cost}}} \) are given by equation (1) and (2), respectively. \( \gamma \) is a weighting factor. If \( \gamma >> 1 \), it means the processing cost at nodes plays
the dominant role to the join/leave delay cost; if $\gamma << 1$, means that the delivering cost over
links dominate the join/leave delay cost; otherwise, the processing at nodes and delivering
over links contribute to the join/leave delay cost in the same level. Therefore, the average
joins /leave delay cost for all receivers is given by

$$\bar{T}_{\text{cost}} = \frac{1}{N_R} \sum_{i=1}^{N_X} T_{\text{cost}}(i),$$

(6)

Where: $N_R$ is the number of receivers

After discussing the simulation environment and the evaluation metrics, the simulation
results are presented. For each simulation result, the simulation was run 10,000 times for
each protocol, by changing the seed value in the Waxman’s probability model[67] used to
generate the topology to minimize the error rate in the output results and then the results are
shown by average.

Figure 4.20 shows the average processing cost as a function of number of receivers for
SReM, SEM, Xcast and Xcast+, where the number of LMRs is fixed at 20, these LMRs
represent the end routers where the receivers are directly connected to these LMRs. It can be
noticed that processing cost in SReM, SEM and Xcast+ increases approximately linearly, but
increases dramatically in Xcast. This is because Xcast encodes the receivers address in its
packets and these packets will grow exponentially when the number of receivers increases, as
a result the cost will increase. This means that Xcast faces the scalable problem and it is just
a solution for small multicast groups and other three protocols are more scalable than Xcast.

![Figure 4.20 Average processing cost as a function of number of receivers per LMR](image-url)
Figure 4.21 shows the average delivery cost as a function of number of receivers per LMR. Again, this result shows that Xcast faces the scalability problem where the other three considered protocols are better than the Xcast protocol. The high delivery cost for Xcast happens because of the big packet size delivered through the links in this protocol. The big size comes from the protocol consideration of encoding the receivers address in the packet header.

Figure 4.21 Average delivering cost as a function of number of receivers per LMR
Figure 4.22 shows the average processing cost as a function of LMRs number. It can be noticed that in all considered protocols the average processing cost increases when the number of LMRs increase. The proposed protocol SReM performs better in comparison with the other three protocols because in SReM the increase of LMRs will cause an increase in the number of BNRs nodes and the size of multicast packet is fixed, so the total processing cost will increase in small values. For the other three considered protocols, the increase of LMRs will cause high increase in processing cost because the size of the multicast packet and the number of destinations in multicast packets will increase.

![Figure 4.22 Average processing cost as a function of LMRs number](image-url)
Figure 4.23 shows the average delivering cost as a function of LMRs number. It can be noticed that SReM increase is less than the increase on the other three considered protocols. This is because this result depends on the size of the multicast packets which it will increase rapidly when the number of LMRs increases in the three considered protocol except for SReM.

![Figure 4.23 Average delivering cost as a function of LMRs number](image)
Figure 4.24 shows the average join/leave delay cost versus the number of LMRs. The join/leave delay is the smallest in comparison with the other three protocols. This result because in SReM the join/leave is done locally and dynamically where in the other three considered protocols joining and leaving of members needs tree reconstruction as in SEM which will cause a high value of delay.

![Figure 4.24 Average join/leave delay cost versus the number of LMRs](image)

4.6.2 Simulation analysis

Using simulation is very helpful when studying the performance of routing protocols before applying these protocols in practice. The simulation gives the opportunity to know the ideal scenarios for the proposed protocol. The purpose of this section and the next subsections is to study and compare the performance of the scalable recursive multicast routing protocol SReM which are discussed previously in this chapter.

The simulation implemented in this research did not consider the joining/leaving messages between LMRs and receivers or the way of delivering data packets from the LMRs to the receivers, the end point of our simulation is the LMRs. This is because the method used for exchanging packets between the LMRs and the receivers is done by standard multicast for the proposed protocol and Xcast+. 
Before starting the simulation, it is important to study the simulation environment, the simulation parameters and the performance metrics used during this simulation. Section 4.6.2.1 describes the simulation environment (i.e. the simulator used, no. of nodes etc) and the parameters values. Section 4.6.2.2 describes the metrics used to evaluate the proposed protocol. Section 4.6.2.3 presents and discusses the results obtained for the proposed protocol.

### 4.6.2.1 Simulation Environment

One of the important things in simulation is to choose the appropriate simulator to test the proposed protocol. At network environment, the simulator to be used can be OPNET, own developed simulator or ns2. In this thesis, ns2 simulator is selected because this simulator is widely used in proposed network protocols, the existence of many previously implemented protocols, the simplicity in creating scenario and the ability to run under different platforms. The reason for not selecting OPNET is that we noticed that this simulator is not ideal for implementing the multicast protocols especially ad hoc protocols. Designing and developing a new simulator needs a lot of time and it will not be very useful to be used in other previously proposed protocols.

So the proposed protocol is implemented using ns2 simulator [1] (version 2.29). The simulation environment consists of 60 nodes, these nodes represent the number of LMRs, and each LMR can carry any number of receivers depending on its configuration. In our simulation, each LMR can connect up to 10 receivers so the maximum number of receivers is 600 nodes. The number of multicast group members (LMRs) varies from 5, 10, 15, 20, 25, 30, 35, 40, and 45 nodes; these nodes represent LMRs in SReM and Designated Router (DR) for xcast+ protocol.

Traffic generation considered at this simulation is CBR traffic with payload size 512 bytes. Data packets are generated at source at a rate of 8 packets per second; this will introduce 4096 byte per second. Each simulation runs for 200 second.
In this protocol evaluation, SReM is compared with Xcast+ protocol, the selection of Xcast+ is done for the following reasons:-

- Xcast+ is widely referenced in the area of explicit multicast protocols.
- The availability of a close implementation for Xcast+ on NS2, so a modification for the available simulation is done instead of implementing this protocol from scratch.
- This protocol (Xcast+) is relatively close to the proposed protocol (SReM), so the comparison will give good indication of the performance of SReM.

### 4.6.2.2 Performance Metrics

The following metrics are used to evaluate the performance of proposed work:-

- **Average Packet Header Size** represents the size of the header included in the data packet in bytes. It represents the size of each data packet header in order to deliver this packet to all destinations.
- **Average End To End Delay**: is the average time that takes the data packet sent by source node to reach its destination node.

\[
\text{Total end to end delay} = \sum (\text{time}_{\text{received}}(\text{pkt}) - \text{time}_{\text{sent}}(\text{pkt})) \text{ for all data packets}
\]

\[
\text{Avg. End To End Delay (AED)} = \frac{\text{Total end to end delay}}{\text{no. of data packets received}}
\]

- **Normalized Routing Overhead**: represents the number of sent or forwarded routing packets to the number of data packets received by the multicast group member.

\[
\text{Normalized Routing Overhead} = \frac{\text{No. routing overhead messages}}{\text{no. of data packet received by multicast group members}}
\]

### 4.6.2.3 Simulation Results

The simulation results will be discussed in this section, this discussion is organized with regard to the metrics used for protocol evaluation. At each part, the results are presented in figure form and then a discussion and explanation for theses figures is mentioned.
Packet header size

This performance metric is to evaluate the scalability feature in the proposed protocol in comparison to Xcast+ protocol. In this evaluation, both the packet header size in the source node and for each data packet received is calculated. In the first part of evaluation, the calculations were done to find the size of the header that initially encoded in the data packet in the source node in order to send this data packet. Moreover, at the second part the calculations were done to find the average header size encoded for each data packet from the source to reach the destinations.

Figure 4.25 shows the results obtained for the first part of packet header size evaluation. It can be noticed that SReM has got static header size even when the group size is increasing. This result comes because whatever the group size SReM will only include the next Branching Node Router (BNR) addresses in the header of each data packet. In Xcast+ the results show that an exponential increase of header size when the group size increases. The increase of header size will cause energy consumption in the intermediate node and a delay of delivery packets, because the intermediate nodes will take long time and energy for processing the header. In conclusion, SReM improve the scalability feature because the header size is constant even when the group size increases.

![Figure 4.25 Extra packet header size as a function of group size](image-url)
The results for the second part of evaluation is shown in Figure 4.26, the size of packet header for SReM increases slightly but in Xcast+ a high increase of packet header is happens again. This result proofs that SReM improves the scalability feature in wired networks.

Figure 4.26 Average header size as function of group size
End To End Delay

Figure 4.27 shows the end to end delays as a function of group size. The bigger value of delay the less efficient the protocol. In reality, SReM shows lower delay in comparison with Xcast+. This is because scalability feature in SReM make the intermediate nodes to forward the data packets faster where in Xcast+ the data packets will take high processing time in the intermediate nodes which lastly will increase the delay for the data packets arrival at the end nodes.

Figure 4.27 End to End Delay.
Normalized routing overhead

Figure 4.28 shows the normalized overhead as a function of group size. It can be noticed that SReM has got constant values even when the group size grows. This is because the joining procedure of new members is done locally in BNRs and the source is informed of this join as discussed in Section 4.4.2. In Xcast+ the values of overhead are higher than values in SReM because in Xcast+ the overhead messages for joining new members must reach the source which will increase the routing overhead in the network.

![Normalized routing overhead](image)

**Figure 4.28 Normalized Routing Overhead (Load).**

### 4.7 Summary

In this chapter a new scalable recursive multicast protocol (SReM) is proposed and a performance analysis study is done. SReM is a recursive multicast protocol, which uses a recursive unicast scheme to build the multicast tree. SReM also improves the scalability issue by dynamically selecting multicast enabled nodes to be a Branching Node Routers (BNRs) in order to take part of joining /leaving and data forwarding responsibility.
The performance analysis shows that SReM scales well when the multicast group size becomes large. The results show that SReM performs a fixed header size in data packets where the header size for Xcast+ protocol increases exponentially when the group size increases. These results show that SReM improves the scalability feature in networks. Other results obtained shows that SReM performs less end to end delay and overhead in addition to scalability feature.

SReM is a scalable multicast protocol for fixed networks. The scalability issue is an important feature for wireless networks. In the next chapter, the mobility feature is added and proposed two new scalable protocols. These proposed protocols are an extension for SReM, this extension makes the new protocol able to guarantee the new features in wireless and wireless ad hoc networks.
Chapter 5

SCALABLE RECURSIVE MULTICAST PROTOCOLS IN MOBILE ENVIRONMENTS

This chapter introduces the mobility feature over the scalable recursive multicast protocol (SReM). A mobile scalable recursive multicast protocol (MoSReM) and a scalable ad hoc recursive multicast protocol (SARM) are proposed. It starts with a description of each of the proposed protocol, followed by a detailed description for the membership management and branching nodes selection as critical features. Then, data structures of the proposed protocols are discussed. Finally, the simulation results are presented and discussed.

5.1 Introduction

Multicast schemes provide a very useful mechanism to satisfy group communication with efficient bandwidth utilization. In particular, multicasting is an important scheme in wireless and wireless ad hoc networks because many wireless applications depend on group communication with bandwidth constraints in these networks.

Different categories of multicast protocols have been proposed in the literature review for ad hoc environment, including Tree-based, Mesh-based and Explicit approaches. These categories have been discussed in details in the literature review chapter (Chapter 3) of this thesis. This chapter introduces two protocols called; mobile scalable recursive multicast protocol (MoSReM) where we consider mobility feature at the edges of the network and a scalable ad hoc recursive multicast protocol (SARM) where the mobility feature is considered at all the nodes in the network.
According to [16] [30] [5], explicit multicast schemes are suitable for ad hoc networks, where there are frequent changes in the network topology. These schemes differ from traditional multicast schemes in that they are trying to reduce the amount of routing information kept in the group members to perform multicast routing. A number of routing protocols, discussed before, have been proposed based in this type of networks like DDM [20], E2M [38] and EM2NET [7].

DDM [20] uses an extended header to include the list of destinations and their next hops. Each next hop is attached to a list of corresponding destinations and these lists are encoded into the Xcast header packet and broadcasted towards the source neighbours. The nodes in the neighbourhood which receive this packet will process the header and check if they are part of the next hop list. It is worth noticing that all the nodes in the neighbourhood will process the received packet header. This issue will cause high overhead in processing and delay in forwarding packets. In our proposed work, SARM, only a set of nodes are involved in processing the packets and the remaining node just forward the packet without any processing procedure. This feature will decrease the time that the packets spend to be received by the destinations.

E2M [38] is proposed to improve the scalability issue in multicast ad hoc networks. It employs the concept of Xcast Forwarder (XF) which is selected dynamically during the message forwarding procedure. The XF selection depends on predefined threshold which represents the number of destinations served by this XF. The major drawback in E2M is that it does not solve effectively the node movement issue in ad hoc networks, especially when XF moves which happens frequently in MANETs. In E2M, an XF moves in a way that cause link failure, the source will be notified and all the destinations served by this XF will be inserted into the next packet header created by the source node. This solution is not efficient because the route is not already established for these destinations which will cause a packet loss in the served destinations. In our proposed protocol in this thesis, the node mobility issue is solved by performing local search for new connection, because of link failure related to node movement. This solution will decrease the time for
establishing a new connection instead of the broken one. This issue will be discussed in detail in Section 5.4.

EM2NET [7] is a routing protocol for explicit multicast ad hoc protocols, which employs the idea of Intercepting Nodes (IN) to serve a set of destinations in the network. These INs will support load distribution among nodes also this proposed protocol gives a suggestion of how to detect link breakage caused by node mobility. EM2NET [7] has similar objectives to our proposed protocol in the way of data packet delivery. However, our proposed protocol uses different types of messages to build the multicast group members tree where the details has not been mentioned in EM2NET. SARM also has the originality of building the tree gradually and recursively which can be important feature for ad hoc networks. Additionally, our proposed work, efficiently covers the joining and leaving of members when node movement happens with low latency.

5.2 Inherited features from SReM

The new proposed protocols are extension to the previously discussed protocol SReM. Hence, there are some features that are inherited from SReM and used by the new proposed protocols, which are as follows:

- **Tree construction and maintenance**
  The proposed protocols use the dynamic branching node-based tree (DBT) to establish the multicast tree, where multicast tree is processed gradually and dynamically among branching nodes as the joining /leaving of members of a multicast group, not in advance. DBT is set up by the use of a pair of Branching Node Message(s) BNMs, i.e., enquiring BNM (eBNM) and replying BNM (rBNM). BNMs carry the information such as the previous BNR, one next BNR, and new LMR, as detailed in the previous chapter (see Section 4.3.2 & Section 4.3.3). These messages also used for tree maintenance in the case where the join/leave of nodes will cause changes in the tree structure discussed in Section 4.4.4.
• **Joining/leaving signalling messages**

Three types of signalling messages inherited from SReM, which are as follows:

- **Join / leave messages**
  These messages are used to establish the joining/leaving procedure of nodes in the multicast tree. They are JoinM/LeavM messages discussed previously in Section 4.3.2.

- **Registration messages**
  These messages are to complete the joining/leaving process and perform the registration process of new node joining the multicast session. These messages are RqM,RpM discussed previously in Section 4.3.2.

- **Branching node messages**
  These messages are implemented to build and maintain the DBT for multicast session. These messages are eBNM, rBNM discussed previously in Section 4.3.2.

• **Packet delivery process**

To start multicasting data packets the source will send a copy of packets with respective MoSReM headers to all of branching nodes in its MFT. Upon receiving this packet, each of branching nodes needs to properly process the MoSReM header. The standard processing for a branching node is discussed in Section 4.6. Furthermore, the new proposed protocols use the same header format discussed in Section 4.6.

In the rest of the chapter, the new features for each of the proposed protocols MoSReM and SARM are discussed in details.

5.3 **MoSReM Protocol Description**

One of the protocol design issues that arise in the field of wireless networks is the mobility management. Designed protocols must be able to guarantee the ability for mobile receivers (hosts) to move from current point of connection to another new point of
connection smoothly and loss-free. MoSReM is a proposed protocol that falls in the scope of handover protocols, which introduces the mobility feature in the edges. MoSReM stands for Mobile Scalable Recursive Multicast protocol and it is an extension to SReM protocol (discussed in Chapter 4).

Providing multicast services to mobile nodes in an IP internet-network [41] [46] [45] has acquired a high attention from researchers because of the high demand on using wireless devices (hosts) for Internet applications. Some challenges are facing the mobile multicasting approach.

- First, IETF Mobile IP [52] supports only unicast delivery, so to support multicasting, additional mechanisms need to be added.
- Second, additional mechanisms should be added to support the dynamic changes of the topology because of receiver node movements.
- Thirdly, existing multicast routing protocols, such as DVMRP [66], MOSPF [49] and PIM [24], implicitly assume static hosts when setting up a multicast tree.

In this part of thesis, it has been proposed that mobile scalable multicast protocol (MoSReM) is built as an extension of Scalable Recursive Multicast protocol SReM. The basic idea of MoSReM is to use the concepts of dynamic branching node-based tree (DBT) to build a multicast tree gradually and dynamically. MoSReM aims to address scalable and join/leave latency at Mobile IP networks.

5.4 MoSReM Details

In the following subsections, the process of mobility in receivers’ i.e roaming process as performed by MoSReM is discussed.

5.4.1 Roaming messages in MoSREM

Mobility management in receivers is the main new feature in the proposed MoSReM protocol. To be able for MoSReM to guarantee this feature new messages are introduced and called the roaming messages. These messages include two main controlling messages: roaming in (Rm_in) and roaming out (Rm_out). They are used to perform
smooth roaming process. To implement these messages in MoSReM, there are two proposed ways as follows:

- First, the old Local Multicast Router (oLMR) will treat this roaming out message as an ordinary leave message and the new Local Multicast Router (nLMR) will treat the roaming in message as an ordinary joining message. This is done by sending a RqM message towards the upstream BNR, the BNR receiving this message should be able to distinguish between the ordinary join/leave messages and the join/leave messages because of roaming. This is done by adding a field as a flag (R) to the RqM message, this field is set to 1 (R=1) if this message regarding a roaming process and the message will not be forwarded to the source, otherwise (R=0) regarding an ordinary joining/leaving messages. One of advantages of using this way is: the control message traffic due to receivers’ roaming is greatly reduced because roaming messages never traverse within the multicast tree. Another advantage is that these messages use similar format as JoinM/LeavM with minor modification. However, the disadvantage of this way is that a modification with respect to join/leave and registration messages is needed for the compatibility between MoSReM and its original version SReM.

- Second way by treating the roaming messages as a new type of messages. The designated LMR just forwards the roaming message towards its up-level BNR. Upon receiving these messages, the up-level BNR updates its MFT and decides if it is still a BNR or not. If yes, this BNR will stop forwarding these messages and there is no more impact on the remaining DBT. Otherwise, these messages will be forwarded to the next up-level BNR, say Y, which start a DBT update process just like join/leave process in SReM. Note that forwarding these messages will be terminated at Y. The advantage of this way is that it is fully compatible with SReM and there is no need to modify the join/leave and registration messages. The disadvantage is that it requires a fully new message format to be used, might quite different from JoinM/LeavM or RqM. Using new format messages will cause more complexity in roaming process, esp., at the time when leaving and
roaming process happen together. In this case, LMRs must be able to decide the priority whether to deal with leaving process first and then roaming process or visa versa.

For each type of the previously discussed signalling messages a specific task is carried out. In the next section there will be a detailed discussion of the role of these messages.

5.4.2 Roaming messages roles in MoSReM

The tasks held by roaming messages are to guarantee the change of point of attachment for mobile receiver in MoSReM. These tasks can be divided into two types as discussed in the next two subsections.

5.4.2.1 Registration/ De-registration process

Registration and de-registration of receiver node with the multicast source is done by LMRs on behalf of mobile receivers decide to join/leave the multicast group in the ordinary case (no roaming). In contrast, when a mobile receiver is roaming, further consideration is needed. The new LMRs in the existing multicast session do not need to re-register mobile receivers. However, the new LMRs should recognize the incoming receiver and register this incoming receiver with the new LMR for multicast service. To solve this issue, it is needed for the new LMR to be able to obtain information (from the Rm_in message) about incoming receivers’ qualification to receive multicast packets. For this, the Rm_in message should carry an identity field for the roaming in receiver to prove that it is authorized to receive multicast data packets without re-registering with the source.

5.4.2.2 Roaming process

The third type of messages discussed in Section 4.3.2 is involved in roaming process in MoSReM. Additionally, it is important to understand that we assume that the actual physical layer handover over process is handled by the appropriate mobile network protocols. Examples of such protocols are HAWAII [55], Cellular IP [15] and Helmy [34, 35]. The process of implementing the movement is out of the scope of this thesis. Briefly, the following steps are to satisfy the roaming process procedure:
The mobile receiver sends a Rm_out message to the old LMR (oLMR).

oLMR starts a procedure of updating the DBT if this mobile receiver is the last receiver attached.

The mobile node sends Rm_in message towards the new LMR (nLMR).

The nLMR starts a DBT update procedure if this mobile receiver is the first receiver attaching to the nLMR.

The nLMR starts the registration procedure for the mobile node; the registration procedure is similar to join/leave procedure in SReM except that this registration will not be sent to the source node.

Depending on the MDT state of x’oLMR and the x’nLMR, two scenarios will be found. The first scenario is that the mobile receiver is not the last receiver attached to the oLMR and it is not the first mobile receiver attached to the nLMR which means that there will not be any DBT update. The second scenario is when either the mobile receiver is the last receiver attached to the oLMR case or it is the first mobile receiver attached to the nLMR case or both cases which means that there will be a DBT update.

In the first scenario, the mobile receiver, say x must be not the last one attached to the oLMR and not the first one attached to the nLMR. x will send a Rm_out message towards its oLMR. Upon receiving this message, the MDT for the x’s oLMR will be updated by removing the x’s entry. No changes will happen to the existing DBT regarding this mobile node roaming out. Similarly, x will send a Rm_in message towards its nLMR when x in the domain covered by its nLMR. Upon receiving the roaming in message, the nLMR MDT will be updated by adding an entry for the mobile receiver x. Similar to the previous case no multicast DBT session update will happen. As mentioned previously, nLRM is able to be aware of the x’s membership belonging to multicast groups. Therefore, from then on, mobile receiver is able to receive multicast packets directly from its nLMR.
In the second scenario a DBT change has happened because of roaming of mobile node x. This scenario can happen in one of the following cases:

- **Case 1**: the mobile receiver x is the last receiver attached to oLMR and it is not the first mobile receiver attached to the nLMR.
- **Case 2**: the mobile receiver x is not the last receiver attached to oLMR but it is the first mobile receiver attached to the nLMR.
- **Case 3**: the mobile receiver x is the last receiver attached to oLMR and it is the first mobile receiver attached to the nLMR.

All of these cases will cause a DBT update, including BNRs update and MFTs update at BNRs.

Starting with case 1, Figure 5.1 shows this case; when roaming is transport to the receivers, it will send an Rm_out message towards its oLMR (M2). Upon receiving this message and checking its MDT, x’s oLMR (M2) will be aware that it will no longer be a multicast branch. As a result, N3 will delete its M2 entry in its MFT for the multicast
group and then send a DBT update message; at the same time N3 will discover that it is no more a BNR. Hence, this DBT needs update by removing N3 as a BNR and inform the upstream BNR (N2) of this update. Upon receiving the update message, N2 will update its MFT table by removing the N3 entry and adding an entry for M1 as a new branch served by N2. This update process is the same process used in SReM for leaving a receiver with DBT update. The Rm_in message sent by x to the new point of attachment (M3) will not cause any DBT changes.

In case 2 shown in Figure 5.2, when the mobile receiver x detects that it is moving from oLMR(M2) domain to a nLMR(M3) domain, it will send a roaming messages Rm_out to M2 and Rm_in message M3. Due to x is the first member of the multicast group connecting through M3, M3 will set up a MDT for the multicast group and simultaneously send a DBT update message (as mentioned above, it might be a modified RqM or Rm_in) towards the multicast source. One of BNRs(N1) in the existing DBT will receive the DBT update message. N1 will then be responsible of the DBT update process by the use of the pair of BNMs messages. Meanwhile, all the MFTs at the BNRs

---

As aforementioned, the DBT update message could take a form of modified RqM message or directly use the Rm_OUT message.

107
(existing (N1) or newly (N2) created) will be updated. After the completion of the DBT update, the mobile receiver x is then able to receive the multicast packet from the nLMR.

In case 3 shown in Figure 5.3, both of the roaming messages will cause a DBT update because the receiver x sending the Rm_out message is the last one connected to M2 and will cause N3 to be no longer a BNR. And the Rm_in message for the receiver x will be the first receiver connected to M3 which will cause new multicast branch in the DBT. This case roaming process will cause N3 to be no longer a BNR and also N2 (previously a BNR) will update its MFT table.

This proposed protocol has been presented as a contribution to the COST 290 Final Report and was included as a book chapter in the book published by Springer Lecture Notes in Computer Science Series [70].

5.5 Scalable Ad hoc Recursive Multicast (SARM) protocol

SARM is an extension to the previously proposed protocol SReM taking into consideration the mobility feature in all multicast nodes. In ad hoc networks, the ability to keep hosts connected and re-establish new route because of node movement is the main challenge. In the proposed protocol, it has been proposed a solution to re-establish new
route, when a route failure happens. This solution will be discussed in details later in this chapter.

In similar fashion to SReM and MoSReM, SARM builds tree connecting group members, these members called a multicast session. This multicast session consists of:

- Multicast source node (S).
- A set of nodes called receivers.
- An multicast enabled intermediate nodes called Intermediate Multicast Routers (IMRs)

5.5.1 Mobility messages in SARM

Performing mobility management in nodes and the ability to re-establish a new route, when a link failure happens, is the key issue of the proposed protocol SARM.

As mentioned, the proposed SARM protocol is designed to take into consideration the mobility in ad hoc networks. As soon as a link failure is detected by the periodical messages (discussed later) or by the network layer, a local search will start to find a new route instead of the broken one. The mobility feature will be discussed in detail in Section 5.6. To perform the mobility a new type of messages is introduced. These messages are used to detect and repair a link breakage in the network. These massages are as follows:-

- BNR_UP:- this message is sent by a BNR to its upstream (previous) BNR stored in previous BNR field in its MFT table. This message is sent periodically to refresh the connectivity of BNR toward previous BNR. Figure 5.4 shows this message format.

<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>15</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source (Current BNR IP address)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination (previous BNR IP address)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.4 BNR_UP message format
• BNR_Down :- this message is sent by a BNR to its downstream BNRs and LMRs stored in its MFT table. This message is sent in periodical time interval to refresh the links between the BNR and its downstream members. Figure 5.5 shows BNR_Down message format.

<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>15</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source (Current BNR IP address)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination (next BNR/LMR IP address)[1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination (next BNR/LMR IP address) [N]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.5 BNR_Down message format**

• Link_failure:­­:- this message is sent by the BNR to its downstream BNRs and LMRs stored in its MFT table, which is used to notify them that a link failure to the upstream nodes has happened. This message is created and sent if the BNR does not receive a BNR_Down from its upstream BNR. Figure 5.6 shows the Link_failure message format.

<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>15</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source (Current BNR IP address)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination (next BNR/LMR IP address)[1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination (next BNR/LMR IP address) [N]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.6 Link_failure message format**

Regarding the new type of messages, new operations are implemented for SARM to guarantee the multicast tree connectivity. These operations can be called the recovery operations, because these operations are used to recover the construction of multicast session after link failure because of mobility. These operations will be discussed in the next section.
5.6 Handling node Movement

In order to handle node mobility in ad hoc networks, SARM uses periodical messages discussed before. These messages take the responsibility to detect any changes in the multicast tree because of node movement.

SARM protocol uses a set of operations to guarantee the mobility feature. These operations are:-

- Each BNR sends a periodical BNR\_Down message to its next BNRs stored in its MFT table. The time interval between messages is defined by BNR\_Down\_refresh value.
- Each BNR sends BNR\_UP message to its previous BNR stored in its MFT table. This message is sent periodically every BNR\_UP\_refresh period of time.
- Each LMR exchanges a periodical member refresh message with its BNR. This message will guarantee the connection of LMRs to their BNRs. Whereas, the first two messages will guarantee the multicast tree session connectivity.

By performing the previous operations the multicast group session will stay connected to each other.

If any of BNRs, for example X, does not receive any of the periodical messages (BNR\_UP, BNR\_Down) in a predefined time interval, it assumes that a link failure has happened upstream or downstream, respectively.

According to node mobility management, link failure happens if either the upstream node does not receive a BNR\_UP message from its downstream BNRs and LMRs (case1) or the downstream BNR node does not receive a BNR\_Down from its upstream BNR (case2). Next each case will be discussed in details.
• **Case 1**
In this case, a BNR node (for example X) stopped receiving BNR_UP message for a period of time exceeding a predefined threshold from its downstream node. Hence, X will perform the following steps:-
  - Updates its MFT table by removing entry for the downstream BNR that stop sending BNR_UP messages.
  - X checks its state after updating its MFT, if it is no longer a BNR it performs a leave message regarding the state change. This message is similar to what discussed in the third case Section 4.4.3.
  - If X stays a BNR after updating its MFT, then it, as a result, will stop sending data packets and other periodical messages towards the disconnected BNR.

• **Case 2**
This case happens if a BNR (for example X) does not receive a BNR_Down message from its previous (upstream) BNR. This means that node X and all nodes connected to it are now disconnected from the tree and a sub-tree is created, which is rooted by X. The following steps are followed:-
  - Node X stops sending the periodical BNR_Down to its downstream nodes.
  - Node X sends immediately a Link_failure message to its downstream nodes informing them of this link failure and to prevent these nodes from starting a repair procedure because they will not receive a BNR_Down message.
  - Node X starts searching for new route to the multicast tree. It broadcasts a route request message to find an upstream node to re-establish the connection to the tree. This message is propagated and controlled by the underlying unicast protocol (e.g., AODV), by controlling the direction of this message to be upstream and to find a node with a route to the source or any upstream group member. After a period of time node X will receive one or more reply messages from its neighbours carrying a route to upstream member nodes, it selects one according to some parameters like the number of hops and any other parameters required for certain applications. Node X will wait for a period of time,
WAIT_TIME, after receiving the first reply in order to receive more than one reply and select the best one according to present criteria.

- After choosing the path, node X will send a BNR_Down to refresh the downstream members and sends upstream an eBNM message to find the previous BNR and update its MFT table with this new connection.
- The upstream BNR will reply with rBNM to the new BNR and update its MFT.

At the end, the new tree construction is ready and the data packets flow will continue.

5.7 Performance evaluation

5.7.1 Simulation

SARM protocol has been implemented in NS (version 2.29) simulator [1]. The environment for this simulation comprised of 60 mobile nodes which are randomly placed in a grid of size 1500m by 800m using random waypoint model [13]. For each node after a pause time value assigned in advance a new destination and speed is given to each node, and then each node moves towards the new destination with the assigned speed. After a node reaches its destination, the same process is repeated until the end of simulation. In this simulation the pause time changed with fixed group size. The pause times used are varied between 0 & 400 seconds with 50s steps. Pause time 0 second means that the nodes are in continuous movement. The value of the speed given to each node is selected randomly from uniform distributed values with minimum value 0 m/s and with maximum value of 10m/s. The network stack of each mobile node consists of link layer, an ARP module, interface priority queue, IEEE 802.11, MAC layer with 250 meters transmission range, and a network interface.

Table 5.1 outlines the different simulation parameters used in the simulation.
In the following investigations, we have considered Continuous Bit Rate (CBR) traffic with payload size set to 512 bytes. Data packets are generated at the source node (a selected node from the multicast session to be a source node) at a rate of 2 packets per second which is equal to 1024 byte per second. Each simulation runs for 500 seconds.

DDM [20] and E2M [38] have been chosen as a comparison protocol, this selection is based on the following:

- A detailed description of E2M and DDM protocols is available.
- E2M is the most recent implemented explicit multicast protocol.
- DDM is relatively close to our proposed protocol.
- The availability of close implementation for DDM, E2M in ns2 simulator. Although, some modifications on these implementations to represent DDM and E2M protocols were needed.

Despite that EM2NET is the most recent proposed protocol addresses the scalability issue in ad hoc networks, it is excluded from the comparison with our proposed protocol. This
is because there is no information mentioned about this protocol regarding the multicast tree construction also there are no details about the messages format and role for joining/leaving the multicast tree. For the previous reasons, the implementation of EM2NET will not give the real behaviour of the protocol.

In the investigated scenario, SARM, E2M and DDM run over the MANET unicast routing protocol AODV (Ad Hoc On-Demand Distance Vector) [16]. NS-2 simulator already provides the AODV routing agent implementation, while a modification for AODV agent is done in order to make AODV agent able to exchange routing information with the Xcast agent. To facilitate the interaction between the Xcast agent and the AODV agent, cross-layer design approach has been adopted, wherein an Xcast agent can access the AODV routing table to group the nodes which are reachable through the same next hop. In case a route is not available, the Xcast agent simply sends a unicast packet to the AODV agent, which triggers a route request for this destination. When the AODV agent receives a route reply from the required destination, it updates its route table accordingly. An Xcast agent can now directly obtain the next hop information from the routing layer.

Table 5.1 outlines some of the AODV parameters selected for our simulation work. The performance metrics considered in this proposed work are extra packet header size, packet delivery ratio and end to end delay. These metrics used to investigate the scalability and the efficiency of the proposed protocol.

5.7.2 Results and discussion

This section will discuss the results obtained from the simulations; these results will be organized according to the metrics used in the evaluation. For each value obtained, the simulation has run for 10 times and the average of the output values taken. Also an error rate is calculated which equals the standard deviation for the 10 output values. For each evaluation metric, the simulation is divided into two groups. In the first group of simulation, the group size is fixed to 30 nodes and the pause time value is varied, whereas in the second group the multicast group size is varied with fixed pause time value to 0s.
The first group of simulation is to investigate the mobility feature of the protocol. Where, the second group of simulation is to investigate the scalability of the protocol.

5.7.2.1 Packet header size

This metric can be considered as a good indication of the scalability feature for any protocol. A protocol according to this metric is considered as scalable protocol if the growth of group size is not followed by growth of header size. Figure 5.7 shows that SARM provides superior scalability compared to the other two protocols. The results obtained for extra header size added to the data packet in order to be delivered to the group nodes. The results show that there is an increase in the header size for all the protocols included in the evaluation. However, the increase in SARM is small in comparison to DDM and E2M; this is because the extra header size in SARM contains only the address of the next BNR and no need to carry any extra address in the header. In E2M and DDM, the extra header size will include the addresses of the destinations for DDM and part of the destination and XFs for E2M. Also it can be noticed that E2M performs better than DDM, this is because E2M encodes part of the destinations addresses and XFs nodes, which relatively the number of theses nodes will be less than the number of the whole destination nodes.

![Packet Header Size](image)

*Figure 5.7 Packet header size for each data packet sent*
5.7.2.2 Packet delivery ratio

The packet delivery ratio is the number of packets that each group member received to the number of packets sent (should be received). Figure 5.8 shows a comparison of packet delivery ratio for the proposed protocol (SARM) with two protocols (E2M, DDM).

![Packet Delivery Ratio as a function of Pause Time](image)

**Figure 5.8 Packet delivery ratio as a function of pause time**

SARM achieved more packet delivery ratio even in high mobility values. This ratio increases when the pause time increases. The reason for the high ratio in comparison with E2M is that SARM, when taking mobility into consideration, will deliver more packets because the probability of failure recovery is better than in E2M. Other reason is that in E2M it is expected that a huge traffic will be in XF, which may cause packets drop because of queue overflow in that XF. Moreover, SARM performs better than DDM, because in DDM there is no efficient solution to the mobility issue, so when the nodes moves this protocol will suffer from data packet loss.
Figure 5.9 shows the packet delivery ratio when the multicast group size changes. This evaluation gives another indication on the scalability degree of a protocol. SARM shows more scalability over E2M and DDM. The reason for this improvement in scalability is the use of BNRs in SARM.

In E2M, when the number of group members increases, it will cause a huge traffic on XFs which will cause loss of packets because of this traffic. At DDM, increasing the number of group members means that this protocol will take time to add the new members to the multicast list kept in the source. During the addition of new members these members will not receive data packet which in conclusion the number of received packets will decrease.
5.7.2.3 End to End Delay

Figure 5.10 shows the end to end delay evaluation as a function of pause time; the figure displays the values for SARM, E2M and DDM. The evaluation result shows that the proposed protocol has less end to end delay values in comparison with the other two protocols. These results because at SARM there is a load distribution of receivers in BNRs for forwarding the traffic to the receivers. Because of the use of XFs in E2M, the traffic will wait in these nodes until it can be delivered, which will cause delay. In DDM the delay comes from the processing time that required by the intermediate nodes on the data packets.

![End to End Delay Graph](image)

**Figure 5.10 End to end delay as a function of pause time**
Figure 5.11 shows the delay time for the packets to be received by the receivers when the group size increases. The results show that SARM performs less delay time to perform the packet delivery and this delay time increases slightly when the group size increases. Whereas, in E2M and DDM one can notice that the values increase in high values when the group size increases. SARM performs better in this evaluation because SARM has the load distribution feature which will cause faster flow of packets. In E2M the delay comes from the traffic waiting in the queues in the XFs. DDM delay comes from the high processing time required in the intermediate nodes.

![End to End Delay Graph](image)

**Figure 5.11 End to end delay as a function of group size**

### 5.8 Conclusions

This chapter proposed a new ad hoc protocol; the proposed protocol supports the mobility in multicast ad hoc networks. This protocol improves the scalability by performing joining and leaving of nodes to be locally and by making the header in the data packets with fixed size. The proposed protocol also introduced an efficient solution for link failure in BNRs because of node mobility.
The proposed protocol (SARM) is implemented using NS2 simulator and an extensive performance study is undertaken to measure the scalability and the impact of mobility on this protocol. The results obtained, in comparison with other protocols in the same area; show that the proposed protocol is a scalable protocol with high packet delivery ratio with reasonable overhead values.
Chapter 6

CONCLUSIONS AND FUTURE WORK

The ability to guarantee efficient data delivery with the increasing number of users is a major challenge in designing routing protocols for wired and wireless networks. Any number of Internet users using fixed or mobile devices should be able to connect to the Internet and expect to get the information they need regardless to the underlying routing protocols running in the network routers. This ability of assurance in delivering required information with increasing number of users resulted in people designing multicast routing protocols taking into consideration the scalability feature in their design.

6.1 General Evaluation

The increase in the number of users requesting access to multicast services will increase the amount of information stored in the source node and intermediate routers to guarantee data delivery. This huge increase of information will cause more delay in forwarding data and will reduce the number of successfully delivered data packets. One of proposed solutions to this issue is to reduce the amount of information stored about users by using explicit messages to carry part of this information during the data delivery.

Providing scalable multicast in the Internet has attracted attention from the research community working in the networking area. This interest creates has grown with the high increase in the Internet users, this can be concluded from the background and related work discussed in Chapter 2 and Chapter 3 in this thesis. In these chapters, it can be noticed that most of the efforts made were to reduce the state information kept in the group members to provide multicast packet delivery. The proposed solutions for reducing the state information are mostly achieved by moving part of the information stored in the
multicast tree member nodes to the header of the control and data packets. However, these solutions create a new dimension in the scalability problem as the group size can become too big. As the header size rapidly increase, this will increase the routing overhead in the network as well as increase the processing time of the packet at each node.

6.2 Contributions of the thesis

This thesis provides solutions based on the Dynamic Branching Tree (DBT) concepts. These proposed solutions include:-

- **Scalable Recursive Multicast protocol (SReM)**, this protocol is proposed to provide a scalable multicast packet delivery over wired network. The cost analysis and the simulation proved that this protocol decreased the processing cost, delivery cost, header size, end to end delay and increase the packet delivery ratio. In conclusion, this protocol provides an efficient scalable multicast data packet delivery.

- **Mobility Scalable Recursive Multicast protocol (MoSReM)**, this protocol is proposed to provide a mobility for the end hosts which guarantee an efficient roaming process.

- **Scalable Ad hoc Recursive Multicast protocol (SARM)**, this proposed protocol provides an efficient solution for scalability issue in ad hoc environment where all nodes are in movement. The simulation results show that this protocol improves the packet delivery ratio, end to end delay and decreases the packet header size.

Through this thesis, the scalability feature of the proposed protocols has been tested and compared to other representative protocols in the wired and wireless ad hoc networks. It can be concluded that the new proposed protocols gave an efficient improvement for scalable multicast data packet delivery for wired and wireless ad hoc networks.

Based on the cost analysis and simulation based for the proposed protocols and performance evaluation using ns2 simulation environment, the following points can be concluded
While working on this thesis, the scalable recursive multicast protocol (SReM), has improved the scalability feature by introducing the branching node routers (BNRs) to keep the routing state information instead of all group members. Also this protocol reduced the join/leave implosion by presenting the locally and dynamically join/leave process. These points have not been studied together and have not been evaluated before.

In SReM, it can be observed that the average processing and delivering cost is better than other protocols in the same area of study. This is because of the reduction in the header size which made the cost for packet processing and delivering small.

The recursive build of tree used in the proposed multicast protocols made the join/leave cost small. The joining and leaving of group members is done locally and dynamically which cause the average cost of this process the minimum between the other related protocols.

The presented recursive multicast protocol is scalable with the number of members in the multicast group (i.e., group size). This scalability can be observed from the stability of the packet header size in the protocol with the group size, and the reduction in the end to end delay and the normalized overhead with the group size. The end to end delay of SReM compared to Xcast+ is about 10%, moreover, the normalized routing overhead of SReM compared to Xcast+ is about 8%. These quantitative comparisons give a conclusion that SReM is a promising approach in the area of scalable multicast routing protocols.

It was observed that the roaming process presented in the scalable recursive multicast protocol guarantees the mobility feature over receivers. It can be observed that roaming process will not affect the improved scalability feature in the original protocol.
In the scalable ad hoc recursive multicast protocol, it can be observed that the scalability feature is satisfied. The scalability can be observed from the tinny increase in the packet header size with the group size. Also small values of the packet delivery ration and the end to end delay with group size in comparison with the other protocols in the same area of study. It can be conclude that end to end delay of SARM compared to E2M and DDM is about 5% when group size changes and 6% when pause time changed. We can conclude that SARM is an efficient scalable multicast protocol for ad hoc networks.

The scalable ad hoc recursive multicast protocol is also reliable with the node movement. It can be observed that the packet delivery ratio is high even when the value of node movement speed is high. Also the end to end delay had good values with the node movement.

6.3 Future Work

This thesis finds a solution to a challenging problem of scalable multicast in wired and ad hoc networks. However, there are still some open research points that can be investigated further to extend the presented approach done in this thesis. This section gives some suggestions for future work as follows:

- In scalable recursive multicast protocol, points like quality of service (i.e., bandwidth, delay, and jitter) and security need further investigations as a future work.
- Implementing cache in branching node routers (BNRs) is an interest point of research. At this point the contents and the size of cache should be widely investigated and studied. Also this point will add the reliability feature to the protocol.
- In mobile scalable recursive multicast protocol, a detailed and extensive evaluation can be done to improve the efficiency of this protocol.
- In scalable ad hoc recursive multicast protocol, the broadcast process in link recovery when a link breakage happened can be reduced by applying a kind of
probability broadcast algorithms. This suggested work will reduce the overhead during the link recovery and will also reduce the recovery time.

- Reliable link between BNRs can be an interested point of study, this point can be done by investigating the way to maintain a spare link between each two BNRs and to be used when the original one breakdown.
REFERENCES


