

TOPOLOGY-AWARE MULTICASTING IN PHASED MOBILE AD HOC NETWORKS
FOR E-LEARNING

Leonard Edmond Khaiko Chalemba

Thesis Presented for the Degree of
DOCTOR OF PHILOSOPHY
in the Department of Electrical Engineering
UNIVERSITY OF CAPE TOWN

August 2012

ABSTRACT

This thesis presents an adaptive multicast routing protocol called a Topology-Aware Polymorphic Multicast Routing Protocol (TAPMRP) for use in Mobile Ad hoc NETWORKS (MANETs). TAPMRP is proposed to serve E-Learning classroom (ELC) environments, which are generally characterised by minimal topology changes and distinct topologies. The research is motivated by the need to support students' learning efforts with MANET multicast services such as video streaming, offline websites and file sharing in a reliable and efficient manner.

Currently, reliable delivery of MANET services is challenged by, among others, the limitation of the battery energy that powers the MANET devices and the limitation of the bandwidth capacity of the shared transmission medium. These challenges impose efficiency requirements on a MANET multicast routing protocol. The existing MANET multicast routing protocols have been shown to have excessive control overhead especially in the form of broadcast redundancy during route discovery operations. In scenarios of multiple sources, broadcast redundancy has the possibility of leading to a more serious problem known as the broadcast storm problem. The ELC environment is envisaged to be highly interactive with very high probability of having many simultaneous sources of multicast traffic.

The approach to address the research problem makes use of the observation that a MANET that is used to serve the ELC environment has the tendency of transforming itself into three distinct topologies, namely matrix, cluster-based and random topology, depending on the activities performed by the students. The network connectivity characteristics of these three topologies were exploited by deriving a highly efficient route discovery method suitable for each of the topologies. The resulting three route discovery methods are integrated and constitute TAPMRP. The route discovery methods are designed to operate one at a time depending on the prevailing topology. TAPMRP was tested by computer simulations which were conducted in GloMoSim-2.03. The results confirm that TAPMRP is more efficient in terms of network resource usage in scenarios of multiple simultaneous sources of multicast traffic than On Demand Multicast Routing Protocol (ODMRP).

DEDICATION

In memory of my parents
Hine Chalemba and Lizzie Chilapula

University of Cape Town

ACKNOWLEDGEMENTS

I would like to acknowledge the financial support rendered to this research undertaking by my employer, Mzuzu University, in Malawi. I would also like to express my gratitude to my supervisor, Dr Alexandru Murgu, for his valuable advice throughout the period of my research. My thanks should also go to Dr Golden Msilimba of Mzuzu University who encouraged me to embark on this research.

On a personal level, I would like to thank my wife, Janet, for her moral support during the time of this research. I also need to thank my two sons, Akuzike and Khaiko, for bearing with my long absence from home.

CONTENTS

Abstract	ii
Dedication	iii
Acknowledgements	iv
List of Figures	x
List of Tables	xiii
Nomenclature	xiv
1 Introduction	1
1.1 Background.....	1
1.2 Overview of MANETs.....	3
1.3 Cost of Route Discovery in MANET Multicasting.....	6
1.4 Problem Statement.....	8
1.5 Research Hypothesis.....	9
1.6 Using an Adaptive Multicast Routing Protocol.....	9
1.6.1 Topologies of the ELC-MANET.....	10
1.6.2 Topology-Aware Polymorphic Multicast Routing Protocol.....	13
1.7 Overall Aims of the Research.....	14
1.8 Thesis Road-Map.....	15
2 Problem Formulation	16
2.1 Introduction.....	16
2.2 Review of MANET Multicast Routing Protocols.....	17
2.2.1 Unicast routing protocols.....	17
2.2.2 Multicast routing protocols.....	18
2.3 Current MANET Broadcasting Methods.....	34
2.3.1 Clustering without CDS.....	35
2.3.2 CDS without clustering.....	36

2.3.3	Clustering combined with CDS.....	37
2.3.4	Other methods of broadcasting.....	38
2.4	Existing MANET Clustering Methods.....	39
2.4.1	Review of active clustering methods.....	40
2.4.2	Basagni’s distributed clustering algorithms.....	41
2.5	Analysis of ODMRP.....	42
2.5.1	Protocol overview.....	42
2.5.2	Route discovery method and multicast tree construction.....	45
2.5.3	Improvements on ODMRP.....	46
2.5.4	Why ODMRP is a reference protocol.....	49
2.6	Research Problem	50
2.6.1	Research problem definition.....	51
2.6.2	Existing efforts that attempt to address the research problem.....	52
2.6.3	Motivation for this research.....	54
2.7	Summary.....	54
3	Recognition of ELC-MANET Phases	57
3.1	Introduction.....	57
3.2	Requirements of the Solution.....	57
3.2.1	Efficiency.....	58
3.2.2	Robustness.....	59
3.2.3	Scalability.....	59
3.2.4	Simplicity.....	60
3.2.5	Adaptability.....	60
3.3	Detection of the ELC-MANET Topologies.....	61
3.3.1	Hypotheses and the formula of the proposed mechanism.....	61
3.3.2	Computation of stdev by an online algorithm.....	62
3.4	Summary.....	64
4	Characterisation of ELC-MANET Phases	66
4.1	Introduction.....	66
4.2	ELC-Matrix Topology.....	67
4.2.1	Characteristics of the ELC-matrix topology.....	67
4.2.2	Matrix Topology Route Discovery Method.....	68

4.2.3	Multicast tree construction method for MTRDM.....	70
4.3	ELC-Cluster-based Topology.....	71
4.3.1	Characteristics of the ELC-cluster-based topology.....	71
4.3.2	Cluster-based Topology Route Discovery Method.....	72
4.3.3	Proposed clustering algorithm for CTRDM.....	74
4.4	ELC-Random Topology.....	92
4.4.1	Characteristics of the ELC-random topology.....	92
4.4.2	Random Topology Route Discovery Method.....	93
4.5	Summary.....	94
5	Design Specification of TAPMRP	96
5.1	Introduction.....	96
5.2	Joining and Leaving a Multicast Group.....	98
5.3	Maintenance of Local Connectivity.....	99
5.4	Multicast Controller Node of TAPMRP.....	100
5.5	Route-discovery-method Switch of TAPMRP.....	104
5.5.1	Computation of the standard deviation of nodes' neighbours.....	105
5.5.2	Determining a switch to another route discovery method.....	106
5.5.3	Announcing a switch to another route discovery method.....	108
5.5.4	The controller hello packet and the data-reply packet.....	109
5.6	Design Specification of MTRDM.....	110
5.6.1	Join-Query and Join-Reply packet.....	110
5.6.2	Operation of MTRDM.....	112
5.6.3	Routing information maintained by MTRDM.....	115
5.7	Design Specification of CTRDM.....	115
5.8	Specification of RTRDM.....	118
5.9	Ensuring a Seamless Route Discovery Method Switch.....	118
5.10	Implementation of TAPMRP.....	120
5.11	Summary.....	124
6	Methodology	126
6.1	Introduction.....	126
6.2	Validation of the Research Problem.....	128
6.3	Detection of the ELC-MANET Phases.....	129

6.3.1	Formulation of the standard-deviation mechanism.....	129
6.3.2	Online algorithm for computing standard deviation.....	133
6.3.3	Justification for separating the ELC-MANET into phases.....	134
6.4	Matrix Topology Route Discovery Method.....	137
6.5	Cluster-based Topology Route Discovery Method.....	139
6.5.1	Introduction of secondary cluster-heads.....	139
6.5.2	Node redistribution parameter.....	140
6.5.3	Effect of the node mobility formula.....	141
6.5.4	Testing of CTRDM as a whole.....	142
6.6	Random Topology Route Discovery Method.....	143
6.7	Simulation Tests on TAPMRP.....	144
6.7.1	Convergence of TAPMRP after a controller node failure.....	145
6.7.2	Seamless route discovery method switch.....	146
6.7.3	Probability of making correct switch decisions.....	147
6.7.4	Benefit analysis of using TAPMRP.....	149
6.8	Summary.....	149
7	Results and Analysis	151
7.1	Introduction.....	151
7.2	Validation of the Research Problem.....	152
7.3	Detection of the ELC-MANET Phases.....	154
7.3.1	Formulation of the standard deviation mechanism.....	154
7.3.2	Online algorithm for computing standard deviation.....	158
7.3.3	Justification for separating the ELC-MANET into phases.....	159
7.4	Matrix Topology Route Discovery Method.....	163
7.5	Cluster-based Topology Route Discovery Method.....	167
7.5.1	Effect of introducing secondary cluster-heads.....	167
7.5.2	Effect of the node redistribution parameter.....	170
7.5.3	Effect of the node mobility formula.....	173
7.5.4	Performance of CTRDM as a whole.....	175
7.6	Random Topology Route Discovery Method.....	179
7.7	Topology-Aware Polymorphic Multicast Routing Protocol.....	181
7.7.1	Convergence of TAPMRP after a controller node failure.....	181
7.7.2	Seamless route discovery method switch.....	185

7.7.3	Probability of making correct switch decisions.....	186
7.7.4	Benefit analysis of using TAPMRP.....	190
7.8	Discussion on the Overall Results.....	194
7.9	Summary.....	197
8	Conclusions and Recommendations	200
8.1	Introduction.....	200
8.2	Summary and Conclusions.....	200
8.3	Major Contributions.....	204
8.4	Minor or Applied Contributions.....	207
8.5	Limitations of the Research.....	208
8.6	Recommendations for Future Work.....	208
Appendix	Pseudo code of TAPMRP's Functions	211
References		235

LIST OF FIGURES

1.1	ELC interactive scenario.....	2
1.2	MANET example.....	4
1.3	Example of (a) unicast (b) multicast and (c) broadcast.....	5
1.4	Example of broadcast redundancy in a MANET.....	7
1.5	ELC-matrix topology.....	10
1.6	ELC-cluster-based topology.....	11
1.7	ELC-random topology.....	12
2.1	Propagation of Join-Query and Join-Reply in ODMRP.....	43
3.1	Online algorithm for computing the standard deviation [145].....	63
4.1	Greedy set cover algorithm for $F(u, v)$ [7, 39].....	69
4.2	Example of a wireless ad hoc network.....	75
4.3	ELC-cluster-based network with 8 nodes.....	81
5.1	Propagation of the Data-Request and Data-Reply packet in TAPMRP.....	103
5.2	Main components of TAPMRP.....	105
5.3	State transitions in TAPMRP.....	107
5.4	Execution flow of the procedures in the route-discovery-method switch.....	108
5.5	Operation of MTRDM during each Join-Query interval.....	112
5.6	Operation of CTRDM during each Join-Query interval.....	117
5.7	Block diagram of TAPMRP's functions related to the neighbour hello message.....	121
5.8	Block diagram of TAPMRP's functions related to the controller hello message.....	122
5.9	Block diagram of functions of TAPMRP related to the Join-Query.....	123
6.1	Example: $(3 \times 9) + 1$ ELC-matrix topology.....	130
6.2	Example: ELC-cluster-based topology with 28 nodes.....	131
6.3	Example: ELC-cluster-based topology with 10 nodes.....	132
6.4	Example: ELC-random topology with 28 nodes.....	132

6.5	Approximate dimensions of clusters in the simulated network.....	136
7.1	Forward nodes in ODMRP with respect to the transmission range.....	152
7.2	Dependence of forward nodes on the network size in ODMRP.....	153
7.3	Variation of stdev with the transmission range.....	155
7.4	Demarcation lines for ELC-MANET topologies.....	156
7.5	ODMRP's performance in the three ELC-MANET topologies.....	160
7.6	Effect of ELC-MANET topology differences on ODMRP's performance.....	162
7.7	Number of forward nodes in MTRDM and the blind flooding method of ODMRP....	164
7.8	Performance comparison between MTRDM and the blind flooding method.....	165
7.9	Effect of node mobility on the performance of MTRDM.....	166
7.10	Number of forward nodes on CTRDM's clustering algorithms on secondary cluster-heads.....	167
7.11	Workload per forward node in CTRDM's clustering algorithms on secondary cluster-heads.....	169
7.12	Performance of CTRDM's clustering algorithms on secondary cluster-heads.....	170
7.13	Forward nodes in CTRDM's clustering algorithms on the node redistribution parameter.....	171
7.14	Workload per forward node for the two CTRDM's clustering algorithms on the node redistribution parameter.....	172
7.15	Performance of the two CTRDM's clustering algorithms on the node redistribution parameter.....	173
7.16	Performance of the two clustering algorithms of CTRDM on node mobility.....	174
7.17	Number of forward nodes in CTRDM and the blind flooding method.....	176
7.18	Performance of CTRDM and the blind flooding method.....	177
7.19	Forward nodes in CTRDM with and without mobility and in the blind flooding method with mobility.....	177
7.20	Performance of CTRDM with and without mobility and the blind flooding with mobility.....	179
7.21	Forward nodes in RTRDM and the blind flooding method.....	180
7.22	Performance in RTRDM and the blind flooding method.....	180
7.23	Convergence curve of new controller node announcements in TAPMRP.....	182
7.24	Variation of new controller election convergence with the transmission range.....	183
7.25	Variation of new controller election convergence with the network size.....	184

7.26 Performance of TAPMRP during election of a new controller node.....	185
7.27 Probability of detecting the ELC-matrix topology by TAPMRP.....	187
7.28 Probability of detecting the ELC-cluster-based topology by TAPMRP.....	188
7.29 Probability of detecting the ELC-random topology by TAPMRP.....	189
7.30 Number of transmissions of control packets in ODMRP and TAPMRP in a network of 55 nodes.....	193

University of Cape Town

LIST OF TABLES

2.1 Comparison of MANET multicast routing protocols.....	33
4.1 Settings of GloMoSim-2.03 for the transmission range of 377m.....	67
4.2 A sample of the degree of local connectivity values.....	84
6.1 Specifications of a laptop and a desktop computer.....	133
6.2 Data set for testing the standard deviation online algorithm.....	134
7.1 Standard deviation values obtained using different approaches.....	159
7.2 Performance of TAPMRP during changing route discovery methods.....	186

NOMENCLATURE

CBR	Constant Bit Rate
CDS	Connected Dominating Set
CTRDM	Cluster-based Topology Route Discovery Method
ELC	E-Learning Classroom
GPS	Global Positioning System
ID	Identification
MAC	Media Access Control
MANET	Mobile Ad hoc NETwork
MCDS	Minimum Connected Dominating Set
MTRDM	Matrix Topology Route Discovery Method
ODMRP	On Demand Multicast Routing Protocol
OSI	Open Systems Interconnection
RTRDM	Random Topology Route Discovery Method
TAPMRP	Topology-Aware Polymorphic Multicast Routing Protocol

CHAPTER 1

INTRODUCTION

1.1 Background

Today, almost any system of education involves E-Learning either directly or indirectly. In this thesis, E-Learning is understood as all forms of electronically supported learning and teaching, and therefore is not confined only to distance modes of learning. In particular, this research is interested in E-Learning as an integral part of hybrid learning which combines E-Learning and conventional face-to-face information delivery methods. Hence, in this case, students are located in a physical classroom and not in a ‘virtual’ classroom which is usually associated with E-Learning. The environment where students are in a physical classroom and use E-Learning methods to supplement traditional methods is hereinafter referred to as an E-Learning Classroom (ELC) environment.

This research proposes the use of Mobile Ad hoc NETworks (MANETs) to support E-Learning offered in ELC environments. The self-configuring and portable nature of MANET devices offers a suitable information and communication technology (ICT) solution to E-Learning mainly in terms of economic viability and setup time [1]. The portability of MANET devices affords students in the ELC environment movement with the devices, enabling group-work inside and outside the classrooms. MANETs are capable of delivering and supporting almost all the E-Learning services which include offline websites, video streaming, file sharing and e-mail. MANETs are also capable of facilitating interactive activities and group communication among students and their teacher in a classroom setup as illustrated in Figure 1.1. A MANET that is used to serve the ELC environment is hereinafter called the ELC-MANET.

Video streaming is envisaged to be one of the major ELC-MANET services. A teacher in the ELC environment may decide to deliver a lesson that includes a video show to his or her students. The video can be multicast (explained in section 1.2) from a MANET device of the teacher to MANET devices of the students by video streaming. Each student can then work independently and conveniently at his or her MANET device. If the lesson includes an

assignment at the end of the video then the student can use the MANET to confidentially unicast (explained in section 1.2) his or her solutions to the teacher's device. A video streaming service in the ELC-MANET environment can be offered through highly efficient multicasting without retransmissions because it can tolerate some packet losses which can easily occur in a MANET due to its wireless nature.

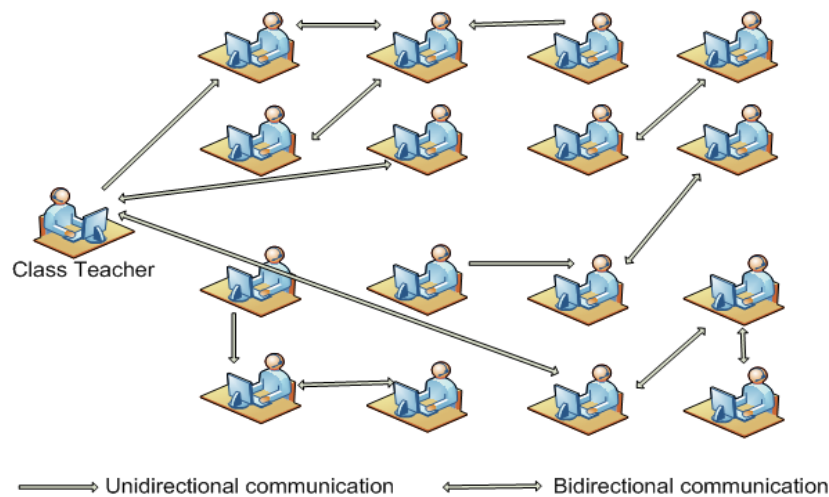


Figure 1.1 ELC interactive scenario

The ELC-MANET is capable of reliably offering services that are based on text data such as offline websites and file sharing. Such services can be offered through unicasting or multicasting. While transmitting reliable text-based data by unicast methods can easily be done by traditional protocols such as a combination of the file transfer protocol (FTP) and the transmission control protocol (TCP), delivering reliably the same type of data by multicasting in a MANET is an area which is still being researched [2]. Retransmission of lost packets and the use of error correcting codes are some of the proposed methods [2] for achieving reliable MANET multicasting. These proposed methods appear to be technically sound such that it is reasonable to expect that MANET multicasting will soon be able to reliably and efficiently deliver data that are highly sensitive to packet losses.

At the network layer, MANET services in the ELC environment can be delivered more efficiently in terms of resource usage by multicasting rather than unicasting [3]. Instead of sending multiple video streams to participants' devices, multicasting allows only one video stream to be sent and copied to multiple receivers simultaneously. To optimise gains from multicasting, coping (or branching) of the data stream is done as close to the receivers as

possible. Specific benefits of multicasting include support for distributed applications, optimised network performance and better scalability of information distribution. The proposal in this thesis is to use multicast routing in the ELC-MANET.

Many multicast routing protocols exist that are possible candidates to serve ELC environments. However, the observation in this research is that the protocols have excessive control overhead which makes them less efficient in ELC-MANET environments which are characterised by limited bandwidth and supply of battery energy used by the MANET devices.

The rest of this chapter is organised as follows. Section 1.2 presents an overview of MANETs followed by an introduction to the problem of this research in section 1.3. The problem statement and the research hypothesis are presented in section 1.4 and 1.5 respectively. The proposed solution to the research problem is presented in section 1.6. Section 1.7 covers the overall aims of the research work reported in this thesis. The chapter ends with the thesis road-map presented in section 1.8.

1.2 Overview of MANETs

A MANET is a self-organising and rapidly deployable network, which does not have a fixed infrastructure [4]. Nodes in MANETs cooperate to form the necessary infrastructure which makes multi-hop communication possible. MANETs are useful in many situations where deployment of fixed wire networks is hindered by factors such as terrain, time, and economic viability. Examples of such situations include military operations, emergency rescue operations especially where fixed infrastructure is destroyed by a natural disaster, conferences and meetings, electronic classrooms (related to ELCs considered in this research), multiplayer computer games, construction sites and vehicular communication.

Implementation of a MANET can be done by using most of the existing wireless technologies. For example, the common wireless Local Area Network technologies belonging to the IEEE802.11 series such as IEEE802.11-1997, IEEE802.11b, IEEE802.11a, and IEEE802.11g are possible candidates that can implement a MANET. Other examples include Bluetooth (IEEE802.15.1) and broadband wireless technologies such as IEEE802.16, IEEE802.16a, and IEEE802.16e [1]. This research is interested in wireless technologies that are for both indoor and outdoor use, and with transmission ranges which are less than 1km.

The simulation experiments of this research used the IEEE802.11-1997 which is assumed (in this research) to be the one implemented in a network simulator called GloMoSim-2.03 [5]. The use of the IEEE802.11-1997 in this research is considered to be adequate due to the nature of the research problem (covered in section 1.4) which is independent of the improved data rates which are herein understood as the main feature of the releases in the IEEE802.11 series that came after the 1997 release (i.e. IEEE802.11-1997).

An example of a MANET is shown in Figure 1.2. Multi-hop communication is one of the useful properties of a MANET. In Figure 1.2, for example, laptop 1 and laptop 6 are not directly connected but they can communicate through intermediate devices such as laptop 2 and laptop 4. This means that devices in a MANET act like routers for other devices (or nodes). Since any device can act as a router at any time it is important that devices in a MANET should be capable of creating and maintaining routing information such as a routing table. It is possible for one or more of the devices in a MANET to get connected to a large wired network such as the Internet. The rest of the nodes not directly connected to such an external network may access services available on the external network through the gateway nodes.

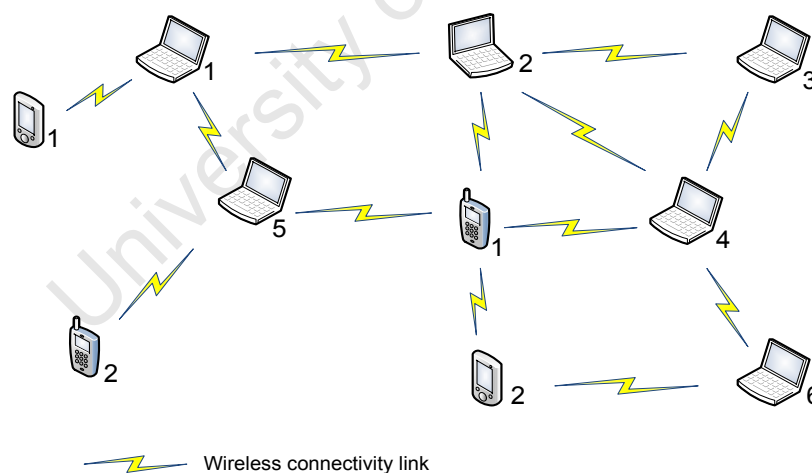


Figure 1.2 MANET example

Figure 1.2 also shows that a MANET may have more than one type of device. For example, in Figure 1.2 we have laptops, smart phones and personal digital assistants (PDAs). However, it is important that any two devices must use the same protocol for them to communicate or a translation of one protocol into another should be used in a situation of having devices with different communication protocols. In this research, the assumption is that all the devices are

homogeneous, implying that they all use the same OSI-based protocol stack for communication.

A MANET can transmit packets through unicasting, multicasting and broadcasting. Figure 1.3 illustrates the difference between the three mentioned methods of packet transmission in a MANET. Note that even though only a subset of nodes (i.e. receivers) is required to receive packets in a unicast or a multicast process, all the nodes that are directly connected to a source node detect the packets from the source node; but only the intended receivers receive and process the packets while the rest of the neighbours of the source node drop the packets upon detecting that they are not the intended receivers. In Figure 1.3 (a) and (b), for example, all the non-receivers can detect packets from the source node but they drop them.

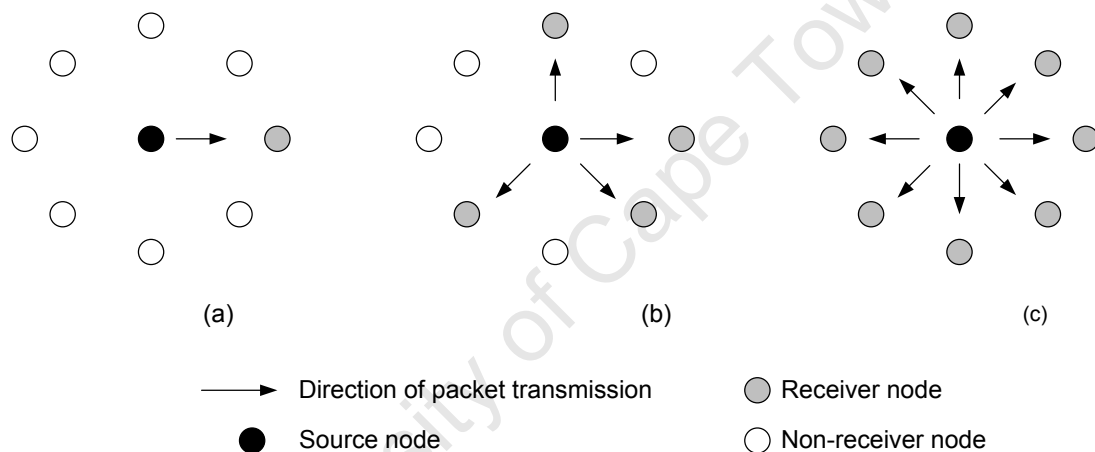


Figure 1.3 Example of (a) unicast, (b) multicast and (c) broadcast

Currently, wide deployment of MANETs is challenged by a number of factors. The main factors appear to be the limited battery energy which powers the MANET devices, limited bandwidth, and unpredictable topology changes due to free mobility of nodes [6]. Examples of MANET devices are laptop computers, notebook computers, PDAs, and smart phones. When these devices are used in a MANET they are powered by a battery. It seems currently battery technologies have not yet reached the stage of supplying enough energy that can power a relatively large MANET such as a laptop computer for a reasonably long period of time without recharging.

MANET devices use a shared wireless medium for transmission which has limited bandwidth capacity due to factors such as interference. This limitation makes it difficult for MANETs to

guarantee quality of service without rejecting service requests through admission control. In a highly dynamic network, connectivity links can easily be broken resulting in loss of packets during a transmission session. Hence, providing service reliability is one of the challenges faced by MANETs.

1.3 Cost of Route Discovery in MANET Multicasting

As mentioned in section 1.2 three of the major challenges which negatively affect operations of MANETs are the limitation of battery energy, limitation of bandwidth, and rapid topology changes. Limitation of the battery energy seems to be the most serious challenge faced by MANETs. Since research in battery technologies does not appear to be making rapid advances, it has become necessary to find ways of using the available battery energy as efficiently as possible. Research reports on MANETs have pointed out that wireless transmissions are the greatest contributor to the consumption of energy in MANETs [7]. Hence, reduction of wireless transmissions is currently one of the important research areas for efficient use of energy in MANETs. (Note that the term “cost” in this thesis refers to the utilisation of network resources such as energy for signal transmission, bandwidth, storage space, and computational time.)

Decisions made by a routing protocol can directly affect the level of transmissions in a MANET. For example, a broadcast protocol which forwards a broadcast packet at every node in a network solely based on whether the packet is a duplicate or not [8, 9] is more likely to result in more redundant retransmissions in a network than a broadcast protocol which forwards a broadcast packet by also considering whether its neighbours have already received the packet or not. Therefore, the design of a routing protocol with respect to how it selects forward nodes requires serious consideration. Although the problem of the forward node selection is largely associated with broadcast protocols, unicast and multicast routing are also affected by the problem as they both use broadcast protocols for route discovery and maintenance. Hence, every unicast and multicast protocol is required to select its broadcast algorithm for route discovery with proper selection of forward nodes in mind.

A review conducted during this research on the current multicast routing protocols reveals that most of the protocols are not efficient enough with respect to usage of energy. Their inefficiency is mainly due to the use of poorly designed broadcast methods for route

discovery. In particular, the inefficiency problem in such broadcast methods comes from broadcast redundancy. In this research, the broadcast redundancy problem is considered to be a major component of the cost of route discovery and maintenance in a MANET multicast protocol. Hence, the terms “broadcast redundancy” and “cost of route discovery and maintenance” are used interchangeably in this thesis.

Figure 1.4 illustrates the problem of broadcast redundancy. Node 1 broadcasts a packet in a network consisting of nine nodes using a blind flooding method [9] in which every node in a network rebroadcasts the packet when it receives it for the first time. Node 3, 4, and 5 act as forward nodes for the broadcast packet from node 1. Clearly, a rebroadcast of the packet performed by node 4 is redundant since node 7 and 8 are covered by a retransmission from node 3 and 5 respectively. In a network of multiple sources of traffic, broadcast redundancy can easily lead to a more serious problem known as the broadcast storm problem [8].

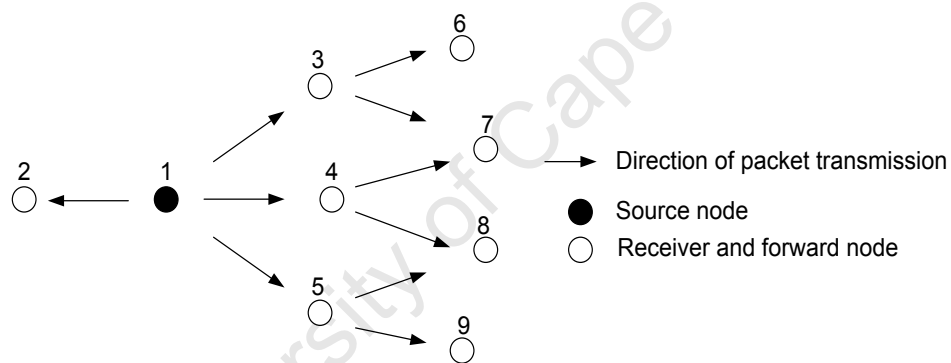


Figure 1.4 Example of broadcast redundancy in a MANET

Broadcast redundancy has some major negative effects in a MANET. First, the unnecessary packet retransmissions lead to a faster depletion of battery energy. Another effect is that the redundant packet retransmissions cause unnecessary contention for the limited bandwidth and collisions in a MANET resulting in reduced throughput [8].

An example of a multicast routing protocol which uses blind flooding for route discovery and maintenance is On Demand Multicast Routing Protocol (ODMRP) [9]. ODMRP is one of the well-known multicasting routing protocols among MANET researchers. Many researchers of MANET multicasting use ODMRP in their research efforts for performance comparison purposes. In spite of having attractive features such as scalability and robustness, ODMRP

uses blind flooding for its route discovery which generates a lot of broadcast redundancy. Multicast Ad hoc On-demand Distance Vector (MAODV) [10] protocol is also one of the multicast routing protocols cited in most of the reported research work on MANET multicasting. MAODV also uses blind flooding for route discovery. The literature review indicates that highly efficient broadcasting techniques have recently been developed but most of the current multicast routing protocols have not exploited such techniques for route discovery.

The cost of route discovery and maintenance is not the only challenge faced by multicast routing but it is the only one considered in this research. There are many other problems affecting multicast routing which mainly arise from the wireless nature of the transmission medium used by MANETs. These include node mobility and topology changes, support for quality of service, scalability, and security. Some efforts have been made in this research to address the other challenges which are related to the cost of route discovery.

1.4 Problem Statement

The proposed application of MANETs in the ELC environment is challenged by broadcast redundancy which exists in current multicast routing protocols. Since the ELC-MANET is likely to offer multimedia services on a relatively large scale, the demand for energy to support ELC services is huge. Considering that MANET devices are powered by a limited supply of battery energy, one possible way of addressing such a huge demand for energy is by using multicasting routing protocols which are as efficient as possible in terms of energy usage. While the broadcast redundancy problem existing in the current multicast routing protocols may be acceptable in some applications, it is considered in this research to be significant for ELC services due to the huge demand for energy in such environments. Hence, the cost of route discovery and maintenance, which mainly comes from the broadcast redundancy, in the current multicast routing protocols is the problem of this research undertaking.

By considering the problem of broadcast redundancy associated with the route discovery operation of a multicast routing protocol, it implies that the scope of this research is limited to the signaling component (or the part that deals with control messages) of multicasting in the ELC-MANET. In addition, the nature of the research problem also implies that the research is confined to the network layer of the OSI reference model.

1.5 Research Hypothesis

The method that is used to address the research problem is based on the overall hypothesis that given a MANET of any size, there exists a subset of nodes of the MANET such that a complete broadcast of a packet to the entire MANET can be achieved by a transmission performed by a source node of the packet and retransmissions of the packet performed by the nodes that belong to the subset. Such a subset of nodes is known as a Connected Dominating Set (CDS) [11, 12] (or the backbone) of a MANET. Finding the minimum CDS (MCDS) of any given MANET is an NP-complete problem [11, 12]. Hence, this research resorts to using approximations of the MCDS determined through heuristic approaches.

The goal of the research is therefore to develop a multicast routing protocol that is capable of using a CDS which is as close as possible to the MCDS of the ELC-MANET in which the protocol operates. The heuristic approaches to the MCDS problem of this research make use of the observation that the MCDS depends on the network connectivity characteristics. In particular, the following postulates are considered in the approaches:

- 1) Since the topology (or arrangement) of nodes determines the network connectivity characteristics of a MANET, it is likely that the MCDS also depends on the topology of the nodes.
- 2) When network connectivity of a particular MANET increases (i.e. more nodes becoming within the reach of one another's transmission range) its MCDS is reduced which implies that fewer nodes are required to relay a broadcast packet for a complete broadcast, and the converse is also true.
- 3) In scenarios of high node mobility, the MCDS of a particular MANET changes frequently in accordance with the mobility pattern or behaviour of the nodes.

1.6 Using an Adaptive Multicast Routing Protocol

The three postulates on the MCDS problem presented at the end of section 1.5 lead to the need to exploit the characteristics of the MANET topologies that arise in the ELC-MANET environment. The observation in this research is that when a MANET is used in the ELC environment its topology dynamically transforms itself from one distinct form into another based on the activities performed by the students. It is further observed that there are three distinct topologies associated with the dynamic transformation of the ELC-MANET. The

three distinct topologies of the ELC-MANET are covered in subsection 1.6.1 followed by the proposed adaptive multicast routing protocol in subsection 1.6.2.

1.6.1 Topologies of the ELC-MANET

To proceed with the explanation on the ELC-MANET transformation, we make two assumptions. The first assumption is that each student participating in an ELC activity uses only one MANET device. Secondly, we assume that all the MANET devices under consideration are able to communicate with one another. A time comes in the ELC situation when students assemble in one big room and sit in rows as illustrated by the devices shown in Figure 1.5. Note that in Figure 1.5 we assume that each device shown represents both a student and a MANET device. Usually, when students are attending a lesson in a classroom setup as depicted in Figure 1.5 there is occasional mobility of few students during the time of the lesson. Mobility in this case may involve changing of seats, walking out of the classroom leaving a seat vacant and entering the classroom to occupy a vacant seat. The arrangement of MANET-enabled devices in this case is hereinafter referred to as the ELC-matrix topology.

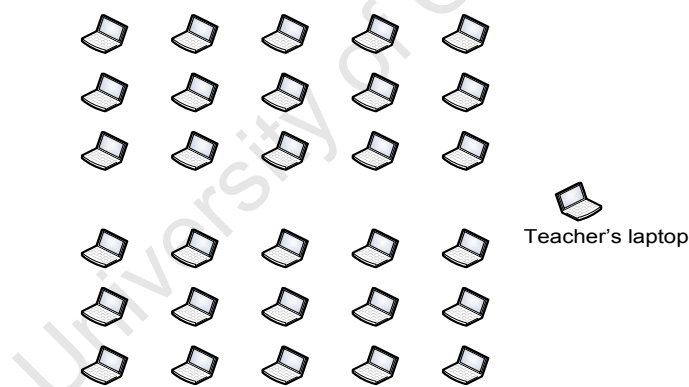


Figure 1.5 ELC-matrix topology

Next, the students may be asked to rearrange themselves into small discussion groups or clusters. Discussion groups can be located inside the classroom or outside or some groups inside and others outside. At this point the students are considered to have formed a cluster-based topology which is hereinafter called the ELC-cluster-based topology. Figure 1.6 shows an example of the ELC-cluster-based topology. There is usually occasional mobility of students once they are in discussion groups. Mobility may take the form of changing seats within a cluster, changing clusters, and leaving and joining a cluster. Note that the term

“classroom” that appears in the term “ELC-cluster-based topology” does not strictly imply that the ELC-cluster-based topology is confined to a classroom situation, but simply implies that the topology is associated with activities that normally take place in a classroom environment.

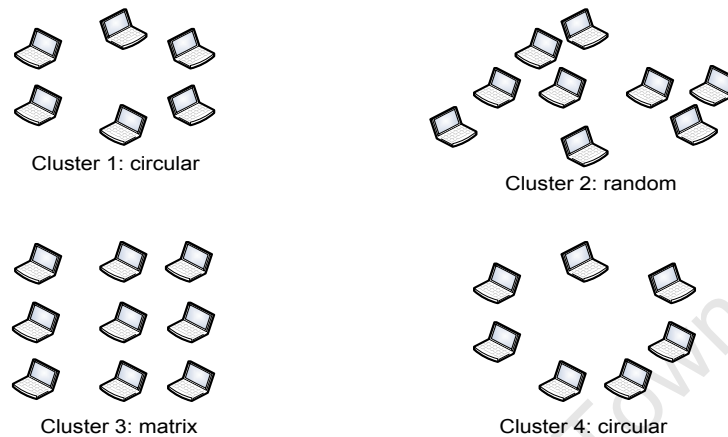


Figure 1.6 ELC-cluster-based topology

After holding discussions in small groups, it is assumed that the students disperse to various locations. This is the time when the students are taking a break. Some students could be taking meals in a cafeteria, while others could be resting in their rooms. It is still assumed that the students can communicate using their MANET-enabled devices. At this stage, the students have formed a topology hereinafter called the ELC-random topology. An example of the ELC-random topology is shown in Figure 1.7. This time mobility is relatively higher than in the other two network scenarios, namely matrix and cluster-based topologies, since there are likely to be more students on the move. Similarly, note that the term “classroom” that appears in the term “ELC-random topology” simply means that the topology is associated with activities that usually take place in a classroom situation. In fact, the entire ELC-random topology usually exists outside the classroom environment.

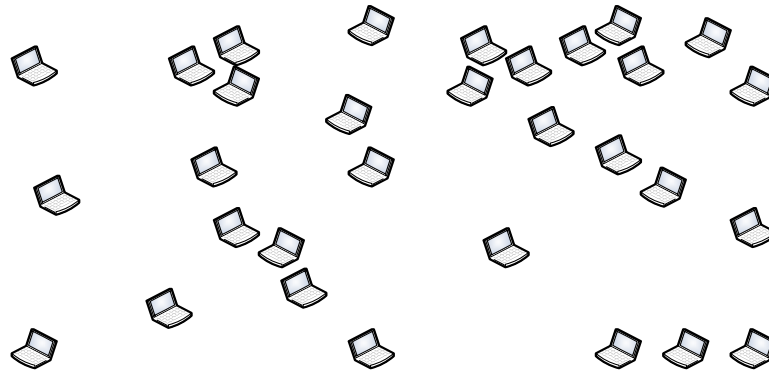


Figure 1.7 ELC-random topology

We now conclude that the ELC-MANET can transform itself into the ELC-matrix, ELC-cluster-based, or ELC-random topology depending on the nature of the students' activities. These three ELC-MANET topologies are also referred to in this thesis as phases of the ELC-MANET. Therefore, the ELC-MANET is modelled as a network which undergoes a transition from one phase to another. The model assumes that the transitions do not have a specific order. This implies that when the ELC-MANET is in a particular phase, it can transition to any of the other two topologies or phases.

After splitting the ELC-MANET into phases, the research proceeded by examining the connectivity characteristics of each of the three phases. The research observed that each of the three ELC-MANET topologies has its own characteristics which are different from those of the other two topologies. In general, the ELC-matrix topology has higher connectivity than any of the other two topologies since it is confined to a small area (i.e. a classroom). The clusters in the ELC-cluster-based topology are areas of high connectivity which result in the ELC-cluster-based topology being characterised by small areas of high connectivity scattered over a relatively large area. The ELC-random topology has no definite connectivity pattern, and this is the connectivity characteristic of the topology. Another characteristic of the ELC-random topology is the likely presence of higher mobility of nodes than in the other two ELC-MANET topologies.

Since the nature of connectivity in a MANET has a direct effect on the performance of a routing protocol, it becomes necessary to develop an adaptive multicast routing protocol capable of dealing with the connectivity characteristics of each of the three phases of the

ELC-MANET. The proposed adaptive multicast routing protocol is presented in the following subsection.

1.6.2 Topology-Aware Polymorphic Multicast Routing Protocol

The problem of this research, which is on the cost of route discovery in a multicast routing protocol operating in the ELC-MANET environment, is addressed through the development of an adaptive multicast routing protocol called a Topology-Aware Polymorphic Multicast Routing Protocol (TAPMRP). TAPMRP is developed in this research as an improvement on ODMRP [9], and consists of three route discovery methods, namely a Matrix Topology Route Discovery Method (MTRDM), a Cluster-based Topology Route Discovery Method (CTRDM) and a Random Topology Route Discovery Method (RTRDM). The term “polymorphic” in TAPMRP implies that TAPMRP is capable of changing from one “form” to another. The “forms” of TAPMRP are its three route discovery methods. Hence, TAPMRP operates by switching from one route discovery method to another depending on the type of the prevailing network topology at any instant. MTRDM is intended to operate when the prevailing topology of the ELC-MANET is the ELC-matrix topology. During the phase of the ELC-cluster-based topology, TAPMRP switches to CTRDM. RTRDM is designed to operate during the time of the ELC-random topology.

To address the problem of broadcast redundancy, MTRDM is proposed to select forward nodes of broadcast packets (i.e. route request packets) based on a pruning technique. According to the technique, a node sending a broadcast packet selects only a subset of its one-hop neighbours to act as forward nodes for the packet. The sender node utilises neighbourhood information to select forward nodes whose retransmission of the packet does not result in broadcast redundancy (refer to Figure 1.4). In CTRDM, a cluster-based broadcasting technique is used in which only cluster-heads and a subset of gateway nodes are allowed to forward a broadcast packet. Broadcasting of a route request packet in RTRDM uses a clustering technique which creates clusters on demand as data transmission is in progress. Only cluster-heads and gateway nodes of the resulting clusters are allowed to forward a broadcast packet.

The use of the three route discovery methods in TAPMRP is able to optimise both efficiency by reducing forward nodes, and reliability by minimising packet losses. Optimising both efficiency and reliability is unlikely to be achieved if only one route discovery method is used

for all the three phases of the ELC-MANET. If only MTRDM is used, it is likely to perform efficiently and reliably in the ELC-matrix topology but unreliably in the high mobility environment of the ELC-random topology due to its lack of advanced capabilities for adapting to mobility of nodes. Using only CTRDM is likely to optimise both efficiency and reliability in the ELC-cluster-based topology but is likely to be inefficient in the high connectivity environment of the ELC-matrix topology due to the unnecessary creation of clusters. Finally, if RTRDM is the only route discovery method, it is likely to optimise both efficiency and reliability in the ELC-random topology, but is likely to be inefficient in the ELC-matrix topology due to the unnecessary creation of clusters and the less effectiveness of the passive clustering technique compared to the pruning technique used by MTRDM. Compared to ODMRP, all the three route discovery methods of TAPMRP are more efficient than the blind flooding method of ODMRP.

TAPMRP uses a procedure called a route-discovery-method switch for making switching decisions from one route discovery method to another. The route-discovery-method switch resides in a controller node of a multicast group. Every multicast group has only one controller node which coordinates network-wide operations of TAPMRP. The route-discovery-method switch has three sub-procedures each one of which has a specific task to perform. The first sub-procedure is responsible for collecting data on the number of neighbours of each member node of a multicast group and uses the data to compute the standard deviation of the number of neighbours of nodes in a network. The second sub-procedure uses the standard deviation information to make a switching decision from one route discovery method to another. After a decision has been made to switch to a particular route discovery method, a separate sub-procedure is used to announce the decision to all member nodes of a multicast group.

All the ideas which constitute the solution to the research problem were tested by computer simulations which were conducted in GloMoSim-2.03 [5]. GloMoSim-2.03 is specifically designed for simulations of wireless networks. The ideas were tested by incorporating them in ODMRP [9] which is already implemented in GloMoSim-2.03.

1.7 Overall Aims of the Research

The purpose of this research undertaking is to develop a multicast routing protocol which is capable of making efficient use of the limited battery energy and limited bandwidth of the

ELC-MANET during a route discovery operation. In order to accomplish the research purpose, the major aims of the research are as follows:

- 1) Determining a computationally efficient mechanism for detecting dynamically the three phases of the ELC-MANET.
- 2) Developing a route discovery method for each of the three phases of the ELC-MANET that minimises the cost of route discovery and maintenance.
- 3) Integrating the route discovery methods obtained in (2) in one multicast routing protocol and use the mechanism obtained in (1) to dynamically switch from one route discovery method to another depending on the prevailing phase (or topology) of the ELC-MANET.

1.8 Thesis Road-Map

The rest of the thesis is organised as follows. Literature review and formulation of the research problem are presented in chapter 2. Chapter 3 covers the proposed mechanism for detecting dynamically the phases of the ELC-MANET. An in-depth analysis of the three phases of the ELC-MANET and the proposed route discovery methods for the three ELC-MANET phases are covered in chapter 4. A design specification of TAPMRP is presented in chapter 5 followed by the research methodology in chapter 6. Chapter 7 covers research results and analysis. Conclusions and recommendations are covered in chapter 8, which is the last chapter of the thesis.

CHAPTER 2

PROBLEM FORMULATION

2.1 Introduction

A literature review done in this research has revealed some work already done by researchers on the use of mobile ICT devices in E-Learning. C. Chang and J. Sheu [13] designed an ad hoc classroom system consisting of an e-blackboard subsystem, a voice and image transmission subsystem, a PowerPoint broadcasting subsystem, and a text communication subsystem. C. Chang and J. Sheu also defined an eSchoolbag system containing materials such as electronic books, notebooks, a parent's contact book, a pencil case, writing materials, sheets, a calculator, and an address book. Later, C. Chang et al. [14] associated the ad hoc classroom concept with four mobile learning scenarios: mobile indoor individual learning, mobile outdoor individual learning, mobile indoor group learning, and mobile outdoor group learning.

While C. Chang et al. focused their work on the application layer of MANETs in the context of a mobile classroom, S. Vijayaragavan et al. [15] worked at the network layer by investigating the performance of two multicast routing protocols, namely MAODV [10] and ODMRP [9]. The results by S. Vijayaragavan et al. [15] showed that for many scenarios in the virtual classroom (or ad hoc classroom) environment, MAODV achieved a higher packet delivery ratio than ODMRP. However, when it came to scenarios of rapid topology changes, ODMRP performed better than MAODV. In fact, what these authors found is in line with the design goals of the two protocols.

This research is similar to the work done by S. Vijayaragavan et al. [15] in that both research efforts investigate the use of MANETs in E-Learning from the viewpoint of multicast routing. However, this research takes a further step by aiming at addressing the high cost of route discovery and maintenance associated with multicast routing protocols. Hence, the literature review of this research also looked at the multicast routing protocols. Broadcasting methods and clustering techniques used by MANETs were also reviewed since they are used by multicast routing protocols for route discovery and maintenance.

The rest of this chapter is organised in the following way. A review of the current multicast routing protocols is covered in section 2.2. The current broadcasting methods are covered in section 2.3 followed by the existing MANET clustering methods in section 2.4. Section 2.5 presents an analysis of ODMRP, which is used in this research as a reference protocol for purposes of performance comparison and also as a basis for the proposed solution to the research problem. The research problem is presented in section 2.6. This chapter concludes with a summary in section 2.7.

2.2 Review of MANET Multicast Routing Protocols

The reasoning in this thesis is that proper understanding of multicast routing begins with having knowledge of how unicast routing operates. In some cases, a multicast protocol such as MAODV uses a unicast protocol for route discovery. Hence, an overview of unicast routing protocols is presented in the next two paragraphs. Thereafter, multicast routing protocols are covered.

2.2.1 Unicast routing protocols

The literature contains a large number of unicast routing protocols for MANETs. Royer et al. [16] classifies MANET unicast routing protocols into table-driven and source-initiated on-demand protocols. An example of a table-driven unicast protocol is Dynamic Destination-Sequenced Distance-Vector (DSDV) protocol [17], while Ad hoc On-demand Distance Vector (AODV) protocol [18] is an example of an on-demand unicast protocol.

Each node in a network using a table-driven protocol such as DSDV is required to maintain one or more tables that store routing information, and any change in the network structure necessitates that all the nodes in the entire network should be updated on the change. In contrast, on-demand routing like the one used by AODV does not require that all the nodes in the network should maintain routing information during the entire lifetime of the network setup. Routes in an on-demand routing technique are created only when wanted by a source node. When a node needs to transmit data, a route discovery process is initiated and completed once a route is found. An established route is maintained only during the time it is needed.

2.2.2 Multicast routing protocols

Recently, researchers have realised that most applications in MANETs involve group communication which can be served more efficiently by multicast routing rather than unicast routing [19]. A major benefit of using a multicast method is that it saves bandwidth and network resources by transmitting the same data packet to multiple receivers simultaneously [3]. Hence, multicast routing offers a suitable routing method to group-oriented applications characterised by collaborative efforts in achieving a particular goal. E-Learning which is being considered in this research is a good example of a group-oriented application which can benefit from multicast routing. Even though multicast routing is undoubtedly a preferred method of MANET routing by many researchers for group-oriented applications, it faces a myriad of challenges which mainly arise from the characteristics of the wireless medium which is used for packet transmission [19, 20]. Unicast and broadcast routing of MANETs also face the same challenges. The challenges include unpredictable and rapid topology changes, limited bandwidth, limited battery energy which power MANET devices, support for quality of service, scalability, and security.

The next subsection covers the challenges of multicast routing in detail. The different types of multicast routing protocols which address the multicast challenges are covered in the subsection which follows the challenges.

Challenges of MANET multicasting

Node mobility and topology changes: Nodes in a MANET are free to move anytime, anywhere and at any speed. In a highly dynamic network, data packets may be dropped during a multicast session resulting in a low packet delivery ratio. Hence, a multicast routing protocol must be robust enough to cope with node mobility and topology changes.

Limited bandwidth: The presence of noise and interference in a wireless transmission medium of MANETs limits the capacity of bandwidth. Therefore, a multicast routing protocol is required to utilise the available bandwidth as efficiently as possible by, among other factors, ensuring that many more data packets are transmitted than control packets.

Limited battery energy: MANETs consist of portable devices which have limited battery life. As a way of prolonging the battery life, a multicast routing protocol should minimise usage of power by reducing the number of packet transmissions performed by a node. One

possible way to minimise transmissions is by reducing broadcast redundancy associated with a route discovery process. Hence, a multicast routing protocol should strive to minimise the forwarding node set for a broadcast packet.

Quality of Service (QoS): QoS is defined as a set of service requirements which a network must meet as it transports data from the source to the receiver [21]. The need to deliver multimedia traffic in MANETs requires that multicast routing protocols should have the capabilities for QoS provisioning. Basically, QoS multicast routing has two goals: the first goal is to meet the QoS requirements of each admitted connection, and the second one is to optimise network resource usage.

Scalability: A multicast routing protocol is required to provide an acceptable level of service to packets even when a network is large. A major design issue to be considered under scalability is to minimise the amount of control overhead as the size of the network increases.

Security: Existence of a wireless medium, the broadcast nature of transmissions, and lack of a centralised administration makes security an important issue in MANETs. A MANET like any other wireless network is vulnerable to eavesdropping, interference, spoofing and many other security threats. The design of a multicast routing protocol needs to consider security issues especially in military operations.

The aforementioned multicast challenges have given rise to a number of multicast routing protocols each aimed at addressing one or more of the multicast challenges. The literature presents various ways of classifying multicast protocols. Common classes of multicast protocols are tree-based and mesh-based protocols. The next two subsections cover tree-based and mesh-based multicast protocols. Hybrid and adaptive multicast routing protocols are presented in a subsection following that of mesh-based multicast protocols.

Tree-based multicast routing protocols

The simplest and perhaps also the most reliable way of transmitting data to a subset of nodes in a MANET is by flooding in which every node of the network receives the data regardless of whether it is the intended receiver or not. Obviously, flooding does not utilise efficiently the MANET's limited bandwidth and the battery energy which powers the devices. Compared to flooding and a mesh-based routing technique, a tree-based routing method uses network resources in an efficient way. Unlike in flooding where every node of a network is

allowed to forward a data packet, a tree-based routing protocol allows only nodes that are part of a flood tree to forward packets to intended receivers.

When a tree-based routing approach is used in MANET multicasting, a multicast group of nodes is created in the form of a tree network structure consisting of a root node, intermediate nodes and leaf nodes. Only one path exists between any source-destination pair of nodes in a tree-based multicast. A tree-based multicast can further be divided into a per-source tree multicast and a shared tree multicast [22]. In a per-source tree multicast, used by on-demand or reactive protocols, each source of a multicast group establishes and maintains its own tree. The advantage of the per-source tree multicast is that the source uses the most efficient path to reach each and every multicast group member, while its major disadvantage lies in excessive control overhead used to maintain many trees. A shared tree multicast, used by proactive protocols, maintains a single tree for a multicast group. The main benefit of sharing a single multicast tree is in having low control overhead. The drawbacks with this technique include the path not necessarily being optimal and overloading of the root node.

In general, tree-based multicast protocols offer low resilience against topology changes. Breakage of a connectivity link in a tree-based multicast protocol triggers a tree reconfiguration process. When node mobility is high, the occurrence of link breakages is also usually high and this results in frequent tree reconfigurations, which in turn lead to excessive control overhead and low throughput.

A review of the various types of multicast routing protocols such as tree-based multicast routing protocols conducted in this research considered seriously the issue of broadcast redundancy associated with route discovery and maintenance. Broadcast redundancy is a significant problem in MANETs because it wastes battery energy and causes congestion and contention in a network [8]. Hence, any design effort on a multicast routing protocol aimed at addressing the challenges of MANETs' limited battery energy and bandwidth is expected to seriously consider broadcast redundancy. Examples of tree-based multicast routing protocols are presented in the following subsections.

Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS): AMRIS [23] is an example of a tree-based multicast protocol. The protocol establishes a shared multicast tree to support multiple senders and receivers in a multicast session. Each node in a multicast session is dynamically assigned a session ID number. A multicast delivery tree is

constructed rooted at a particular node with the smallest session ID number called the Sid. Usually the Sid is a source node if there is only one source node for a group. In the case of multiple source nodes, the source node with the smallest session ID becomes the Sid. When a link between two nodes breaks, the node with a larger session ID is responsible for a rejoining process. AMRIS repairs broken links by performing a local route repair without the intervention of a central controlling node thereby minimising control overhead. The introduction of session ID numbers prevents the formation of routing loops. Every node sends out a beaconing message which results in excessive control overhead. AMRIS has not explicitly dealt with broadcast redundancy associated with a route discovery process; it is therefore likely that its control overhead has a significant component of broadcast redundancy.

Multicast Ad hoc On-demand Distance Vector (MAODV) protocol: Another example of a tree-based multicast routing protocol is MAODV [10] which extends AODV [18]. MAODV uses a bidirectional shared multicast tree. Maintenance of the shared multicast tree is divided into three operations: selecting and adding a new link to the tree when a new node joins a multicast group, pruning the tree when a node decides to leave the group, and repairing a broken link. Member nodes of a multicast group maintain up-to-date membership information through receiving periodic group hello messages from a multicast group leader. The route discovery mechanisms used by MAODV is that of AODV which consists of a route request phase and a route reply phase. A route request (RREQ) packet is broadcast in a network by using a method which allows every node in the network to broadcast the RREQ packet as long as the packet is received for the first time and its time to live is non-zero. This broadcasting method, known as blind flooding, generates excessive broadcast redundancy which contributes significantly to the cost of route discovery in MAODV. Another major drawback of MAODV is that it yields a low packet delivery ratio in situations with high node mobility.

Associativity-Based Ad hoc Multicast (ABAM) routing protocol: ABAM [24] is also another example of a tree-based multicast routing protocol. It is an on-demand source-based protocol which establishes a multicast tree for each multicast session rooted at a multicast source. The construction of a multicast tree is primarily based on association stability which helps the source to select routes to receivers which are likely to last longer and need less reconfiguration. To initiate a multicast session, a multicast sender broadcasts a Multicast

Broadcast Query (MBQ) message throughout a network. On receiving the MBQ packet, a node appends to the packet its address and other information such as route relaying load, associativity ticks (i.e. number of beacons from neighbours), signal strength, and power life before the packet is rebroadcast. Hence, each MBQ packet collects information about the visited path as it is forwarded. A multicast receiver responds to an MBQ message by preparing and sending an MBQ-Reply message back to the source node. The source node uses MBQ-Reply messages to compute a stable multicast tree for delivery of data packets.

It is reported in [20] that ABAM generates less control overhead traffic and achieves a higher packet delivery ratio in comparison with ODMRP due to the stability of the paths between the sender and the receivers. One of the disadvantages of ABAM is latency since in some cases long paths are used to deliver data packets. Another problem with ABAM is that it begins to perform poorly as the size of the network increases. ABAM has not explicitly dealt with the efficiency of broadcasting the MBQ packets. To make the situation worse, ABAM encourages forwarding of duplicate MBQ packets for the purpose of increasing the chances of identifying more stable routes. Hence, the contribution of the broadcast redundancy to the cost of route discovery and maintenance in ABAM is likely to be very big.

Preferred Link-Based Multicast (PLBM) protocol: PLBM [25] is one of the tree-based multicast routing protocols which has considered efficient usage of network resources (i.e. bandwidth and energy) from the viewpoint of the cost of route discovery. It is a tree-based receiver-initiated multicast routing protocol. Each member node is required to initiate a connection to a multicast source. Each node in a multicast group maintains two tables: a Neighbours Neighbours Table (NNT) for local 2-hop topology information and a Connect Table (CT) for multicast tree information. When a new member wants to join a multicast group, it first checks its NNT to determine if the table has at least one multicast tree node (a connected member node, a forward node, or a source node). If at least one tree node is available, the new member node sends a Join-Confirm message to one of the tree nodes without flooding the network with Join-Query packets; otherwise, it computes a preferred list of its neighbours eligible to forward a Join-Query, piggybacks the computed preferred list in a Join-Query, and finally broadcasts the Join-Query message. On the receiving the Join-Query packet, each node checks if it is eligible to forward the packet or not. Selection of forward nodes for the Join-Query is based on a pruning technique which disqualifies nodes whose retransmission of the packet is likely to result in broadcast redundancy.

PLBM significantly reduces broadcast redundancy during a route discovery process. However, the protocol has two major drawbacks. First, every node sends a beacon message periodically which introduces considerable control overhead. Second, being a tree-based protocol and combined with the fact that it uses a hard state approach for maintaining information of a multicast tree, PLBM is likely to perform poorly in situations of frequent topology changes.

Additional examples of tree-based multicast routing protocols: Other examples of tree-based multicast routing protocols include Shared-Tree Ad hoc Multicast Protocol (STAMP) [26], Weight-based Clustering Multicast Protocol (WCMP) [27], Tree-based Multicast Protocol Using Multi-Point Relays (TMP-MPRs) [28], Bandwidth-Efficient Multicast Routing Protocol (BEMRP) [29], Spiral-fat-tree-based On-demand Multicast (SOM) protocol [30], Shared-Tree-based multi-source Multicast Routing Protocol (STM RP) [31], and Independent-Tree Ad hoc Multicast Routing (ITAMAR) protocol [32]. STAMP and WCMP construct their multicast trees with the help of a cluster-based organisation of nodes mainly for purposes of scalability. Both STAMP and WCMP have not explicitly considered broadcast redundancy associated with a route discovery process. TMP-MPRs has considered broadcast redundancy associated with any broadcast operation of the protocol by using only a subset of neighbours, called multi-point relay nodes, to forward a broadcast packet. Performance of TMP-MPRs in situations of high mobility is likely to be poor since it has not seriously considered mobility of nodes. A review of BEMRP, SOM, STM RP and ITAMAR shows that all these protocols have not explicitly considered broadcast redundancy associated with a route discovery operation.

Mesh-based multicast routing protocols

A mesh-based multicast can have multiple paths between any sender-destination pair of nodes in a network. The existence of multiple paths provides better resilience against topology changes compared to a tree-based multicast. However, the availability of extra paths in a mesh-based multicast requires more network resources, computational capacity and control overhead to maintain the paths. A mesh-based protocol, like a tree-based one, can be established in a proactive mode as well as in a reactive or on-demand mode.

Core-Assisted Mesh Protocol (CAMP): CAMP [33] is an example of a mesh-based multicast routing protocol. It is a proactive multicast routing protocol which constructs a

shared mesh for each multicast group. As a way of eliminating the need for flooding during a join operation, CAMP defines one or multiple cores for each multicast group. CAMP uses a receiver-initiated approach for receivers to join a multicast group. A node wishing to join a multicast group first determines if it has a neighbour which is already a member. If it has such a neighbour, the node announces its membership via a CAMP update process. Otherwise, the node either propagates a join request towards one of the cores or attempts to reach a node in the mesh by broadcasting a join request. CAMP needs an underlying proactive unicast routing protocol to maintain routing information about the cores.

According to the analysis carried out in [3], CAMP has good control traffic scalability for increasing multicast group size due to elimination of flooding of join requests. The main drawback of CAMP is its reliance on a unicast routing protocol which implies that any degradation in the performance of the unicast protocol during high mobility has a direct effect on the performance of CAMP. CAMP has not explicitly considered broadcast redundancy associated with a route discovery process since route discovery operations are performed by its underlying unicast routing protocol.

Forwarding Group Multicast Protocol (FGMP): FGMP [34] creates a multicast mesh on-demand and is based on the forwarding group concept. A forwarding group is a subset of nodes in a network which is eligible to forward packets belonging to a multicast group. Hence, each multicast group has a forwarding group. A forwarding group is periodically refreshed to address membership changes due to factors such as topology changes. There are two ways to advertise membership in FGMP: a receiver advertising (FGMP-RA) approach and a sender advertising (FGMP-SA) approach. In FGMP-RA, each receiver periodically floods its member information through a join request. When a sender receives the join request from a receiver member it updates its member table. A sender transmits multicast data packets only when the member table is not empty. In the FGMP-SA approach, senders periodically flood sender information to announce their presence in a network. Multicast receivers join a group by sending replies to the sender. The establishment of a mesh structure makes FGMP robust to node mobility. FGMP has a scalability problem due to flooding of control packets [19]. Another problem with FGMP is that it has not explicitly considered broadcast redundancy associated with the flooding of the join requests.

On-Demand Multicast Routing Protocol (ODMRP): Another example of a mesh-based multicast routing protocol is ODMRP [9]. ODMRP uses a forwarding group concept in which only a subset of nodes in a network is allowed to forward multicast packets. Group membership and multicast routes are established and updated by a source node on demand. When a multicast source has packets to send, but does not have any route to the multicast group, it broadcasts a Join-Query to the entire network. The Join-Query is periodically broadcast to refresh the membership information and routes. When an intermediate node receives the Join-Query packet, it retransmits the packet if the node detects that the packet is not a duplicate and that its time to live is non-zero. When a multicast receiver receives the Join-Query it prepares a Join-Reply and sends it back to the source node (section 2.4 covers ODMRP in detail). ODMRP does not generate excessive control overhead in high mobility situations because no control packets are triggered by link breaks. The method used by ODMRP to propagate the Join-Query is what most multicast routing protocols use and is known as blind flooding [8]. Blind flooding causes ODMRP to generate excessive control overhead when multiple sources are present.

An attempt has been made in ODMRP to address the problem of broadcast redundancy through the use of passive clustering as an optional broadcasting method for the Join-Query packet. In passive clustering, clusters are created dynamically as data transmission is in progress. Forwarding of a broadcast packet is only done by cluster-heads and gateway nodes. However, the use of passive clustering by ODMRP cannot adequately minimise the cost of route discovery in all the three topologies of the ELC-MANET being considered in this research. For example, using passive clustering in an ELC-matrix topology which is characterised by high connectivity is likely to yield very large clusters which usually lead to low throughput [35, 36]. Passive clustering is likely to yield low throughput in the ELC-matrix topology because it does not provide load balancing through limiting the size of a cluster.

Source Routing-based Multicast Protocol (SRMP): SRMP [37] is also one of the mesh-based on-demand multicast routing protocols. SRMP constructs a mesh for each multicast group using the concept of the forwarding group similar to that of ODMRP and FGMP. Four metrics are used to select forwarding group nodes: neighbourhood association stability, link signal strength, battery life, and link availability estimation. SRMP provides stable paths based on future prediction for a link state. The paths also guarantee minimum energy

consumption, nodes' stability with respect to their neighbours, and strong connectivity between nodes. The protocol consists of a route request phase and a route reply phase. When a source node that is not a group member wishes to join a group, it broadcasts a join request. On receiving the join request, a multicast receiver prepares a join reply and sends it back to the source node through neighbour nodes (or forward nodes) which satisfy predefined thresholds of the four mentioned metrics. This selection criterion of forward nodes is repeated at each intermediate node until the join reply reaches the source node.

SRMP offers reliability due to its selection of routes based on stability. The mesh structure enables SRMP to be robust to node mobility. However, SRMP has not specifically looked at broadcast redundancy associated with broadcasting of join requests. SRMP is likely to have a large component of broadcast redundancy in its control overhead.

Neighbour-Supporting Multicast Protocol (NSMP): NSMP [38] is a source-initiated mesh-based multicast routing protocol. Creation and maintenance of a multicast mesh in NSMP use a soft state approach similar to that of ODMRP. The key characteristic of NSMP is that it minimises control overhead by using a technique in which periodic broadcast messages from a source node aimed at maintaining a multicast mesh are not flooded to an entire network but are allowed to only reach mesh nodes and their neighbour nodes. Hence, NSMP has attempted to address the problem of broadcast redundancy which occurs in a process of route discovery and maintenance. However, the method used by NSMP to deal with broadcast redundancy can be improved by incorporating advanced pruning techniques such as those proposed in [7, 39, 40].

Additional examples of mesh-based multicast protocols: Other examples of mesh-based multicast routing protocols include Dynamic Core based Multicast routing Protocol (DCMP) [41], Protocol for Unified Multicasting through Announcements (PUMA) [42], and Multicast for Ad hoc Networks with Swarm Intelligence (MANSI) [43]. DCMP is an extension to ODMRP which attempts to address the broadcast redundancy problem by allowing only a subset of source nodes of a multicast group to broadcast messages aimed at route discovery and maintenance. The method used by DCMP for addressing the broadcast redundancy problem can also be made more effective by incorporating advanced pruning techniques of forward nodes reported in [7, 39, 40]. PUMA uses a receiver-initiated approach in which receivers join a multicast group by using the address of a core node thereby avoiding flooding a network with join requests from all sources of a multicast group as the case is with

protocols such as ODMRP. MANSI adopts the concept of swarm intelligence to minimise the number of nodes used to establish a forward node set of a multicast group as a whole. Both PUMA and MANSI have not explicitly considered reduction of broadcast redundancy generated by announcements from the multicast core nodes.

Hybrid multicast routing protocols

The tree-based multicast routing protocols and the mesh-based multicast routing protocols presented in the last two subsections are herein referred to as pure tree-based and pure mesh-based multicast routing protocols respectively. These pure tree-based and pure mesh-based multicast routing protocols usually have a primary goal of addressing only one of the aforementioned multicast challenges. The desire to address two or more of the multicast challenges in one protocol led researchers to the design of hybrid multicast routing protocols. A hybrid multicast protocol integrates beneficial attributes from a number of protocols into one protocol. Examples of hybrid multicast routing protocols are presented in the following paragraphs.

Ad hoc Multicast Routing (AMRoute) protocol: AMRoute [44] is an example of a hybrid multicast routing protocol. It is a proactive protocol which creates a bidirectional shared multicast tree using only source nodes and receiver nodes as tree nodes for data delivery. A key characteristic of AMRoute is that it establishes a multicast tree on top of a mesh structure. The mesh structure is created by using a unicast protocol. The creation of the mesh structure is in such a way that between any pair of multicast group members there is a mesh of unicast tunnels connecting the pair. The presence of the mesh structure enables AMRoute to maintain its multicast tree during situations of topology changes. One of the disadvantages of AMRoute is that it may have temporary routing loops. When node mobility is high, AMRoute may construct non-optimal trees. Being a proactive protocol, AMRoute maintains routing information throughout its execution time or “lifetime”, and therefore maintaining a mesh structure for such a period of time means that a significant amount of the limited battery energy of a MANET device can be used for control overhead. Since AMRoute relies on a unicast routing protocol for its operations, it may have no control over broadcast redundancy associated with a route discovery process initiated by the unicast protocol.

Multicast Core-Extraction Distributed Ad hoc Routing (MCEDAR) protocol: MCEDAR [45] is another example of a hybrid multicast routing protocol. It integrates the approach of

tree-based protocols and that of mesh-based protocols into one protocol. A mesh is used by MCEDAR as the underlying infrastructure so as to tolerate a few link breakages without reconfiguration of the infrastructure. MCEDAR achieves efficiency by using a forwarding mechanism on the mesh that creates an implicit route-based forwarding tree. The forwarding tree provides minimum distances for propagation of packets from a source node to the receiver nodes.

MCEDAR has attempted to address broadcast redundancy resulting from any broadcast operation through the use of a subset of nodes in a network called core nodes which act as forward nodes. It is possible that the clustering approach used by MCEDAR through the use of core nodes can be made more effective in reducing broadcast redundancy by incorporating advanced cluster-based pruning techniques of forward nodes proposed in [46, 47]. According to the work reported in [45], MCEDAR had not yet been tested by a simulation or other means and therefore it is difficult to tell if MCEDAR remains efficient in terms of resource usage as the size of a network increases. O. Badarneh and M. Kadoch [20] have pointed out that in a high mobility environment, nodes running MCEDAR need to change their cores frequently thereby increasing the control overhead.

Multicast Zone Routing Protocol (MZRP): MZRP [48] is a hybrid multicast routing protocol which establishes a shared multicast tree. It uses a proactive approach to maintain multicast membership information at every node located within a local routing zone of a multicast tree node, and uses a reactive approach to establish a multicast tree across local routing zones. MZRP is a source-initiated protocol. The first source node to join a multicast group becomes the group leader of the multicast group. The group leader periodically broadcasts group leader messages to the entire network so as to inform nodes about the existence of a multicast group and the group leader. Every multicast tree node periodically broadcasts multicast tree membership messages to nodes within its local routing zone in order to inform the nodes about the presence of the multicast tree node. The messages broadcast within a local routing zone of a multicast tree node helps the nodes within the routing zone to join a multicast group with minimal control overhead since a join request from such nodes can be sent by a unicast routing method to a nearest multicast tree node instead of broadcasting it to an entire network. Hence, a second source node wanting to join a multicast group only adds a branch to a multicast tree established by a multicast group leader.

MZRP has attempted to reduce the cost of route discovery by preventing source nodes except the group leader from broadcasting their join requests to an entire network. However, the proactive approach of maintaining multicast membership information within a local routing zone of a multicast tree node makes maintenance of routes in MZRP to be very costly in terms of bandwidth and energy usage. MZRP has not addressed broadcast redundancy arising from the periodic messages sent by a group leader to an entire network.

Efficient Hybrid Multicast Routing Protocol (EHMRP): EHMRP [49] is an extension to ODMRP which is aimed at solving the scalability problem of ODMRP in the presence of multiple sources. The design of EHMRP incorporates the routing approach of Differential Destination Multicast (DDM) protocol [50] and a low overhead clustering mechanism of MCEDAR [45] into one protocol. Based on MCEDAR's clustering approach, nodes are classified in EHMRP into core and normal nodes. Join requests from source nodes are broadcast to an entire network but only core nodes are allowed to forward the join request packets. Data packets are forwarded to receivers using the DDM protocol. In DDM, a source node encodes multicast receiver addresses in multicast data packets using a special DDM packet header field. The multicast data packets are routed to their destinations using an underlying unicast routing protocol. Simulation experiments conducted in [49] indicate that EHMRP has a better packet delivery ratio than ODMRP under high node mobility and a small network load; however, EHMRP produces less packet delivery ratio than ODMRP as the network load increases. This low performance of EHMRP in the presence of high network traffic is not a surprise because the DDM protocol which EHMRP uses for data forwarding is designed for small networks [49, 50].

Additional examples of hybrid multicast routing protocols: Additional examples of hybrid multicast routing protocols include Power-Controlled Hybrid Multicast Routing (PCHMR) protocol [51], Hybrid Overlay Multicast Routing Protocol (HOMRP) [52], and Hybrid Zone-based Multicast Ad hoc On-demand Distance Vector (HZMAODV) routing protocol [53]. PCHMR constructs a hybrid network structure that combines tree-based and mesh-based approaches. Route discovery in PCHMR is the same as that of AODV [18] which consists of a route request phase and a route reply phase. Since AODV has not seriously considered reduction of broadcast redundancy associated with route discovery and maintenance, it means that PCHMR has also not considered broadcast redundancy. HOMRP is aimed at providing efficient packet delivery by integrating multicasting and unicast tunnels

in one protocol. Simulation experiments conducted in [52] show that HOMRP performs worse than ODMRP with respect to packet delivery ratio and end-to-end delay as mobility of nodes increases. HZMAODV divides a geographical network area into zones based on nodes' positions and constructs inter-zone forwarding paths and intra-zone paths for multicasting. HZMAODV can only be used with the help of a positioning system such as a Global Positioning System (GPS).

Adaptive multicast routing protocols

The hybrid technique can be considered as additional intelligence given to multicast routing protocols for dealing with multicast challenges. However, since the hybrid protocol remains static as the network environment changes, the protocol is still not intelligent enough to effectively address the multicast challenges associated with the dynamics of the network environments in MANETs. Consequently, researchers came up with adaptive multicast routing protocols as a way of providing more intelligence in the protocols. Examples of adaptive multicast routing protocols are given the following paragraphs.

Adaptive Multicast Routing Protocol (AMRP): AMRP [54] is an example of an adaptive multicast routing protocol. It is aimed at addressing the scalability problem of ODMRP in situations of having multiple source nodes. In AMRP, source nodes broadcast join request packets depending on a probability variable which is determined by the current packet delivery ratio in the network. When the packet delivery ratio is low, the probability of senders broadcasting join request packets is lowered as well so as to prevent escalation of collisions. Senders are given a high probability of broadcasting join request packets when the packet delivery ratio is high. The packet delivery ratio is recalculated periodically to reflect real-time network conditions. AMRP significantly minimises control overhead and greatly improves packet delivery ratio of ODMRP. However, AMRP can reduce further control overhead by minimising broadcast redundancy of join requests through the use of advanced pruning techniques of forward nodes proposed in [7, 39, 40].

Adaptive Shared Tree Multicast (ASTM) routing protocol: ASTM [55] is another adaptive multicast routing protocol which combines the approach of a shared multicast tree and that of a per-source multicast tree. Upon receiving a data packet from a multicast source, a receiver node can decide to have the next data packets from the source delivered to it through either a shared multicast tree rooted at a rendezvous point or a per-source multicast

tree depending on the path length from the source node recorded in the packets. Thus, ASTM attempts to maintain minimum distances between the source node and the receivers through adaptive switching between a shared tree and a per-source tree delivery method. ASTM establishes a multicast tree on top of a cluster-based organisation of nodes. The use of a multicast tree approach together with the establishment of clusters enables ASTM to be generally efficient in terms of resource usage. However, incorporation of advanced cluster-based pruning techniques of forward nodes proposed in [46, 47] can significantly minimise broadcast redundancy associated with any broadcast operation performed by ASTM.

Adaptive Core Multicast Routing Protocol (ACMRP): ACMRP [56] is an adaptive multicast routing protocol. It is an on-demand core-based multicast routing protocol. Each multicast group has only one core node. Unlike in non-adaptive core-based multicast routing protocols in which a core is not changed once elected, a core node in ACMRP is changed periodically. A node is elected as a core node if it has the minimum hop counts of routes towards multicast group members and its residual battery energy is enough to support operations until the next election time. A core node initiates the creation of a multicast mesh. Once a mesh is established, it is maintained by the core node through periodic flooding of join requests. In ACMRP, source nodes join a multicast group by replying to join requests broadcast by a core node and not by flooding join requests like in ODMRP. Hence, ACMRP generates less control overhead than ODMRP as the number of sources increases. ACMRP is robust against topology changes due to the establishment of a mesh structure. The first drawback of ACMRP is that its paths for packet delivery are not optimal. ACMRP has not considered reduction of broadcast redundancy arising from the flooding of join requests from a core node.

New Polymorphic Multicast Routing Protocol (NPMRP): NPMRP [57] is an adaptive as well as a hybrid multicast routing protocol which operates by dynamically switching between a proactive mode and a reactive mode depending on the current measurements of a node's battery life and the level of mobility. Generally, NPMRP switches to a proactive mode when both a node's battery energy and mobility speed are high. The reason given by the authors for such a switch is that since the battery energy is high the node can maintain topology information, which in turn helps the node to react faster to topology changes resulting from high mobility. Hence, the switch is intended to maintain better connectivity during situations of high mobility. When a node's battery energy is low, NPMRP generally switches to a

reactive mode so as to prolong a node's battery life. NPMRP is implemented in [57] using MZRP [48] for the proactive mode and ODMRP [9] for the reactive mode. According to the simulations conducted in [57], NPMRP performed better on power conservation compared to zone-based variants of ODMRP and MAODV. However, NPMRP has inherited some of the drawbacks of ODMRP and MZRP. For example, NPMRP is likely to generate excessive broadcast redundancy like ODMRP when the number of sources increases.

Additional examples of adaptive multicast routing protocols: Other examples of adaptive multicast routing protocols include Hierarchical Multicast routing PROtocol (HIMPRO) [58], Optimized Polymorphic Hybrid Multicast Routing protocol (OPHMR) [59], Enhanced On-Demand Multicast Routing Protocol (E-ODMRP) [60], and Adaptive Demand-driven Multicast Routing (ADMR) protocol [61]. HIMPRO reduces broadcast redundancy by using a cluster-based organisation of nodes in which only cluster-heads and gateway nodes are adaptively selected to forward packets. It is difficult to tell if HIMPRO (a mesh-based protocol) performs better than other mesh-based protocols since a performance evaluation conducted in [58] used MAODV, a tree-based protocol, for comparison. OPHMR is an improvement on NPMRP aimed at reducing broadcast redundancy by using a multi-point relay approach in which only a subset of neighbours of a sender node are selected to forward packets. It is possible that OPHMR can be more effective in reducing broadcast redundancy if advanced pruning techniques of forward nodes proposed in [7, 39, 40] are incorporated in the protocol.

E-ODMRP is a variant of ODMRP which performs a periodic refresh of routing information at a rate which is dynamically adapted to node's mobility. The key feature of ADMR is that it uses the pattern of packet flows to detect link breaks and expired routes. Both ADMR and E-ODMRP have not explicitly addressed broadcast redundancy resulting from their flooding operations.

Table 2.1 presents a comparison of the MANET multicast routing protocols which were reviewed in this research. The comparison is based on whether broadcast redundancy is considered in a routing protocol or not. Table 2.1 also serves as a summary of all the reviewed multicast routing protocols.

Table 2.1 Comparison of MANET multicast routing protocols

Protocol	Routing method	Broadcast redundancy consideration
AMRIS	tree	not explicitly considered
MAODV	tree	no
ABAM	tree	not explicitly considered
PLBM	tree	yes
STAMP	tree	not explicitly considered
WCMP	tree	not explicitly considered
TMP-MPRs	tree	yes
BEMRP	tree	not explicitly considered
SOM	tree	not explicitly considered
STM RP	tree	not explicitly considered
ITAMAR	tree	not explicitly considered
CAMP	mesh	not explicitly considered
FGMP	mesh	not explicitly considered
ODMRP	mesh	yes, significantly considered
SRMP	mesh	not explicitly considered
NSMP	mesh	yes
DCMP	mesh	yes
PUMA	mesh	not explicitly considered
MANSI	mesh	not explicitly considered
AMRoute	hybrid	no
MCEDAR	hybrid	yes
MZRP	hybrid	no
EHMRP	hybrid	not explicitly considered
PCHMR	hybrid	no
HOMRP	hybrid	not explicitly considered
HZMAODV	hybrid	no
AMRP	adaptive	yes
ASTM	adaptive	yes
ACMRP	adaptive	no
NPMRP	adaptive	not explicitly considered

HIMPRO	adaptive	yes
OPHMR	adaptive	yes
E-ODMRP	adaptive	not explicitly considered

Remarks on existing multicast routing protocols

This research considers the four categories of multicast routing protocols, namely tree-based, mesh-based, hybrid and adaptive multicast routing protocols, reviewed in this research to be the current major classes of multicast routing protocols for MANETs. It is further believed in this research that a routing strategy of any existing multicast routing protocol is either directly or indirectly related to one of more of the four major categories of the multicast routing protocols identified in this research. Hence, it is assumed in this thesis that the literature review on the current multicast routing protocols for MANETs is exhaustive enough for meaningful generalised deductions to be made on the current status of the protocols.

2.3 Current MANET Broadcasting Methods

A simple method of message broadcasting is blind flooding in which every node in the network that receives the packet for the first time is required to rebroadcast it. Blind flooding generates many redundant retransmissions. When there are multiple source nodes simultaneously releasing broadcast packets by the blind flooding method, the resulting retransmissions may lead to the broadcast storm problem [8, 62] in which redundant packets cause bandwidth contention and collisions.

The literature contains a substantial amount of work on the issue of efficient broadcasting in MANETs. Researchers have used various ways to tackle the efficiency problems of blind flooding [46, 63, 64]. For a better understanding of this topic on efficient broadcasting in MANETs with respect to clustering and the concept of Connected Dominating Set (CDS), this thesis organises the methods reported in the literature into four categories: (1) clustering without CDS, (2) CDS without clustering, (3) clustering combined with CDS, and (4) other methods.

CDS is a concept used in graph theory [11, 65, 66]. Given an undirected graph $G(V, E)$, where V is a set of vertices and E is a set of edges, a dominating set is a subset V' of V such

that every vertex in $V - V'$ is adjacent to at least one vertex in V' . A dominating set V' which produces a connected vertex induced subgraph on V' is known as a Connected Dominating Set (CDS). MANETs are modelled using graph theory and that is why CDS concepts are associated with MANET broadcasting methods. The application of the CDS concept in MANET broadcasting is explained in the first paragraph of subsection 2.3.2.

2.3.1 Clustering without CDS

Clustering is used as a method for addressing broadcast redundancy in large networks [67]. A cluster-based network is a two-level hierarchical network. A clustering process converts a dense network into a sparse one that consists of cluster-heads and gateways as the only nodes that are responsible for propagation of packets in a network. The literature contains some work on non-CDS-clustered MANETs with emphasis on broadcasting.

R. Purtoosi et al. [68, 69] proposed a cluster-based flooding algorithm that classifies flooding traffic into internal and external traffic from each cluster's point of view. The algorithm proposes three sets of nodes to be responsible for rebroadcasting packets, namely cluster-heads, gateways and border nodes. Only gateways are in charge of sending the internal traffic to the adjacent clusters. External traffic can be rebroadcast by cluster-heads, gateways and border nodes.

Another cluster-based technique for broadcast was proposed by A. Pandey and J. S. Lim [70]. The technique uses the Time Division Multiple Access (TDMA) method to assign a channel to a cluster which is different from channels used by neighbouring clusters. Relay nodes are used to relay packets from one cluster-head to another, and are therefore equipped with the capability of hearing more than one cluster-head. A. Pandey and J. S. Lim [70] argue that their protocol does not only save network resources by minimising redundant packets, but also minimises end-to-end delay.

I. Stojmenovic et al [71] proposed a method which reduces the overhead of broadcasting by applying the concept of internal nodes. They pointed out that maintenance of internal nodes requires much less communication overhead than the maintenance of cluster structure of nodes. I. Stojmenovic et al [71] also produced a clustering algorithm for broadcasting. They finally compared the performance of the two algorithms, and found out that the one based on internal nodes used less network resources than the clustering algorithm to accomplish the

same task. This observation is in line with that of other researchers who came up with the idea of adding CDS functionality to clustering algorithms.

2.3.2 CDS without clustering

A major approach to efficient broadcasting models a broadcast process as a flood tree operation in which a source node and a subset of nodes form the flood tree, while the rest of the nodes in the network are adjacent to at least one node on the tree [11]. The nodes on the tree are called forward nodes and form a CDS. Redundant retransmissions in a broadcast process are therefore reduced by minimising the size of CDS, which is done by finding a solution (usually heuristics) to the Minimum Connected Dominating Set (MCDS) problem [11, 12].

The literature presents a number of approximation algorithms of the MCDS problem. J. Wu and H. Li [72] present a simple heuristic algorithm of MCDS which uses a marking process to mark every vertex in a given connected and unweighted graph $G(V, E)$. The process defines $m(v)$ as a marker for vertex $v \in V$, which is either **T** (marked) or **F** (unmarked). All vertices are assumed unmarked initially. $N(v) = \{u | \{v, u\} \in E\}$ represents the open neighbour set of vertex v and v has $N(v)$ initially. The process begins by first assigning marker **F** to every v in V . Next, every v exchanges its open neighbour set $N(v)$ with all its neighbours. Finally, every v assigns its marker $m(v)$ to **T** if there exist two unconnected neighbours. The marking process is followed by the application of two rules which help to reduce the number of nodes marked **T**. When the marking process is over, all the nodes marked **T** form a CDS. Only nodes that belong to CDS are required to rebroadcast a packet when they receive it, while the rest of the nodes in the network just receive the packet.

Other MCDS heuristics which aim to reduce redundant transmissions in blind flooding are self pruning and dominant pruning proposed by H. Lim and C. Kim [7, 40]. The self pruning algorithm only utilises information of directly connected neighbours (or 1-hop neighbours). A node that runs a self pruning algorithm does not rebroadcast a packet if all its neighbours have been covered by the previous transmission. Dominant pruning exploits 2-hop neighbour set information to compute each node's forward set.

W. Lou and J. Wu [39] extended the work of H. Lim and C. Kim [7, 40] by proposing two algorithms: total dominant pruning and partial dominant pruning. Both algorithms use the previous forward node's 2-hop neighbour set information to further minimise redundant

retransmissions. In the total dominant pruning, the sender uses the broadcast packet to piggyback all its 2-hop neighbour set information and deliver the information at the receiver. The receiver uses the delivered information to prune all the nodes in the sender's 2-hop neighbour set from the receiver's 2-hop neighbour set that need to be covered. The partial dominant pruning algorithm avoids the piggybacking technique and operates by extracting the 1-hop neighbour set of the common neighbours of both the sender and the receiver from the receiver's 2-hop neighbour set. Simulation results reported in [39] indicate that both total dominant pruning and partial dominant pruning have better performance than dominant pruning, one of the pruning algorithms by H. Lim and C. Kim [7, 40]. The simulation results in [39] further indicate that the difference in performance between total dominant pruning and partial dominant pruning is insignificant.

M. Hoque et al. [73] worked on the partial dominant pruning algorithm and proposed enhanced partial dominant pruning algorithm. The algorithm improves upon partial dominant pruning by taking advantage of information that a node about to retransmit a broadcast packet can overhear from its neighbours. The main idea in the enhanced partial dominant pruning algorithm is to delay the retransmission for a moderate amount of time so as to give chance to the neighbours to go first. M. Hoque et al. [73] argue that with the proper selection of the delays, a noticeable reduction in redundancy can be achieved.

2.3.3 Clustering combined with CDS

Some researchers have worked on how the flood-tree-based broadcast can be applied to clustered MANETs. W. Lou and J. Wu [11] pointed out that cluster-based broadcast still has more broadcast redundancy than the flood-tree-based broadcast since two neighbouring clusters usually have more than one gateway to forward the broadcast packet. This observation plus perhaps other factors led to the development of clustered MANETs based on CDS.

A clustering algorithm with emphasis on broadcasting is presented by N. Mitton and E. Fleury [74]. The algorithm uses the concept of "clustering trees" in which each cluster maintains a tree structure of nodes that belong to a dominating set. The clustering trees form a spanning forest which is a dominating set of a network but not a CDS because the trees are independent. To perform a reliable broadcasting task in the whole network, the trees are connected by a gateway selection process. N. Mitton and E. Fleury [74] showed by

simulation that their algorithm saves more retransmissions than most of the existing methods. The method looks powerful however, as pointed out by W. Lou and J. Wu [39], maintaining tree structures in mobile nodes is not an efficient way of using the limited battery energy which powers the nodes.

K. Drira and H. Kheddouci [75] developed a clustering algorithm based on the concept of CDS. The algorithm is similar to that of N. Mitton and E. Fleury [74] in the sense that they both use tree and forest data structures. As a way of eliminating loops, each cluster maintains a tree structure formed by nodes belonging to a dominating set and rooted at the cluster-head.

One of the serious research efforts on broadcast protocols of CDS-based clustered MANETs was conducted by J. Wu and W. Lou [46]. The two researchers developed a broadcast protocol which partitions a network into clusters, each of which is controlled by a cluster-head. Each cluster-head applies a greedy set cover algorithm [7] to select its forward node set locally to cover all cluster-heads with its vicinity defined to be within 2.5 or 3-hop coverage area. The forward node set (or gateways) is selected only at the time when a cluster-head wants to relay a packet (i.e. dynamically or on-demand). The algorithm uses a pruning technique to remove forward nodes already covered in the previous transmissions before applying the greedy set cover algorithm.

W. Lou and J. Wu [47] extended their work in [46] by developing a static cluster-based backbone infrastructure for broadcasting and compared its performance to the dynamic backbone infrastructure previously developed in [46]. The static backbone (cluster-based source-independent CDS) consists of fixed cluster-heads and selected source-independent gateways while a dynamic backbone (cluster-based source-dependent CDS) consists of fixed cluster-heads and dynamically selected gateways. W. Lou and J. Wu [47] pointed out that maintaining a static backbone at all times for broadcasting is costly and unnecessary. Their simulation results showed that broadcasting in the dynamic backbone that uses the pruning technique has less broadcast redundancy than in the static backbone.

2.3.4 Other methods of broadcasting

The literature contains a number of MANET broadcasting methods with little or no dependence at all on clustering and/or CDS. Such methods are hereinafter referred to as non-cluster-CDS broadcasting methods. Probabilistic schemes [76, 77] are among the non-cluster-CDS methods. In a probabilistic method, a node that receives a broadcast packet for the first

time decides to rebroadcast it with probability p and takes no action with probability $1 - p$. Some variants of probabilistic algorithms use a dynamically adjusted p that depends on the density of nodes, among other factors [78, 79]. Some researchers have analysed the performance of probabilistic methods and have found them worthwhile [80 - 82].

Counter-based broadcasting methods [83] are somehow related to probabilistic schemes in the sense that instead of using a probability p they use a packet counter to determine whether to rebroadcast a packet or not. When a node receives a broadcast packet for the first time it initiates a counter and delays the retransmission for a specified period. The counter is incremented each time a copy of the packet is received. At the end of the delay period, the packet is rebroadcast if the counter value is less than a threshold value; otherwise the node does nothing with the packet. Variants of counter-based schemes exist, for example, [84 - 86] use adjustable counters which depend on the density of nodes, among other factors; and [87] uses colour instead of a counter.

Another category of non-cluster-CDS broadcasting schemes is that of location-aided broadcasting methods [88 - 91]. When a node using a Location-based scheme receives a broadcast packet, it uses the location of the sender to calculate the additional coverage which can be covered by rebroadcasting the packet. The node rebroadcasts the packet if the additional coverage area is more than a specified minimum value. Every node in the network is required to determine its location by methods such as the Global Positioning System (GPS).

Some non-cluster-CDS schemes seem not to belong to any of the well known categories such as location-aided or probabilistic schemes, and among such methods we have the one by Y. Cai et al. [92] which only allows nodes that are close to some partition edge of the broadcast coverage to forward a broadcast packet. Other “difficult to classify” techniques are [93, 94] which are both based on dynamically adjusting delaying time for retransmission so as to collect duplicate packets and minimise broadcast redundancy.

2.4 Existing MANET Clustering Methods

Some MANET broadcasting methods such as those presented in subsection 2.3.1 and 2.3.3 make use of clustering techniques. Some researchers classify MANET clustering methods into two categories: active and passive clustering [95, 96]. In active clustering [97 - 99],

mobile nodes set up and maintain clusters independent of data transmission. Nodes periodically and cooperatively exchange information which is used to elect cluster-heads. One of the major advantages of active clustering is that clusters are always available and therefore a node can transmit its time sensitive data without delay. Periodic flooding of control packets in the network, which happens even in the absence of data transmission, is a major drawback of active clustering since the limited network bandwidth and battery energy resources are utilised inefficiently.

Passive clustering [100 - 102] creates clusters “on-the-fly” as data transmission is in progress. On-going data traffic is used to send clustering related information such as node ID and the state of a node in a cluster. Mobile nodes collect neighbour information through promiscuous packet receptions. Compared with active clustering, passive clustering significantly reduces control overhead. However, passive clustering requires a noticeably long set up time which is experienced whenever data transmission is being started. Such set up latency may not be acceptable in some time critical applications.

2.4.1 Review of active clustering methods

Apparently, most of the research work currently available in the literature is on active clustering. Hence, in this thesis as is also the case in the literature, the term “active clustering” is used interchangeably with the term “clustering”. The method of clustering (or active clustering) of mobile nodes with the purpose of improving network performance started more than two decades ago as is evidenced by the work of A. Ephremides et al. [103], who developed a clustering algorithm that selects a node with the lowest-ID in a neighbourhood to be a cluster-head. The lowest-ID algorithm’s strength is in its implementation simplicity [104]. Another historical and well known clustering algorithm is the Highest-Degree proposed by M. Gerla and J. Tsai [105]. In the Highest-Degree algorithm, a node with the highest number of neighbours (i.e. highest degree) is elected as a cluster-head. Both Lowest-ID and Highest-Degree algorithms do not take mobility metrics into consideration, and hence highly mobile nodes are equally likely to be elected as cluster-heads thereby resulting in frequent re-clustering [104].

Most of the clustering algorithms which came after the Lowest-ID and the Highest-Degree base their operations on the concept of computing a weight (or a score) for each node so that the node with the highest or the lowest score in its neighbourhood is elected as a cluster-head

[35, 106 - 109]. The computation of a score is based on weighting factors such as the number of neighbours (i.e. degree of connectivity), the amount of residual battery energy, and the degree of mobility. This research considers the weight-based algorithms of S. Basagni [106] as among the best clustering algorithms currently available in the literature in terms of clarity and mathematical soundness. A good implementation of S. Basagni's work is in [110].

A clustering algorithm can be referred to as either a 1-hop algorithm or a k -hop (or d -hop) algorithm depending on the number of hops from a member node of a cluster to a cluster-head. In 1-hop clustering algorithms [35, 106, 111, 112], the connection between any member of a cluster and the cluster-head is through one hop (i.e. every cluster member is adjacent to the cluster-head). Connection between a member node and a cluster-head is through at least one but at most k hops [113 - 115].

Some researchers have worked on some other aspects of clustering in addition to the conventional weight-based selection of cluster-heads and number of hops between member nodes and the cluster-head. For instance, the work in [116 - 118] has focussed on security issues. Quality of service in a clustering scenario is covered in [119, 120]. Clustering with emphasis on group mobility is covered in [121]. While almost all the clustering algorithms have been covered so far in this thesis are on 2-level hierarchies of clusters, the work in [122] is on a 3-level hierarchical design of clusters. Some critical evaluations of clustering algorithms are presented in [123 - 125].

2.4.2 Basagni's distributed clustering algorithms

S. Basagni [106] developed two algorithms for the setup and maintenance of a clustering organisation of a wireless ad hoc network. The choice of a cluster-head in the algorithms is based on a generic *weight* (a real number ≥ 0) associated with each node. The bigger the *weight* of a node, the more suitable the node is for the role of a cluster-head. The first of the two algorithms is the Distributed Clustering Algorithm (DCA), which is suitable for ad hoc networks whose nodes do not move or move "slowly" (i.e. "quasi-static" networks). DCA assumes that nodes are stationary during the setup phase of a clustering process. The second algorithm which is intended for highly mobile networks is called the Distributed and Mobility Adaptive Clustering (DMAC) algorithm. With the DMAC algorithm, nodes can move while the cluster setup phase is in progress.

In both algorithms, every node v in the network is assigned a unique identifier (ID). A weight w_v is assigned to each node $v \in V$ of the network. The algorithms are executed at every node (i.e. distributed) by using information gathered from only the neighbourhood of each node. Basagni's algorithms ensure that the following properties of clustering mobile ad hoc networks are satisfied:

- 1) Every ordinary node has at least a cluster-head as a neighbour (dominance property).
- 2) Every ordinary node affiliates with the neighbouring cluster-head that has a bigger weight.
- 3) No two cluster-heads can be neighbours (independence property).

S. Basagni [106] did not specify what constitutes a node weight w_v in both DCA and DMAC. However, the two algorithms form a solid framework upon which any weight-based distributed clustering algorithm can be built. DCA is used in this research as a framework for the proposed CTRDM's clustering algorithm (covered in chapter 4). The clustering algorithm of CTRDM extends DCA by first introducing the concept of setting the maximum size for clusters with the aim of achieving load balancing and by specifying the factors which make up a node weight.

2.5 Analysis of ODMRP

ODMRP [9] is an on-demand multicast routing protocol which implies that routes are established and maintained only when there is data to be transmitted. The reactive technique used by ODMRP is aimed at reducing control overhead and improving scalability. ODMRP uses the concept of a forwarding group, which is a set of nodes responsible for forwarding multicast data, to create a forwarding mesh for each multicast group. The use of a forwarding mesh avoids drawbacks associated with tree-based multicast such as intermittent connectivity, traffic concentration, frequent tree reconfiguration, and non-shortest path in a shared tree. Properties of ODMRP include simplicity, low channel and storage overhead, use of up-to-date and shortest routes, robustness to host mobility, maintenance and exploitation of multiple redundant paths, and scalability to a large number of nodes.

2.5.1 Protocol overview

ODMRP, like an on-demand unicast routing protocol, basically consists of a route request phase and a route reply phase. A source node in ODMRP creates and updates group

membership and multicast routes on an on-demand basis. When a multicast source has packets to send to a group whose routing information is not known, it broadcasts a Join-Query message with data payload piggybacked. The Join-Query packet is periodically broadcast to the entire network throughout the time when the source node still has packets to send. The periodic flooding of the Join-Query packet is aimed at refreshing the membership information and updating the routes.

When a multicast receiver (a group member) receives a Join-Query message, it creates and broadcasts a Join-Reply message to its neighbours. When a node receives a non-duplicate Join-Reply, it checks if the node is on the route to the source by matching its own ID with the next hop recorded in one of the entries of the Join-Reply. If it does, the node realises that it is part of a forwarding group and sets the FG_FLAG (forwarding group flag). After setting the FG_FLAG, the node creates its own Join-Reply and broadcasts it by using a next hop address obtained from its routing table. In this way, the Join-Reply packet is propagated by each forward group member until it reaches the source node via the selected path. This whole process explains how routes from sources to receivers are constructed and a multicast mesh of nodes called a forwarding group is built. Figure 2.1 shows an example of how a Join-Query and its corresponding Join-Reply propagate in a network.

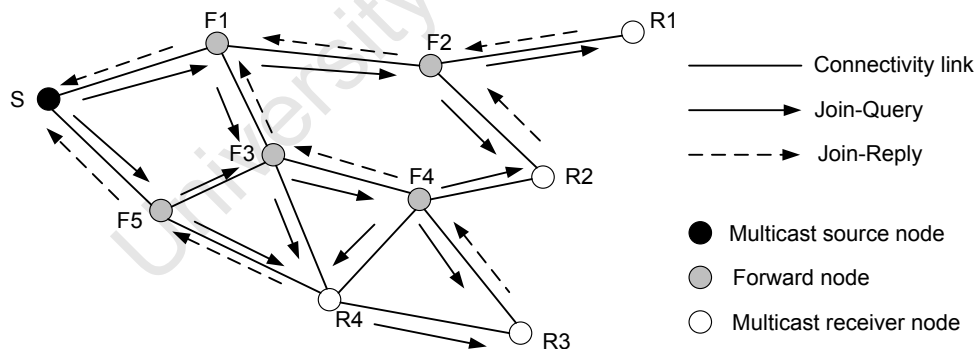


Figure 2.1 Propagation of Join-Query and Join-Reply in ODMRP

The process of establishing and refreshing multicast routes and forwarding groups depends on reliable transmission of Join-Replies such that improper delivery of Join-Replies may result in failure to achieve effective multicasting. ODMRP assumes that the task of ensuring reliable transmission of Join-Replies is not performed by the MAC layer, and therefore it implements its own packet acknowledgement mechanism to guarantee reliable delivery of a Join-Reply. The packet acknowledgement technique used by ODMRP is what is known as

“passive acknowledgment”. When a node v transmits a packet to its upstream neighbour w , its downstream neighbour u can overhear the broadcast of the packet. ODMRP uses this overhearing of packet transmission as a passive acknowledgment which ensures that a Join-Reply from node u has been delivered to its neighbour v . The source node sends an explicit packet acknowledgment to its downstream neighbour (i.e. the neighbour from which it received a Join-Reply) since it does not have any next hop to transmit the Join-Reply to unless it also acts as a forwarding group node for other multicast sources.

It is possible for a node to receive no “passive acknowledge” from an upstream neighbour due to factors such as the hidden terminal problem [126] and the moving away of the upstream neighbour. When no acknowledgement is received from an upstream neighbour or when the node has received acknowledgements from some of the upstream neighbours but not from all its upstream neighbours within a timeout period, it retransmits the Join-Reply. If no acknowledgement is received from a particular upstream neighbour after a number of retransmissions, the node concludes that the route is invalid. When this happens, the node tries to find an alternative route by broadcasting a message to its neighbours specifying that the next hop to a set of multicast sources cannot be reached. Upon receiving the message, each neighbour creates and unicasts a Join-Reply to its next hop for onward transmission if it has a route to the multicast sources. If no route is known, a node broadcasts a message specifying that the next hop is not available.

When a node wants to leave a multicast group in ODMRP, it is not required to send explicit control packets. When a multicast source needs to leave the group, it simply stops sending Join-Query packets. A multicast source stops sending Join-Query messages when it does not have any more data to send to the group. A multicast receiver which is no longer in need of data from a particular multicast group stops sending Join-Reply packets for that group. Failure to refresh the forwarding status of nodes in the forwarding group within a timeout period results in demotion of the nodes to the non-forwarding status. Forwarding nodes refresh their status information when they relay Join-Replies. A node that wants to join a multicast group as a source simply starts sending Join-Query packets. Joining a particular multicast group as a receiver is done by sending a Join-Reply to the source node of that multicast group.

Selection of timer values for the route refresh interval and forwarding group timeout can affect the performance of ODMRP. Timer values need to be adaptive to network

environments such as traffic type, traffic load, mobility pattern, mobility speed, and channel capacity to achieve optimum ODMRP performance. Using small route refresh intervals yields frequent availability of fresh route and membership information at the expense of more control packets which can easily lead to network congestion. On the other hand, choosing large route refresh intervals may result in outdated route and membership information even though less control traffic is generated. Hence, in highly mobile networks using large route refresh intervals may produce poor performance of ODMRP. Selecting small values for the forward group timeout interval in networks with heavy traffic load is desirable since unnecessary nodes can timeout quickly thereby avoiding excessive redundancy. In networks with high node mobility, high values of the forward group timeout are preferred since more alternative paths become available. The value for the forward group timeout interval must be larger than (e.g. 3 to 5 times) that of the route refresh interval.

Operation of ODMRP requires that a node should maintain a routing table. An entry is inserted or updated in a routing table when a non-duplicate Join-Query is received. Fields of the table include the destination address (i.e. the source of the Join-Query) and the next hop to the destination (i.e. the last node that propagated the Join-Query). Information in the routing table is used to provide the next hop address when transmitting Join-Replies. Another ODMRP table is a forwarding group table which is maintained by a forwarding group node of a multicast group. Information contained in a forwarding group table includes a multicast group ID and the time when the node was last refreshed. A node uses a message cache to detect duplicate packets. When a node receives a new Join-Query or data, it stores the source address and the ID of the packet in its message cache. Information in the message cache is stored temporarily. Methods such as LRU (i.e. Least Recently Used) and FIFO (First In First Out) are used to expire and remove old entries so as to prevent the size of the cache from growing too big.

2.5.2 Route discovery and multicast tree construction

Route discovery in ODMRP operates by broadcasting a Join-Query packet. According to the basic operation of ODMRP's broadcast method, every node in the network rebroadcasts the packet if it has received it for the first time. This method of broadcast is known as blind flooding. Blind flooding is an inefficient way of broadcasting since it gives rise to redundant retransmissions [8, 62]. Therefore, even though ODMRP is regarded as an efficient and

scalable protocol as a whole especially in terms of minimising control overhead, its basic or original broadcast method for route discovery is not efficient.

ODMRP offers passive clustering as an optional broadcast method aimed at addressing the broadcast redundancy of blind flooding. Passive clustering is an on-demand method which constructs and maintains clusters only when transmission of data packets is in progress. The data packets piggyback cluster-related information such as node ID information and the state of a node in a cluster. Selection of a cluster-head is based on the “First Declare First Wins” rule in which a node which first declares to be a cluster-head takes control of a cluster and all the nodes within its transmission range get affiliated to it. Passive clustering minimises broadcast redundancy by letting only cluster-heads and gateway nodes rebroadcast packets while the rest of the nodes in the network just receive the packets.

When a multicast receiver sends a Join-Reply to a multicast source, it uses the route taken by the first Join-Query packet to be received. Hence, the route taken by the Join-Reply offers minimum delay and is therefore considered as the shortest path between the multicast source and the receiver. A combination of all the routes from multicast receivers to a particular multicast source forms a multicast tree, which is based on the shortest path. For example, the dashed arrows in Figure 14 form a shortest-path multicast tree. Multicast sources use the multicast tree for delivery of data to receivers. ODMRP uses the multicast tree construction method based on the shortest path as its basic or original method. An optional multicast tree construction method is available in ODMRP, which uses a mobility prediction technique based on GPS to select the most stable routes that would remain connected for a longest duration of time. According to this method, a multicast receiver waits after receiving the first Join-Query for an appropriate amount of time so as to examine the stability all the possible routes and select the most stable one for propagation of a Join-Reply.

2.5.3 Improvements on ODMRP

ODMRP has undergone some major improvements since it was first specified as an Internet draft. The Internet draft of ODMRP [127], which appeared in 1998, contains what may be referred to as a basic (or original) specification of the protocol. The broadcast method in the original version of ODMRP is solely blind flooding without an option for an efficient method. Another important part of ODMRP that has no alternatives in the original version of

ODMRP is the multicast tree construction method which is solely based on the shortest path selection criterion.

In the ODMRP's Internet draft version of the year 2000 [128], a mobility prediction technique based on GPS was included. The technique makes use of location and movement information of nodes. There are two main areas where the mobility prediction technique is used: in adapting route refresh intervals and in multicast tree construction. The technique is able to predict the duration of time when any two nodes will remain connected and hence it is possible to determine the right interval for invoking Join-Queries aiming at refreshing route information. The 2000 version of ODMRP's Internet draft contains an alternative multicast tree construction method which uses the mobility prediction technique to select routes based on stability.

An attempt to address the drawbacks of blind flooding through passive clustering appears in the Internet draft of 2002 [9]. The inclusion of passive clustering provides an efficient broadcast method in terms of minimising broadcast redundancy. The 2002 version of ODMRP's Internet draft is considered to be the latest version in this thesis. The aforementioned stages of ODMRP's development (i.e. the three versions of the Internet draft) provide evidence for one to believe that ODMRP is a well developed multicast protocol, and this is perhaps why it is one of the popular multicast protocols of MANETs.

Improvements on ODMRP have not been conducted solely by its owners but also by other researchers. The main areas of improvement include reduction of broadcast redundancy during the route discovery process, detection of link failure and local recovery, multicast tree construction based on route stability with respect to residual energy of nodes, and the ability to guarantee quality of service.

H. Dhillon and H. Q. Ngo [129] have attempted to minimise ODMRP's broadcast redundancy by consolidation of Join-Query packets. In a network environment with multiple multicast sources, an intermediate node consolidates Join-Queries from a number of sources into one Join-Query which is then broadcast to downstream neighbours. The consolidated Join-Query has entries for identification information of all the sources that make up the joint Join-Query. Upon receiving the consolidated Join-Query, a multicast receiver node builds a Join-Reply based on all the sources contained in the Join-Query and transmits it to upstream neighbours of the routes to the sources. Another effort to reduce ODMRP's broadcast

redundancy was proposed by Z. Yao et al. [130] who employed a technique that uses only a subset of neighbours as multipoint relays. Only multipoint relays are allowed to rebroadcast packets. The method proposed by H. Peng et al. [131] attempts to reduce ODMRP's broadcast redundancy through deciding whether to rebroadcast a Join-Query or not based on the relative distance between a node v and each one of its neighbours. A packet is rebroadcast by a neighbour of node v , say node w , if the distance between node v and w is long enough so that the additional coverage area resulting from the retransmission is large enough.

Upon realising that ODMRP does not address a link failure during data transmission until the next Join-Query is issued to refresh routing information, A. Ganguli et al. [132] proposed a local recovery mechanism to provide an alternative route for continuation of the data transmission. The idea behind the mechanism is that when a receiver finds that it does not receive data from a source for a while, it must probe the network and try to find an alternative route through which it can start to receive the packets again. On detecting that it is taking too long to receive the next data packet, a multicast receiver initiates a reconnection process which may eventually yield an alternative path at the point of failure. M. Naderan-Tahan et al. [133] proposed a local recovery mechanism for ODMRP similar to the one developed by A. Ganguli et al. [132]. The main difference between the two methods is that while the mechanism by A. Ganguli et al. [132] detects a link break based on the knowledge of time intervals between data packets that have to be received, a method by M. Naderan-Tahan et al. [133] detects a link failure based on the knowledge of neighbour nodes. The method by Naderan-Tahan et al. [133] specifies that a particular type of node, for example a source node, must have a minimum number of certain types of neighbours, such as forwarding nodes and receivers, for the node to be attached to a mesh. Failure to meet the minimum number requirement triggers a reconnection process.

H. Ying-xin and J. Yu-feng [134] proposed a method for improving stability of multicast routes in ODMRP with respect to the residual energy of nodes. The method selects only nodes that have enough residual energy to be forwarding nodes. Another research effort that attempts to improve stability of multicast routes in ODMRP from the viewpoint of residual energy is presented by S. J. Begdillo et al. [135]. However, the method of S. J. Begdillo et al. [135] seems to be more advanced than that of H. Ying-xin and J. Yu-feng [134] since it selects stable routes based on a computed weighted value which takes residual energy, delay and route expiration time into consideration.

The need to use ODMRP to deliver multimedia traffic motivated Y. Yao et al. [136] to work on how ODMRP can offer quality of service. The support for quality of service proposed by Y. Yao et al. [136] evaluates available bandwidth ahead of packet delivery, and builds multicast routes that guarantee availability of sufficient bandwidth. Another work on the provision of quality of service in ODMRP is reported by A. Darehshoorzadeh et al. [137] who focussed on the unicast mode of operation of ODMRP. The method proposed by A. Darehshoorzadeh et al. [137] reserves bandwidth with the help of admission control.

2.5.4 Why ODMRP is a reference protocol

Before presenting reasons for using ODMRP in this thesis as a reference multicast routing protocol, let us have a look at some of the research work available in the literature in which ODMRP is used as a reference protocol. A multicast zone routing protocol developed by X. Zhang and L. Jacob [48] used ODMRP as its reference protocol. A performance comparison between the two protocols was based on the packet delivery ratio and control overhead. Another protocol that used ODMRP for performance comparison is a hybrid multicast routing protocol developed by C. K. Chen et al. [49]. Performance metrics used in the comparison between the two protocols include packet delivery ratio, control overhead and end-to-end delay. ODMRP is also used for reference purposes in a hybrid multicast routing protocol proposed by J. Biswas et al. [46]. A performance comparison between the protocol by J. Biswas et al. [46] and ODMRP was based on the packet delivery ratio, number of data packets transmitted per data packet received, number of control bytes transmitted per data byte received and total number of packets transmitted per data packet received.

Some work on the provision of quality of service by multicast routing has also used ODMRP for purposes of performance comparison. A multicast routing protocol equipped with capabilities for support of quality of service proposed by H. Tebbe and A. J. Kassler [138] uses ODMRP for performance comparison. Average delay, packet delivery ratio and control overhead are used as metrics for the performance comparison between the two protocols. Another multicast routing protocol dealing with quality of service that uses ODMRP as its reference protocol is the one developed by J. Biswas and S. K. Nandy [139]. Performance metrics used in the comparison of the two protocols include packet delivery ratio, number of control bytes transmitted per data byte delivered and number of all packets transmitted per data packet delivered.

The number of research articles in which ODMRP appears as a protocol for performance comparison is much more than the five research papers presented in this section. The review of the five papers done in this section is only intended to provide evidence of the existence of research work which has relied on ODMRP for purposes of performance comparison.

While researchers who use ODMRP as their reference protocol have their own specific reasons, the reasons for adopting ODMRP as a reference protocol in this research are as follows:

- 1) Being an on-demand routing protocol implies that ODMRP minimises control overhead since it is invoked only when there is data to be transmitted.
- 2) ODMRP does not generate excessive control overhead as mobility of nodes increases since no control packets are triggered by link breaks.
- 3) Redundant paths provided by the mesh structure of ODMRP provide alternative paths for data transmission when links break thereby making ODMRP robust to mobility.
- 4) Based on the first three reasons, it is arguably the best multipurpose multicast routing protocol that can be used in all the three topologies of ELC-MANETs.
- 5) It is a highly refined multicast protocol as evidenced by the work in the three versions of its Internet draft specification covered in this section.
- 6) It is a popular multicast routing protocol since many researchers have used it for performance comparison.
- 7) Errors introduced by poor implementation of a reference protocol in a network simulator are avoided since ODMRP is already implemented by the owners in GloMoSim-2.03 [5] used in this research.

2.6 Research Problem

As mentioned in chapter 1, this research proposes multicasting as the preferred routing method in the ELC-MANET. The success of deploying the proposed ELC-MANET multicasting mainly depends on how it is prepared to meet service requirements as expected by the intended users. Among other expectations, the users of the ELC-MANET are likely to expect a multicast service which is always reliable. Thus, the service should be available whenever it is needed, and when it starts operating, it should continue working properly until its proper termination time.

Ensuring service reliability in the ELC-MANET multicasting is extremely difficult due to some of the multicast challenges outlined in the opening paragraphs of subsection 2.2.2. Some of the multicast challenges that are related to service reliability are topology changes, limited bandwidth, limited supply of energy, and QoS. Frequent and rapid changes in network topology lead to frequent connectivity breakages which in turn negatively affect service reliability. The availability of limited bandwidth in a MANET implies that only a limited number of service requests can be accepted at a time for the network to guarantee service reliability. Depletion of battery energy which power MANET devices leads to a node failure which in turn causes a connectivity link break. Hence, fast depletion of MANET device's battery energy can lead to frequent breakages of connectivity. Service reliability is a component of QoS. Guaranteeing QoS is currently a major challenge in MANETs due to factors such as rapid and unpredictable topology changes, limited supply of energy for the devices and limited bandwidth.

This research is interested in ensuring service reliability in the ELC-MANET through efficient use of the limited battery energy which power MANET devices. As one way of prolonging battery life of a MANET device which in turn ensures service reliability, it is important that a routing protocol should try as much as possible to minimise maintenance and transmission of control overhead. Transmission of control overhead requires energy just like transmission of data packets. When a routing protocol generates and transmits excessive control overhead it implies that energy is wasted on transmissions which can be avoided. One of the major areas of a MANET routing protocol which easily generates excessive control overhead is that of route discovery and maintenance. In particular, the excessive control overhead problem considered in this research associated with route discovery and maintenance is the broadcast redundancy resulting from broadcasting or flooding operations [6, 36]. This research considers broadcast redundancy to be a major component of the cost of route discovery and maintenance.

2.6.1 Research problem definition

The ELC-MANET is expected to offer multimedia services. The demand for energy for transmission of multimedia services is relatively higher compared to the energy demand for non-multimedia services, and therefore the ELC-MANET requires a much more efficient use of the limited battery energy of a MANET device than an ordinary MANET. The literature review of the existing multicast routing protocols presented in subsection 2.2.2 shows that

most of the current multicast routing protocols are not efficient enough in terms of resource usage. In particular, the existing multicast routing protocols have not minimised the broadcast redundancy associated with route discovery to a level which can be regarded as low enough for efficient operation in the ELC-MANET. Hence, the broadcast redundancy that occurs during a route discovery operation of a MANET multicast routing protocol is the problem of this research undertaking.

Apart from the huge energy demand in the ELC-MANET, another reason that makes the broadcast redundancy of control packets in a multicast routing protocol to be an important problem of this research is related to the likelihood of having multiple simultaneous sources of traffic in the ELC-MANET due to the highly interactive nature of the environment. A multicast routing protocol operating in the ELC-MANET is required to generate join request packets (i.e. control packets) periodically, and that the generation frequency is required to be high enough to cope with node mobility. In the presence of multiple sources of traffic as is likely to be the case in the ELC-MANET, the broadcast redundancy can lead to the broadcast storm problem [8] which can significantly affect utilisation of network resources. Therefore, even though transmission of control packets of a routing protocol in any network usually uses a negligibly small amount of network resources compared to the transmission of data packets, which utilise most of the network resources; in the ELC-MANET, the situation has the possibility of being different.

2.6.2 Existing efforts that attempt to address the problem

Some of the current multicast routing protocols possess what are considered in this thesis as attempts to address the research problem of this research. One of such protocols is PLBM [25] which uses a pruning technique to select only a subset of neighbours of a sender node to act as forward nodes of a broadcast packet. PLBM's technique is basically similar to the advanced pruning techniques of forward nodes proposed in [7, 36, 37]. Being a tree-based protocol, PLBM is not a suitable candidate for the ELC-MANET since it is likely to offer low resilience against frequent topology changes which are expected in certain scenarios of the ELC-MANET. TMP-MPRs [28] is also another tree-based multicast routing protocol which has attempted to reduce broadcast redundancy but it is also not a suitable candidate protocol for the ELC-MANET based on the same reason as that of PLBM.

ODMRP, a mesh-based protocol, has seriously considered broadcast redundancy associated with route discovery and maintenance. The technique used by ODMRP to reduce broadcast redundancy is that of passive clustering. As pointed out in the review of ODMRP, passive clustering can easily lead to low performance of a protocol in scenarios of high network connectivity (refer to the review of ODMRP). Hence, ODMRP is not suitable for some scenarios of the ELC-MANET. Other mesh-based multicast routing protocols which have attempted to address broadcast redundancy include NSMP [38] and DCMP [41]. However, this research considers the pruning techniques of forward nodes used by NSMP and DCMP to be less advanced than those proposed in [7, 39, 40].

Some of the hybrid multicast routing protocols have also made attempts to address broadcast redundancy. MCEDAR [45] is an example of such protocols. MCEDAR uses a clustering approach in which forwarding of packets is only handled by core nodes. MCEDAR is not a suitable candidate for the ELC-MANET mainly due to its likelihood of generating excessive control overhead resulting from frequent changing of core nodes in situations of high node mobility. MZRP [48] has also made attempts to address control overhead by preventing source nodes from flooding join requests to an entire network except for the group leader. Control overhead in MZRP can be improved further by minimising broadcast redundancy resulting from flooding operations performed by the group leader.

AMRP [51] is an adaptive multicast routing protocol which has made attempts to minimise control overhead, including broadcast redundancy, by allowing only a subset of source nodes to broadcast join requests at a time. However, AMRP can reduce further its control overhead by using advanced pruning techniques of forward nodes proposed in [7, 36, 37] to minimise broadcast redundancy associated with broadcast operations performed by the source nodes that are allowed to broadcast. HIMPRO [55] addresses broadcast redundancy by using a cluster-based approach in which only cluster-heads and gateway nodes are allowed to forward broadcast packets. It is possible that the clustering technique used by HIMPRO to minimise broadcast redundancy can be improved by incorporating advanced cluster-based pruning techniques proposed in [43, 44]. The third adaptive protocol reviewed in this research which has explicitly addressed broadcast redundancy is OPHMR [56]. OPHMR selects a subset of neighbours of a sender node to act as forward nodes of a broadcast packet. Incorporation of the advanced pruning techniques of forward nodes proposed in [7, 36, 37] in

the forward node selection process of OPHMR can help to improve further the broadcast redundancy in OPHMR.

2.6.3 Motivation for this research

The literature review of this research shows that none of the existing multicast routing protocols is efficient enough in terms of resource usage to be regarded as a reasonable solution to the problem of this research. The expected solution is required to perform efficiently in ELC-MANET situations of high node mobility which usually happens when students are outside their classrooms as well as in situations of nearly no mobility which normally occurs when students are fully engaged in a lesson activity inside their classrooms. The solution is also expected to perform efficiently in ELC-MANET environments of high wireless connectivity which occurs when students are in a classroom as well as in situations of low wireless connectivity which exists when students are scattered outside their classrooms during break time such as lunch time. These requirements of the expected solution necessitate the development of a new multicast routing protocol. This research is therefore aimed at developing a new multicast routing protocol as a solution to the research problem.

2.7 Summary

The primary aim of this chapter is to present previous research work and then eventually formulate the problem of this research. The chapter has started by presenting a review of some of the work reported in the literature on the use of ICT in E-Learning. One of such research efforts is the work by C. Chang and J. Sheu [13] who defined an eSchoolbag system containing learning materials such as electronic books, notebooks and a calculator. The interest of this research is in the network layer of the OSI reference model unlike most of the work in the literature on the use of MANETs in E-Learning which is on the application layer.

This research focuses on multicasting in the ELC-MANET. The chapter presents a number of challenges which affect multicasting in MANETs. Some of the multicast challenges are unpredictable and rapid topology changes, limited availability of bandwidth, limited supply of energy for MANET devices and QoS.

The multicast challenges have led to the development of various types of multicast routing protocols. Each of such types of protocols is aimed at addressing one or more of the multicast

challenges. A detailed review of four major types of multicast routing protocols is presented: tree-based, mesh-based, hybrid and adaptive multicast routing protocols. Examples of the reviewed tree-based multicast routing protocols are AMRIS, MAODV and ABAM. CAMP, FGMP and ODMRP are some of the reviewed mesh-based protocols. Hybrid protocols reviewed include AMRoute, MCEDAR and MZRP, while examples of the reviewed adaptive protocols are AMRP, ASTM and ACMRP.

This chapter has also presented a review of the current broadcasting methods in MANETs. The broadcasting methods are organised into four categories: clustering without CDS, CDS without clustering, clustering combined with CDS, and other methods. The broadcasting methods were reviewed with the intention identifying suitable ones for the solution to the research problem (covered in chapter 4).

A literature review of the existing clustering methods for MANETs is also covered in this section with the aim of identifying suitable clustering methods for CTRDM (covered in chapter 4). The review presents two types of clustering methods, namely passive and active clustering. The observation in the review is that most of the clustering techniques in the literature are on active clustering. Most of the recent clustering techniques select cluster-heads based on the weight (or the score) of a node which is computed by considering factors such as the number of neighbours of a node, the level of node mobility and the remaining battery energy.

The chapter presents a detailed analysis of ODMRP. Areas covered include the operation of ODMRP, focussing on the route request phase and the route reply phase. ODMRP is used as a reference protocol in this research. Hence, the section on ODMRP has included a brief literature review of research work in which ODMRP is also used for reference purposes. The analysis of ODMRP ends by giving reasons for using ODMRP as a reference protocol in this thesis.

The actual formulation of the research problem is presented as the last section of the chapter. It is observed that the current multicast routing protocols have not seriously looked into the reduction of broadcast redundancy associated with route discovery and maintenance. The idea of using MANETs in E-Learning proposed in this research demands that a MANET multicast routing protocol to be used in the ELC environment should use the battery energy that powers MANET devices as efficiently as possible in order to reliably deliver E-Learning

multimedia services. This efficiency requirement of a multicast routing protocol combined with the fact that broadcast redundancy can cause the broadcast storm problem make broadcast redundancy to be an important problem in the context of the ELC-MANET. Hence, the problem of this research is the broadcast redundancy associated with a route discovery operation of a MANET multicast routing protocol.

University of Cape Town

CHAPTER 3

RECOGNITION OF ELC-MANET PHASES

3.1 Introduction

The approach taken to address the problem of this research begins with the analysis of the organisation of nodes in the ELC-MANET. As mentioned in chapter 1, the observation in this research is that the ELC-MANET has the tendency of transforming itself from one distinct topology to another as a result of the students' activities. The ELC-matrix topology is formed when students are seated in a classroom during a lesson activity. When students sit in discussion groups, they form the ELC-cluster-based topology. The ELC-random topology is created when students are dispersed to various locations but within the campus of their school during break time such as lunch time.

Each one of the three ELC-MANET topologies (or phases) has its own characteristics which impose certain conditions on a multicast routing protocol for efficient operation in the topology. In general, nodes in the ELC-matrix topology are almost stationary and have high wireless connectivity because they are confined in a classroom which has a small geographical area. In the ELC-cluster-based topology, the nodes are also almost stationary and have high wireless connectivity within clusters and low connectivity between clusters. The ELC-random topology is characterised by high mobility of nodes. The nodes in the ELC-random topology have no definite pattern of connectivity. The solution to the problem of this research is required to taken into consideration all the characteristics of the ELC-MANET.

The organisation of the rest of this chapter is as follows. Section 3.2 covers the requirements of the solution to the research problem. The proposed mechanism for detecting dynamically the phases (or topologies) of the ELC-MANET is covered in section 3.3. A summary presented in section 3.4 concludes this chapter.

3.2 Requirements of the Solution

Based on the hypothesis of this research presented in chapter 1, the main requirement of the solution to the problem of this research undertaking is to have the capabilities of using a CDS

for forwarding of control packets which is as close as possible to the MCDS of the network in which the protocol (i.e. the solution) operates. Note that the MCDS requirement is about efficient utilisation of network resources in which only a subset of nodes (theoretically nodes belonging to the MCDS) is allowed to take part in the relaying of a broadcast message. In addition to the MCDS requirement, the solution is also expected to meet the traditional requirements for purposes of enhancing efficiency and reliability. Some of the traditional requirements which also include some aspects of the MCDS requirement are high efficiency in terms of resource usage, robustness, scalability, simplicity, and adaptability. Each of these requirements is explained in the subsequent subsections.

3.2.1 Efficiency

High efficiency especially in the utilisation of the limited energy supply available in the ELC-MANET is one of the requirements to be met by the solution of this research. As a starting point, the expected multicast routing protocol is required to use a tree structure for delivery of multicast packets. The resulting multicast tree should always provide optimal paths (i.e. minimum distances, most stable routes, etc) between a source node and multicast receivers.

Creation and maintenance of the multicast tree should also be as efficient as possible. A multicast tree which is constructed by first creating a spanning tree based on topology information of an entire network is not suitable in the ELC-MANET since such a multicast tree is likely to consume a lot of energy through intensive computations, storage of the results, and transmission of data needed in the computations. A good example of such an energy demanding approach of constructing and maintaining a multicast tree is the Steiner tree method [140, 141].

The ELC-MANET has scenarios of high node mobility which can be associated with frequent topology changes that usually lead to breakages in connectivity links. A multicast routing protocol may reconfigure its multicast tree so as to address link failures resulting from the topology changes. When node mobility is high, the frequency of performing such multicast tree reconfigurations may also be high. Therefore, the expected multicast routing protocol for the ELC-MANET is required to generate as little control overhead as possible during times of frequent multicast tree reconfigurations.

3.2.2 Robustness

Having a multicast routing protocol which is purely based on a tree network structure for transmission of multicast packets in the ELC-MANET can easily result in unreliable delivery of services especially in the ELC-random topology environment. When a link fails during data transmission, packets can easily be lost at the point of failure since there is only one path between each source-receiver pair of nodes in a tree structure. The expected multicast routing protocol should therefore use a mesh-based structure in addition to the tree structure for transmission of multicast traffic. A mesh-based structure provides alternative paths for continuation of data transmission at a point of link failure.

Maintenance of a mesh network structure is more costly than that of a tree structure. The high maintenance costs of a mesh structure come from computations of many routes, storage of a large amount of routing information and transmission of data needed for route computations. The expected multicast routing protocol is required to minimise the generation and maintenance of routing information for a mesh structure as much as possible.

One way of minimising costs associated with maintenance of a mesh structure is by using an on-demand approach for route discovery and maintenance rather than a proactive approach. The main advantage of the on-demand approach is that it saves network resources (i.e. bandwidth, energy, storage space, computation time, etc) since routing information is generated only when a source node has packets to transmit and maintained only during the time of the packet transmission. However, the on-demand method has the problem of the setup latency which may not be acceptable for some time-sensitive applications. If the expected multicast routing protocol decides to use the on-demand approach, it should ensure that the setup latency is reduced as much as possible.

3.2.3 Scalability

Multicast groups in the ELC-MANET can range from the size of 2 members to a very large size of over 500 members. It is possible for a teacher to multicast information to about 5 or 10 students during a lesson activity in a classroom situation. This may be regarded as a case of a small multicast group. The expected multicast routing protocol is required to perform efficiently in terms of resource usage when a multicast group size is small.

It can happen that the head teacher of a school may want to multicast information to almost every student in the school. This is an example of a situation when a multicast group can exceed 500 members. Most multicast routing protocols available in the literature begin to show signs of inefficiency when the size of a network (or a multicast group) and/or the network traffic becomes very large. The expected multicast routing protocol for the ELC-MANET is expected to remain efficient as the size and/or the network traffic of a multicast group increases.

3.2.4 Simplicity

A more complex design of a multicast routing protocol is likely to demand more network resources than a less complex design. This research prefers that the complexity of the design of the ELC-MANET multicast routing protocol should be minimised as much as possible. Another reason for preferring simplicity in the design of the ELC-MANET multicast routing protocol is that this research believes that a more complex protocol has a higher probability of making errors in its operation than a less complex protocol.

3.2.5 Adaptability

In order for the expected multicast routing protocol to meet most of the aforementioned requirements while operating in any of the three phases of the ELC-MANET, it is necessary that the protocol should have an adaptive design approach. To keep the design of the expected protocol simple, this research prefers that the protocol should have three distinct modes of operation which should correspond to the three phases (or topologies) of the ELC-MANET.

To have a protocol with three modes of operation in accordance with the three phases of the ELC-MANET demands that the protocol should have a mechanism capable of recognising dynamically the three topologies of the ELC-MANET. The task of detecting dynamically the three ELC topologies is a challenge especially with respect to the accuracy of the mechanism used and costs of the computations involved. The goal of this research is to use a mechanism which has a high degree of accuracy in detecting the topologies, while incurring minimal computational costs.

3.3 Detection of the ELC-MANET Topologies

There is limited literature on mechanisms which could be relevant in detecting the three phases of ELC MANETs. A mechanism based on the Global Positioning System (GPS) is a possible candidate but it cannot be used in this research as GPS performs poorly inside buildings [142]. The location fingerprinting technique is a possible mechanism to use [143, 144]. Even though the fingerprinting mechanism appears feasible, its large computational costs in constructing a radio map make it unsuitable to use in the power-constrained devices of ELC MANETs.

The research proposes the use of “the number of neighbours” of a node as reliable information that can lead to the detection of the three topologies. Since the three ELC MANET topologies have different patterns of node placement, the standard deviation of the number of neighbours can be used to identify the topologies. The first reason for using the number of neighbours is that such data are likely to be more accurate than GPS-based data in indoor environments. Secondly, the resulting mechanism is likely to have less computational costs than mechanisms based on signal strengths.

3.3.1 Hypotheses and the formula of the proposed mechanism

The reasoning upon which the proposed mechanism for identifying the topologies of the ELC-MANET is based is expressed in the form of two hypotheses, which are as follows:

- The high uniformity in the node distribution and the geometric symmetry of the matrix topology ensures, with high probability, that the standard deviation (*stdev*) of the number of neighbours of nodes is lower than in a cluster-based and random topology with the same number of nodes.
- As an extension to the first, a random topology is likely to have a higher *stdev* than a cluster-based topology with the same number of nodes.

The preferred formula for computing the standard deviation of the number of neighbours of nodes in the ELC-MANET for the purpose of detecting the phases of the ELC-MANET is that of the sample standard deviation, which is given in the following equation:

$$stdev = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (3.1)$$

In Equation (3.1), N is the total number of nodes in a network or the network size, x_i represents the number of neighbours of node i , and \bar{x} is the mean value of the number of neighbour nodes in the network obtained by using the following expression:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.2)$$

The reason for preferring the sample standard deviation (i.e. Equation (3.1)) in this research is that the ELC-MANET has the possibility of having very small networks of possibly fewer than 10 nodes whose standard deviation of the number of neighbours of nodes is significantly larger when obtained using the formula of the sample standard deviation than the value obtained using the formula of the population standard deviation. The understanding in this research is that the use of large values of the standard deviation of the number of neighbours of nodes has a higher probability of yielding accurate decisions in detecting the ELC topologies than the use of small values with respect to the presence of round-off errors and the need to create non-overlapping subsets of the values to represent the topologies.

3.3.2 Computation of *stdev* by an online algorithm

This research proposes the use of an online algorithm to compute Equation (3.1) in the expected multicast routing protocol for the ELC-MANET. Each multicast group is expected to choose only one node to be responsible for running the online algorithm. Each of the rest of the nodes in a multicast group is required to submit the number of its neighbours to the node responsible for computing *stdev* (i.e. Equation (3.1)). The online algorithm is expected to compute *stdev* each time it receives data from a multicast member node. Therefore, the online algorithm is able to obtain *stdev* for the nodes that have submitted their data so far without waiting for all the nodes in a multicast group to submit data about their neighbours. However, the expected multicast routing protocol is required to ensure that the *stdev* to be used for decision making should be the one resulting from the submission of data from almost all the members in a multicast group. This requirement is aimed at increasing the probability of identifying the topologies correctly.

The recommended online computing algorithm for *stdev* is shown in Figure 3.1. The shown algorithm is considered in the literature as the basic algorithm for online computation of standard deviation [145]. When the algorithm is used to compute a large data set, the error in

the final result obtained can be quite significant. The error in the results comes mainly from the round-off errors in the division operations.

An attempt to address the error in the computed results has been made by ensuring that the reference values (i. e. values used for comparison purposes when making decisions) of the standard deviation are also obtained by the same online computing algorithm (i.e. Figure 3.1). The thinking behind this approach is that what matters most is not that a value should have no errors but that the computation should consistently give the same value for the same network topology and size under different networking situations in terms of factors such as place, purpose, and devices used.

Online algorithm for computing the standard deviation:

- 1) $n = 0$, $\text{sum1} = 0$, $\text{sum2} = 0$.
 - 2) for each x received:
 - 3) $n = n + 1$
 - 4) $\text{sum1} = \text{sum1} + x$
 - 5) $\text{sum2} = \text{sum2} + (x \times x)$
 - 6) $\text{mean} = \text{sum1} / n$
 - 7) if $n > 1$ then
 - 8) $\text{variance} = (\text{sum2} - \text{sum1} \times \text{mean}) / (n - 1)$
 - 9) else
 - 10) $\text{variance} = 0$
 - 11) endif
 - 12) $\text{standard deviation} = \sqrt{\text{variance}}$
-

Figure 3.1 Online algorithm for computing the standard deviation [145]

The reasoning behind the attempt of dealing with the error associated with the online algorithm of Figure 3.1 with respect to the reference values is expressed in the form of the following hypothesis:

- Any two values of the standard deviation of the number of neighbours of nodes of the same ELC-MANET topology obtained under different networking situations are likely to be equal or very close to each other provided that the method (or methods)

used to compute the values is capable of generating the same error (any type of error) for the same input values.

As already mentioned in the second paragraph of this subsection, the online algorithm for the standard deviation shown in Figure 3.1 is regarded as a basic method in the literature. Advanced methods which minimise round-off errors for online computation of the standard deviation exist in the literature. However, the basic method is preferred in this research for two reasons. First, the presence of the errors does not negatively affect the usefulness of the computed results. Second, the design of the expected multicast routing protocol is aimed at minimising control overhead and therefore using an advanced method which generates unjustifiably more computational costs defeats the aim of the expected protocol.

3.4 Summary

This chapter marks the beginning of the solution to the problem of this research undertaking. The first to be presented in this chapter, which builds a foundation for the solution to the research problem, is a brief description of the concept of phases of the ELC-MANET. The observation in this research is that the ELC-MANET is capable of undergoing a transformation from one distinct topology (or phase) to another depending on the activities of students. The three topologies of the ELC-MANET are the ELC-matrix topology, the ELC-cluster-based topology and the ELC-random topology.

The characteristics of the ELC-topologies require certain conditions to be met by the expected solution to the research problem. Among other requirements, the solution to the research problem, which has to be a multicast routing protocol, is supposed to seriously consider efficiency in terms of resource usage, robustness, scalability, simplicity and adaptability.

The expected multicast routing protocol is proposed to have three modes of operation corresponding to the three topologies of the ELC-MANET. This proposal requires that the protocol should have a mechanism of dynamically detecting the three topologies of the ELC-MANET. Hence, the standard deviation of the number of neighbours of nodes in a network is proposed as reliable data that can lead to identifying dynamically each of the three ELC-topologies.

The research proposes the use of an online algorithm for computation of the standard deviation to be used for detecting ELC-topologies. The algorithm is capable of computing the standard deviation for any number of nodes that have submitted their data without waiting for all the nodes in a multicast group to submit their data. The proposed online algorithm is considered as a basic method in the literature which usually generates a significant round-off error in the output when computing a large data set. However, the presence of the error cannot negatively affect the use of the algorithm in the expected multicast routing protocol because what matters most is that the algorithm should consistently give the same result for a network of the same size and topology while operating in different environments in terms of factors such as time, place and devices.

University of Cape Town

CHAPTER 4

CHARACTERISATION OF ELC-MANET PHASES

4.1 Introduction

This chapter presents the characteristics of the three distinct topologies of the ELC-MANET and also presents route discovery methods of the proposed multicast routing protocol for the ELC-MANET. The details for each ELC-MANET topology include the activities of students that give rise to the topology, the expected coverage area of the network (or topology), and the presence of mobility of nodes in the network. This research is particularly interested in the connectivity characteristics of the ELC-MANET phases because they have a direct effect on the performance of a MANET multicast routing protocol.

The research problem is addressed by proposing the use of an adaptive multicast routing protocol called Topology-Aware Multicast Routing Protocol (TAPMRP). TAPMRP is made up of three route discovery methods. Each one of the three route discovery methods is intended to serve its own ELC-MANET phase. The proposed route discovery method for each ELC-MANET phase is presented in this chapter following the characteristics of the phase. The main factors that were considered in the development of each route discovery method were the network connectivity characteristics and the level of mobility of nodes. This chapter only presents overview descriptions of the route discovery methods while the design specifications of the methods are covered in chapter 5.

The organisation of this chapter is as follows. Section 4.2 covers the ELC-matrix topology and its proposed route discovery method of TAPMRP. The ELC-cluster-based topology and its proposed route discovery method of TAPMRP are covered in section 4.3. Section 4.3 also covers a proposed clustering platform used by the route discovery method of TAPMRP. The clustering platform covers two novel techniques for load balancing, namely secondary cluster-heads and the node redistribution parameter. Section 4.4 presents the ELC-random topology alongside its proposed route discovery method of TAPMRP. The chapter ends with a summary in section 4.5.

4.2 ELC-Matrix Topology

4.2.1 Characteristics of the ELC-matrix topology

The ELC-matrix topology is one of the three ELC-MANET topologies considered in this research. Only the ELC-matrix topology out of the three ELC-MANET topologies is likely to occur in indoor environments only. Students create the ELC-matrix topology when they sit in rows inside a classroom with each student assumed to hold only one MANET-enabled device. ELC-MANET devices are heterogeneous in nature, and therefore are expected to have different networking and computing capabilities such as transmission range, battery energy capacity, memory capacity, and instruction execution rate of the microprocessor. Examples of ELC-MANET devices are laptop computers, notebook computers and personal digital assistants (PDAs).

In spite of having different transmission ranges, ELC-MANET devices forming an ELC-matrix topology are likely to have very high connectivity among them. Thus, compared to the other two ELC-MANET topologies, the ELC-matrix topology is characterised by higher connectivity. The high connectivity in the ELC-matrix topology is as a result of the devices being within one room such that the geographic distances between the devices is much shorter than the transmission ranges of the devices. The default transmission range for the devices used in GloMoSim-2.03 is 377m based on the settings shown in Table 4.1 [146]; this may provide a rough idea about the transmission range of ELC-MANET devices. Given the heterogeneous nature of the ELC-MANET devices, the smallest possible transmission range may be much less than 377m; but since the classroom is an open space without obstacles for radio signals, we expect the smallest possible transmission range to be long enough to create a topology of high connectivity inside the classroom.

Table 4.1 Settings of GloMoSim-2.03 for the transmission range of 377m

Parameter	Value
MAC protocol	IEEE802.11
Transmitting power	15dBm
Radio propagation pathloss model	Ground reflection (or two-ray)
Receiver sensitivity threshold	-81dBm

Modelling the ELC-matrix topology as an undirected unit disk graph $G(V, E)$, the high connectivity in the ELC-matrix topology implies that $G(V, E)$ contains large cliques approaching the size of $G(V, E)$. A clique is a subset $V' \subseteq V$ such that there is an edge between any two nodes in V' [147]. The presence of large cliques in the ELC-matrix topology has an effect on the performance of routing methods that can be used to operate in it. In particular, not all broadcast methods for route discovery can give optimum performance in the ELC-matrix topology. For example, blind flooding is likely to yield high broadcast redundancy. Another example is that using a cluster-based broadcast scheme is likely to perform poorly since without limiting the cluster-size, the clustering scheme is likely to produce a very large cluster plus possibly few small ones, while limiting the cluster-size is likely to yield unnecessary control overhead.

In addition to having minimal mobility of nodes and high wireless connectivity, the ELC-matrix topology environment is also characterised by having a uniform distribution of nodes. The matrix topology of nodes in the ELC-matrix topology is considered as a special case of the uniform distribution of nodes. In this research, the hypothesis is that generally the uniform distribution of nodes is likely to affect positively the performance of a MANET multicast routing protocol unlike a distribution of nodes which is not uniform.

4.2.2 Matrix Topology Route Discovery Method

The route discovery method of TAPMRP proposed in this research to serve the ELC-matrix topology is called a Matrix Topology Route Discovery Method (MTRDM). A route discovery process in a MANET multicast routing protocol is performed by a broadcasting algorithm. A number of broadcasting methods available in the literature were examined in order to identify the most suitable one for MTRDM. A broadcasting algorithm proposed by W. Lou and J. Wu [39] that uses the partial dominant pruning technique of forward nodes to minimise broadcast redundancy is recommended in this research for use as the broadcasting method for route discovery in MTRDM. Note that the recommended broadcasting method [39] was developed as a generic broadcasting scheme to be used for a number of purposes such as efficient flooding of data packets. The use of the broadcasting method in MTRDM implies that the method is being applied to achieve a specialised function.

From the viewpoint of the research hypothesis and postulate (1) and (2) presented under the research hypothesis in chapter 1, the adoption of the partial dominant pruning technique of

forward nodes by W. Lou and J. Wu [39] in the ELC-matrix topology is likely to yield a CDS that is very close to the MCDS of the topology due to the fact that the ELC-matrix topology has high connectivity. Note that the partial dominant pruning technique by W. Lou and J. Wu [39] is essentially a heuristic approach that tackles the MCDS problem. The contribution of this research from this section is the addition of another layer of a heuristic approach on top of the heuristic approach by W. Lou and J. Wu [39] by way of applying the pruning technique in an environment which has the potential of enhancing the reduction of the MCDS. Note that the potential to enhance the reduction of the MCDS comes from the high connectivity of the ELC-matrix topology environment.

To briefly explain the operation of the partial dominant pruning technique [39] we start by assuming that u is a sender node and v is a receiver node. We let $N(u)$ represent the neighbour set of u (including u), $N(N(u))$ represent the neighbour set of $N(u)$ (i.e. the set of nodes that are within two hops from u), and U represent the two-hop neighbour set of v that is yet to be covered by a broadcast process. The set U is given by $U = N(N(v)) - N(u) - N(v) - N(N(u) \cap N(v))$. The greedy set cover algorithm [7] shown in Figure 4.1 is then applied on U to obtain a forward node set $F(u, v)$ of a broadcast packet for node v .

Selection Process [7, 39]:

- 1) Let $F(u, v) = []$ (empty list), $Z = \emptyset$ (empty set), and $K = \cup S_i$ where $S_i = N(v_i) \cap U(u, v)$ for $v_i \in B(u, v)$.
 - 2) Find set S_i whose size is maximum in K . (In case of a tie, the one with the smallest ID i is selected.)
 - 3) $F(u, v) = F(u, v) \parallel v_k$, $Z = Z \cup S_i$, $K = K - S_i$, and $S_j = S_j - S_i$ for all $S_j \in K$.
 - 4) If $Z = U(u, v)$, exit; otherwise go to step 2.
-

Figure 4.1 Greedy set cover algorithm for $F(u, v)$ [7, 39]

In Figure 4.1, Z is a subset of $U(u, v)$ covered so far, S_i is the neighbour set of v_i in $U(u, v)$, K is the set of S_i , and $B(u, v) = N(v) - N(u)$. W. Lou and J. Wu [39] recommend that for the selection process of $F(u, v)$ (Figure 4.1) to terminate while being used by the partial dominant pruning algorithm, step 4 of the selection process should be changed to: If no new node is added to Z , exit; otherwise, go to step 2.

The partial dominant pruning technique is preferred in the route discovery process of MTRDM for the following reasons:

- It is more efficient than blind flooding and more effective than other pruning techniques, namely self pruning and dominant pruning [7, 40].
- It consumes less bandwidth than total dominant pruning [39] since it uses neighbour information available through hello messages only while total dominant pruning uses information from neighbours piggybacked in data packets in addition to information available through hello messages.
- Unlike probabilistic and counter-based broadcasting techniques which do not guarantee 100% network coverage, it is a deterministic broadcasting algorithm which ensures that a broadcast packet reaches every node in a network.
- It is likely to perform better in indoor environments associated with ELC-MANETs than location-aided broadcasting techniques based on GPS which performs poorly in such environments.
- Since it is intended for the ELC-matrix topology which is characterised by very high wireless connectivity, it is likely to perform better than cluster-based broadcasting techniques which can introduce unnecessary control overhead in such environments.

MTRDM is basically an improvement on the blind flooding method used by ODMRP. Like the blind flooding method, MTRDM consists of a route request phase and a route reply phase. MTRDM uses a Join-Query packet for the route request phase and a Join-Reply packet for the phase of route reply just like in the blind flooding method of ODMRP. A Join-Query is piggybacked in a broadcast data packet while a Join-Reply is propagated from a receiver to the source as a separate control packet. Propagation of Join-Replies to the source node selects routes that give minimum delay, resulting in the construction of a multicast tree based on the shortest path method. Once routes to receivers are established, a data transmission procedure is invoked and uses the constructed multicast tree to deliver data packets to the receivers. Both the established routes and their corresponding forwarding group nodes are refreshed periodically according to the specification of ODMRP [9]. A detailed design specification of MTRDM is presented in chapter 5.

4.2.3 Multicast tree construction method for MTRDM

A MANET multicast routing protocol uses a multicast tree usually rooted at the source node to deliver packets to all the receivers [148, 149]. The cost of transmission of a packet over the

multicast tree is expressed in terms of factors such as bandwidth consumption and delay which affect packet delivery in one way or another. In graph theory, the cost of transmission is represented as a weight of an edge in a multicast tree such that the problem of minimising the cost of transmission in the entire multicast tree is addressed by constructing a multicast tree that has the minimum total weight of the edges in the tree [150 - 152].

Two of the approaches used to construct a multicast tree are Steiner tree and shortest path tree method [7]. Given a weighted graph $G(V, E)$ and a set of nodes $D \subseteq V$, a minimum Steiner tree is such a tree which connects all the nodes in D using a subset of the edges in E that give the minimum total weight [153]. Constructing a minimum Steiner tree given an arbitrary graph is an NP-complete problem [140, 141]. Approximation or heuristic methods of the Steiner tree construction are available in the literature but in this research they are still considered to be computationally too inefficient for use in the energy-constrained devices of ELC-MANETs.

The shortest path tree construction ensures that the path between the source and each destination node offers minimum delay. Unlike the Steiner tree method, the shortest path method does not minimise the total cost of the multicast tree. The advantage of using the shortest path method is that it can be constructed in polynomial time and is easily implemented [7]. On the basis of reducing computational costs, this research proposes the shortest path tree method as the suitable multicast tree construction method for MTRDM and the other two route discovery methods of TAPMRP.

4.3 ELC-cluster-based Topology

4.3.1 Characteristics of the ELC-cluster-based topology

The second distinct phase of the ELC-MANET is the ELC-cluster-based topology. The ELC-cluster-based topology is formed by students when they sit in discussion groups and where each student is assumed to hold only one MANET-enabled device. Students' clusters can be located inside the classroom, outside the classroom, or some inside and some outside the classroom. Hence, the ELC-cluster-based topology can occupy a larger geographical area than that of the ELC-matrix topology which is confined to a classroom environment.

Wireless connectivity is expected to be very high within a cluster. In fact, every device within a particular cluster is expected to have a direct connectivity link with any other device

belonging to the same cluster. Since the ELC-cluster-based topology is expected to cover a relatively large geographical area, wireless connectivity between clusters is expected to be low. In such an environment, connectivity between clusters is established through the use of gateway nodes which are usually located at the edges of clusters. The ELC-cluster-based topology is, therefore, characterised by “spots” or “islands” of high connectivity “dotting” a particular area.

In terms of the unit disk graph theory [62], the clusters in the ELC-cluster-based topology form cliques. A node belonging to a particular clique is adjacent to any other node belonging to the same clique. It is reasonable to assume that every node in the ELC-cluster-based topology, except for the node representing the teacher, belongs to a clique (or a cluster). The minimum number of neighbours for a node v , except the node for the teacher, is given by the total number of nodes in a cluster to which node v belongs minus 1. Any gateway node of a particular clique has more neighbours than any non-gateway node belonging to the same clique.

4.3.2 Cluster-based Topology Route Discovery Method

The route discovery method of TAPMRP proposed to serve the ELC-cluster-based topology is called a Cluster-based Topology Route Discovery Method (CTRDM). Development of CTRDM, like that of the other route discovery methods of TAPMRP, is concentrated on minimising broadcast redundancy caused by a broadcast operation of join request packets. Hence, one of the tasks of the development process of CTRDM was to find an efficient broadcasting method suitable for the ELC-cluster-based topology. The recommended broadcasting method for route discovery in CTRDM is the one proposed by J. Wu and W. Lou [46], but with a major modification of replacing its method for the construction of a clustering platform with a method proposed in this thesis (covered in subsection 4.3.3).

The broadcasting method proposed by J. Wu and W. Lou [46] operates in the following way. When a broadcast process is initiated by a source node, the following steps are used:

- 1) If the source is not a cluster-head, it just sends the broadcast packet to its cluster-head.
- 2) When a cluster-head receives the broadcast packet for the first time, it chooses its forward node set to forward the packet to all its adjacent cluster-heads. The adjacent cluster-heads as well as the forward node set of the forwarding cluster-head are

piggybacked with the broadcast packet for the forwarding purposes. If the received packet is a duplicate one, the cluster-head does nothing.

- 3) When a non-cluster-head node receives the broadcast packet for the first time, and if it is in the forward node set, it relays the packet; otherwise, it does nothing.

The approach ensures that all the cluster-heads in the network will eventually receive the broadcast packet provided that the network is connected. A cluster-head selects a subset of gateway nodes as its forward nodes using the greedy set cover algorithm [7]. After all the cluster-heads have rebroadcast the packet in their clusters, all the nodes in the entire network are expected to receive the packet.

The reasons for adopting the broadcasting method developed by J. Wu and W. Lou [46] in CTRDM are as follows:

- Compared with many broadcast schemes including blind flooding, the method is more efficient in terms of reducing broadcast redundancy.
- The method guarantees high network coverage unlike probabilistic and counter-based methods.
- Being a cluster-based method, it naturally suits the ELC-cluster-based network which already has physical clusters.
- The method can perform well in indoor environments since it does not rely on GPS which performs poorly in such environments.

Like MTRDM, CTRDM is also an improvement on the blind flooding method used by ODMRP. CTRDM has a route request phase and a route reply phase just like any other route discovery method of an on-demand routing protocol of MANETs. In addition to performing the route discovery operation, CTRDM establishes and maintains clusters throughout the period in which it is allowed to operate. The clusters are refreshed or re-created at regular intervals so that cluster-heads which have become less suitable for the cluster-head role due to possible depletion of battery energy or dynamics in the network can handover the role of the cluster-head to nodes with better scores. A design specification of CTRDM is covered in chapter 5.

4.3.3 Proposed clustering algorithm for CTRDM

A clustering method based on the Distributed Clustering Algorithm (DCA) proposed by S. Basagni [106] is proposed as the clustering method for CTRDM. The proposed method adds the following to DCA:

- Load balancing
- A specification of the node weight, and
- Identification of bridge (or gateway) nodes

Load balancing is addressed at two levels in the proposed scheme: through the introduction of secondary cluster-heads, and through the use of a parameter hereinafter called a node redistribution parameter. The parameter attempts to reduce possible imbalances in the number of nodes per cluster by using a method which exploits the prior knowledge that the ELC-cluster-based topology for which CTRDM is proposed already has physical clusters. A heuristic algorithm of finding the minimum connected dominating set, called a marking process proposed by J. Wu and H. Li [72] (refer to subsection 2.3.2 of chapter 2), is used by CTRDM to identify bridge nodes in the underlying ELC-cluster-based topology. Note the proposed clustering algorithm for CTRDM is also referred to as the clustering platform in this thesis.

Several researchers have proposed various compositions for a node weight or score as it is called by other researchers [35, 96, 107, 111]. Well known factors that constitute a score upon which the selection of a cluster-head is based include the degree of connectivity (i.e. the number of neighbours of a node), residual battery energy and the degree of mobility. All these three well known factors for scoring nodes are considered in CTRDM. However, the nature of the ELC-cluster-based topology makes the use of the degree of connectivity unsuitable for cluster-head selection, and is therefore replaced by the node redistribution parameter.

Load balancing through secondary cluster-heads

Almost all weight-based clustering algorithms currently available in the literature include the degree of connectivity as one of the factors for selecting cluster-heads [35, 96, 107, 111]. A clustering algorithm solely based on node connectivity selects a node with the maximum number of neighbours to be a cluster-head, and any tie is broken by the unique node

identifiers. Experiments have shown that clustering algorithms based on the highest degree of node connectivity suffer from decreased throughput as the number of nodes in a cluster (i.e. cluster-member nodes) increases [35]. Y. Qin [36] has pointed out that as a cluster increases in size, the cluster-head becomes more overloaded since every ordinary node wants to route its traffic through the cluster-head to reach any destination outside the cluster. Hence, cluster-heads create bottlenecks in the network leading to reduction in throughput. The decreased throughput observation led to the introduction of the upper bound on the number of nodes in a cluster [35, 111]. The idea behind the restriction on the maximum number of nodes in a cluster is to achieve load balancing so that none of the cluster-heads are overloaded at any instant.

While the idea of placing the upper bound on the cluster size to achieve load balancing is indeed desirable, certain scenarios of mobile ad hoc networks such as the one shown in Figure 4.2 seem to be in disagreement with the idea, if the requirement of not having any two cluster-heads as neighbours is maintained in one-hop clustering schemes. Clustering of mobile ad hoc networks requires that no two cluster-heads can be neighbours; this is known as the independence property [106]. Maintaining the cluster-head independence property in clustering the network shown in Figure 4.2 with the maximum size of a cluster placed at 3, for example, leads to failure of clustering to be established. While the threshold value of 3 nodes per cluster is not realistic, it is not hard to see that a similar observation of clustering failure is possible with large networks with realistic upper bounds on the cluster size.

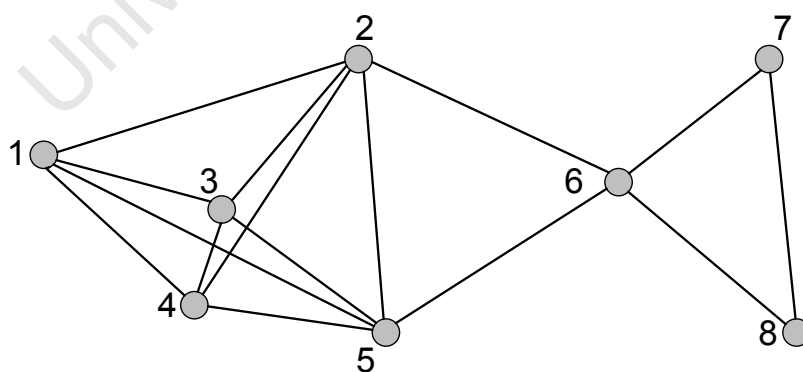


Figure 4.2 Example of a wireless ad hoc network

Let us now attempt to cluster the network in Figure 4.2 using the maximum value of 3 nodes for the cluster size. We assume that the degree of connectivity is the only criterion for cluster-

head selection. The following possible candidate clusters are obtained: cluster-1(1, 2, 3), cluster-2(4, 5, 6), and cluster-3(7, 8). In this clustering scenario, node 2 and 5 are cluster-heads which is a violation of the cluster-head independence property. A clustering algorithm may attempt to solve the problem by letting node 6 be the cluster-head of cluster-2, and when this happens node 2 in cluster-1 will have to handover the cluster-head role to node 3. However, the problem is not yet solved since node 8 in cluster-3 and node 6 in cluster-2 are neighbours, and if the algorithm attempts to use node 7 in cluster-3 as the cluster-head the same problem applies since node 7 and 6 are also neighbours. Clearly, the clustering algorithm will keep on looking for suitable candidates for the cluster-head role, leading to cluster instability. Hence, clustering the network in Figure 4.2 using 3 as the upper bound for the cluster-size fails since it leads to a non-converging clustering process.

To analyse the encountered clustering problem of the network example of Figure 4.2, let us begin by modelling the network as an undirected unit disk graph $G(V, E)$ [62]. The set of vertices V of $G(V, E)$ consists of 8 elements which represent the nodes in Figure 4.2. Figure 4.2 shows that $G(V, E)$ is a connected graph but is not a completely connected graph since there exists some pairs of nodes in V such as (2, 7) and (3, 6) which are not connected by an edge. In other words, some node pairs in V do not exist in E . However, a subgraph $G'(V', E')$ of $G(V, E)$ with $V' = \{1, 2, 3, 4, 5\}$ is completely connected. Thus, there is an edge between any two nodes in V' . In graph theory, the subset $V' \subseteq V$ is known as a clique [147, 154] and sometimes the subgraph $G'(V', E')$ induced by V' is also referred to as a clique (as the case is in this thesis).

A maximum clique is the largest clique among all the cliques in an undirected graph. A graph can have more than one maximum clique. Closely related to a maximum clique is a maximal clique which is defined as a clique whose vertices are not a subset of the vertices of a larger clique [147]. In the network of Figure 4.2, $V' = \{1, 2, 3, 4, 5\}$ is a maximal clique as well as a maximum clique, while $V'' = \{6, 7, 8\}$ is a maximal clique but not a maximum clique.

The argument here is that the clustering problem of the network in Figure 4.2 is due to the presence of cliques in $G(V, E)$. The observation is that the presence of cliques in a network imposes a restriction on the maximum size of a cluster that can be chosen without violating the cluster-head independence property. This observation leads to the following theorems

involving cliques, upper bound for the cluster-size, cluster-head independence requirement and load balancing.

Theorem 4.1: Given an undirected graph $G(V, E)$, the maximum clique $G(V', E')$ in $G(V, E)$ imposes a minimum value given by the cardinality of V' on the upper bound of the cluster-size that can be chosen without violating the cluster-head independence requirement at all times.

Proof: We assume a one-hop clustering scheme is used and that the graph $G(V, E)$ is connected. Other assumptions are that $G(V, E)$ can be decomposed into maximal cliques only such that the smallest possible maximal clique can have only two elements (or nodes), and that each one of such maximal cliques must have at least one node which does not belong to any other maximal clique. So we begin by identifying the largest maximal clique V' , and then the second largest maximal clique V'' and continue until all the nodes are placed in maximal cliques and the smallest maximal clique V^k is obtained. The cliques are now arranged from the largest to the smallest and come up with (V', V'', \dots, V^k) as a clique-decomposition of $G(V, E)$. Note that since $G(V, E)$ is connected some nodes which may be regarded as gateways belong to more than one maximal clique.

We now apply a clustering algorithm to the clique-decomposition of $G(V, E)$ by choosing the cardinality of V' as the upper bound on the cluster-size. It is not hard to see that such an algorithm will eventually terminate with $\{V', V'', \dots, V^k\}$ as the last possible set of clusters in which the cluster-head independence requirement is not violated. Since each maximal clique has at least one node which does not belong to any other maximal clique, that node becomes the cluster-head if there is no other possible candidate node to take up the role of the cluster-head. It follows that using any value greater than the cardinality of V' for the maximum cluster-size will yield a converging clustering algorithm without violating the cluster-head independence requirement.

Next, we apply the same clustering algorithm to a special case of $V' = V$ (i.e. $G(V, E)$ is made up of only one clique, V') using the cardinality of V' minus one (i.e. $|V'| - 1$) as the upper bound on the cluster-size. The algorithm will create a cluster of $(|V'| - 1)$ nodes and remain with one node to be placed in another cluster. This left out node must create a cluster of its

own and declare itself as the cluster-head, which contradicts the cluster-head independence requirement since it is a neighbour of the cluster-head of the first cluster. This one instance of violation of the cluster-head independence property confirms that it is possible to violate the property if $(|V'| - 1)$ is chosen as the upper bound for the cluster-size. It follows that choosing any value less than $|V'|$ as the maximum size of a cluster can also have an instance of violation of the cluster-head independence property. ■

Theorem 4.2: Load balancing cannot be achieved in a completely connected undirected graph $G(V, E)$, in which any two nodes are neighbours, while maintaining the cluster-head independence requirement.

Proof: We prove by contradiction. First, we assume it is possible. Given that $G(V, E)$ is a completely connected graph, it means that any two nodes in V are neighbours. Hence, regardless of the size of $G(V, E)$ we can only have one cluster-head without violating the cluster-head independence requirement. To achieve load balancing, we must have at least one more cluster-head which must come from V . Having an additional cluster-head from V contradicts the cluster-head independence property. ■

Theorem 4.3: Given an undirected graph $G(V, E)$, the probability of achieving load balancing while maintaining the cluster-head independence requirement decreases to zero as the cardinality of the maximum clique in $G(V, E)$ approaches that of V .

Proof: We assume a one-hop clustering scheme is used. Since the subgraph $G(V', E')$ of $G(V, E)$ induced by the maximum clique V' is completely connected, only one cluster-head can come from V' , and therefore additional cluster-heads must come from the set $V - V'$. So as the cardinality of V' increases, the size of $V - V'$ becomes smaller resulting in fewer candidates for additional cluster-heads, which in turn leads to decreasing probability of achieving load balancing. Eventually, when $V' = V$ we have $V - V' = 0$ resulting in failure to achieve load balancing (*Theorem 4.2*). ■

Based on *theorem 4.1* through *theorem 4.3*, it is clear that achieving load balancing by setting arbitrarily the limit on the maximum size of a cluster leads to the violation of the cluster-head independence property. It is tempting to use the cardinality of the maximum clique (*theorem 4.1*) in a network for the upper bound of the cluster-size, but the size of the maximum clique

may be too small or too big for practical use. Moreover, determining the maximum clique of an arbitrary undirected graph is a well known NP-hard problem [155, 156], and it is likely that its approximation methods may be too inefficient (i.e. using too much energy for computations) for the energy-constrained devices of MANETs.

To solve the load balancing problem in question, it is proposed in this research that secondary cluster-heads be introduced. Introducing secondary cluster-heads in a maximal clique which already has a cluster-head (a primary cluster-head) appears to adhere to the cluster-head independence property in principle, but the truth is that it violates the cluster-head independence property in as far as actual network operations are concerned. The argument in this thesis is that the likely violation of the cluster-head independence requirement in the actual operations of a network is significantly outweighed by the gains obtained in load balancing. Nevertheless, it is a fact that violating the cluster-head independence property leads to redundancy in network operations, and therefore defeats the purpose for which clustering is adopted in this research.

The proposed maximum number of nodes per cluster is 15. The number of 15 for the upper bound of the cluster size has been arrived at almost arbitrarily, and therefore it may be too small or too big to achieve acceptable network performance in some ELC-MANET environments. However, the belief in this thesis is that the chosen upper limit is realistic since it is close to 18 which produced satisfactory results in certain scenarios in the research conducted by T. Ohta et al. [157]. Choosing a value for the upper bound of the cluster size which is much lower than 15 is likely to result in more control overhead, while a value much higher than 15 is likely to result in overloading of the few resulting cluster-heads.

For every primary cluster-head, the number of secondary cluster-heads affiliated to it is not limited to one but can be more than one depending on the availability of neighbour nodes of the primary cluster-head. However, the proposal is to limit the number of secondary cluster-heads affiliated to a single primary cluster-head to 2. The secondary cluster-heads are recognised as first secondary cluster-head and second secondary cluster-head. The second secondary cluster-head is required to have all the remaining nodes not yet assigned to a cluster-head. The maximum number of 2 secondary cluster-heads has been arrived at by considering that in the targeted ELC-MANET environments the maximum class being considered is 50 and therefore assuming that we have a maximum clique of size 50 we need about 3 cluster-heads according to the proposal of having 15 nodes per cluster. Thus, the

primary cluster-head and the first secondary cluster-head can take 30 nodes, while the second secondary cluster-head can take the remaining 20 nodes.

Secondary cluster-heads are created only where setting a maximum number of 15 nodes per cluster leads to the violation of the cluster-head independence property. Where violation of the cluster-head independence property cannot take place, only primary cluster-heads are put in place. Note that all primary cluster-heads in a network adhere to the cluster-head independence requirement.

Only primary cluster-heads are applicable in the implementation of Basagni's distributed clustering algorithm [106]. The introduction of secondary cluster-heads necessitates that Basagni's algorithm be extended. The extension is that when a primary cluster-head announces its role to its neighbours it includes in the message the nodes that have been selected as secondary cluster-heads and the member nodes belonging to each selected secondary cluster-head. Thus, it is the primary cluster-head that selects secondary cluster-heads. The primary cluster-head ensures that only nodes that are one-hop neighbours of both the primary cluster-head and a selected secondary cluster-head are assigned as cluster member nodes of the selected secondary cluster-head. This ensures that all the nodes assigned to a secondary cluster-head are reached by the secondary cluster-head through one-hop communication only.

Load balancing through the node redistribution parameter

The idea of setting the upper bound for the cluster-size is used in this research to address load balancing in a network. In addition to setting a maximum size for clusters, a novel technique that uses the node redistribution parameter is introduced in this research to achieve load balancing. The reasoning behind the introduction of the node redistribution parameter is that it is possible to have uneven distribution of nodes among clusters in spite of setting the maximum size for clusters. The observation is that the ELC-cluster-based network for which CTRDM is developed already has physical clusters. Assuming that the physical clusters have almost the same number of nodes and that such a number of nodes is less than or equal to the maximum size set for the logical clusters (i.e. 15) produced by a clustering algorithm, the node redistribution parameter tries to reduce any imbalance in the number of nodes per logical cluster which may be introduced by a clustering algorithm. In other words, the node

redistribution parameter attempts to create logical clusters that have the same number of nodes as the physical clusters.

Considering that the existence of the ELC-cluster-based topology is a result of students' activities, it is likely that under normal circumstances the students are placed in almost equal-sized groups resulting in physical clusters of almost the same size. Hence, the assumption that the ELC-cluster-based topology has almost equal-sized physical clusters is realistic.

The derivation of the node redistribution parameter and its application are explained with the aid of Figure 4.3. Figure 4.3 illustrates possible connectivity links in a cluster-based network. Considering that the transmission range of current MANET devices is well above the diameter of a medium-sized discussion group of students (based on the default transmission range of 377m for the settings of GloMoSim-2.03 shown in Table 4.1 [146]), it is likely that physical clusters formed by students' discussion groups may have very high wireless connectivity such as the one shown in Figure 4.3. Therefore any two nodes belonging to a particular cluster, such as cluster-1 or cluster-2 in Figure 4.3, are likely to be neighbours. The problem that may prevent any two nodes in the same cluster from communicating directly are likely due to factors other than the transmission range such as the hidden terminal problem [126] and the use of different channels. In this research, we assume that such factors do not exist. In fact, we assume that the ELC-MANET uses one channel as its transmission medium.

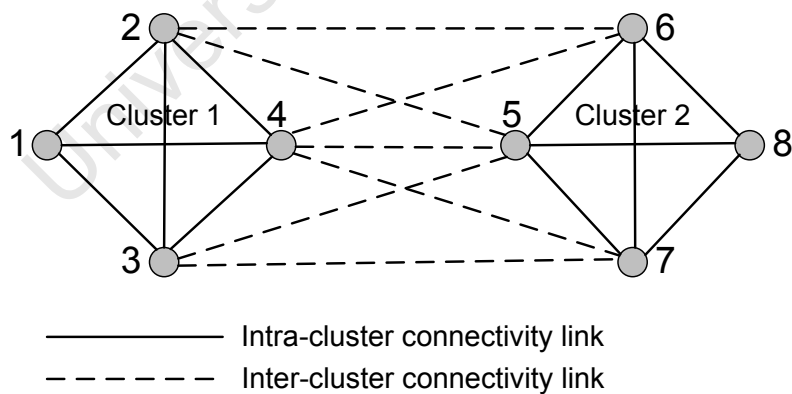


Figure 4.3 ELC-cluster-based network with 8 nodes

Based on the wireless connectivity within a physical cluster formed by students, we observe that the ELC-cluster-based network consists of cliques which represent physical clusters. Hence, cluster-1 and cluster-2 in Figure 4.3 are cliques. The cluster cliques of the ELC-cluster-based topology may not be maximal cliques. The connectivity patterns between any

two adjacent clusters (or cliques) can easily be inferred from Figure 4.3. Assuming that the nodes in Figure 4.3 have the same transmission range, it is likely that inter-cluster connectivity links are few when adjacent clusters are far apart. For example, when cluster 1 and 2 in Figure 4.3 are far apart, only nodes at the edges of the clusters such as 4 and 5 may be directly connected. As the distance between the clusters becomes smaller, even nodes situated in the middle of the clusters may have direct inter-cluster links. Eventually, as the clusters become very close to each other they form one clique. Note that this reasoning of having two separate cliques eventually merging as a resulting of clusters moving towards each other may also be observed by allowing the transmission ranges of the nodes to increase while the clusters remain stationary. This possible connectivity pattern between clusters is used as a basis for the derivation of the node redistribution parameter (covered in the subsequent paragraphs).

We now present the effect of applying certain types of clustering algorithms to the ELC-cluster-based topology such as the one shown in Figure 4.3. Applying a clustering algorithm that uses the degree of connectivity (i.e. the number of neighbours) as the only criterion or as the dominant scoring factor for selecting a node for the role of the cluster-head may have a possibility of selecting nodes 4 and 8 as cluster-heads. It is possible to have the following as the resulting clusters: cluster-1(1, 2, 3, 4, 5, 6, 7) and cluster-2(8). Nodes 4 and 5 have the highest number of neighbours which is 6 and therefore would declare themselves, independent of each other, as cluster-heads. Since nodes 4 and 5 are neighbours, one of them gives up the cluster-head role. Assuming that a tie on the degree of node connectivity is broken by a lower ID, node 5 withdraws its cluster-head declaration and may get affiliated to node 4 as its cluster-head. Upon hearing that node 5 is no longer a cluster-head, nodes 6 and 7 may also join node 4 as their cluster-head. Node 4 welcomes the nodes 5, 6 and 7 since the addition of the three neighbours does not exceed the upper limit of 15 for the cluster-size. Meanwhile, node 8 which has been waiting for nodes 5, 6 and 7 to announce that one of them is a cluster-head declares itself as a cluster-head upon hearing that all the three neighbours have joined cluster-1 as ordinary nodes.

Clearly, the number of nodes in the two clusters, namely cluster-1 and cluster-2, is not balanced. Such an imbalance in the cluster sizes may lead to less performance in the network compared to a situation of having balanced cluster sizes. Considering that the ELC-cluster-based network is composed of clusters, which form cliques, and that any node belonging to a

cluster that acts as a gateway (i.e. has an inter-cluster link) to another cluster has more neighbours than any other node in its cluster or clique, the possibility of creating clusters with imbalanced sizes is likely to happen if the degree of connectivity is used as a selection criterion for cluster-heads. Hence, the proposal in this research is to replace the degree of node connectivity with the node redistribution parameter as one of the factors for selecting a node to become a cluster-head.

The reason why the degree of connectivity is leading to imbalances in the cluster-sizes is because gateways are selected for the role of the cluster-head. Obviously, rejecting gateways from becoming cluster-heads may not be a straightforward solution considering that the introduction of the upper limit on the cluster-size may yield a situation of having only gateways wanting to form a cluster. Another problem with a criterion based on whether a node is a gateway or not is in its implementation in a system where a node score is computed based on several factors such as mobility and remaining battery energy. Since gateway nodes have some degree of “external connectivity”, the proposal in this research is to use the degree of “local connectivity” as one of the weighting factors when computing a score for selecting nodes for cluster-head role. The term “local connectivity” now needs to be defined.

Definition 4.1: Assuming that node v has M^0 as its set of one-hop neighbours that include node v itself, $k+1$ as the size of M^0 and that M^1, M^2, \dots, M^k are the neighbour sets of node v 's first neighbour, second neighbour and k^{th} neighbour respectively, the term “local connectivity” for node v is defined as the intersection of the sets M^0, M^1, M^2 up to M^k (i.e. node v 's local connectivity = $M^0 \cap M^1 \cap M^2, \dots, M^{k-1} \cap M^k$).

A node v has a large intersection of the neighbour sets when node v , neighbours of node v and neighbours of node v 's neighbours are mostly within the same clique. A gateway node is likely to have a small intersection of the neighbour sets since some of its neighbours and neighbours of its neighbours are in other cliques. The degree of local connectivity, L_v , for node v is given by the following formula:

$$L_v = \frac{|(M^0 \cap M^1 \cap M^2, \dots, M^{k-1} \cap M^k)|}{k+1} \quad (4.1)$$

such that $L_v \in [0, 1]$.

We now apply Equation (4.1) to the ELC-cluster-based network shown in Figure 4.3. Before applying a clustering algorithm that uses the degree of local connectivity as the only criterion

for selecting nodes to be cluster-heads, we need to have the values of L_v for each of the nodes in the network. Table 4.2 presents the L_v values of the nodes in cluster 1 of the network in Figure 4.3.

Table 4.2 A sample of the degree of local connectivity values

Node	Neighbour Lists (M^0, \dots, M^k)	L_v
Node 1	$M^0 = \{1, 2, 3, 4\}$, $M^1 = \{1, 2, 3, 4, 5, 6\}$, $M^2 = \{1, 2, 3, 4, 5, 7\}$, $M^3 = \{1, 2, 3, 4, 5, 6, 7\}$	$L_1 = 1.0$
Node 2	$M^0 = \{1, 2, 3, 4, 5, 6\}$, $M^1 = \{1, 2, 3, 4\}$, $M^2 = \{1, 2, 3, 4, 5, 7\}$, $M^3 = \{1, 2, 3, 4, 5, 6, 7\}$, $M^4 = \{2, 3, 4, 5, 6, 7, 8\}$, $M^5 = \{2, 4,$ $5, 6, 7, 8\}$	$L_2 = 0.33$
Node 3	$M^0 = \{1, 2, 3, 4, 5, 7\}$, $M^1 = \{1, 2, 3, 4\}$, $M^2 = \{1, 2, 3, 4, 5, 6\}$, $M^3 = \{1, 2, 3, 4, 5, 6, 7\}$, $M^4 = \{2, 3, 4, 5, 6, 7, 8\}$, $M^5 = \{3, 4,$ $5, 6, 7, 8\}$	$L_3 = 0.33$
Node 4	$M^0 = \{1, 2, 3, 4, 5, 6, 7\}$, $M^1 = \{1, 2, 3, 4\}$, $M^2 = \{1, 2, 3, 4, 5,$ $6\}$, $M^3 = \{1, 2, 3, 4, 5, 7\}$, $M^4 = \{2, 3, 4, 5, 6, 7, 8\}$, $M^5 = \{2, 4,$ $5, 6, 7, 8\}$, $M^6 = \{3, 4, 5, 6, 7, 8\}$	$L_4 = 0.14$

In cluster 2, an analysis similar to the one that produces the results shown in Table 4.2 yields $L_5 = 0.14$, $L_6 = 0.33$, $L_7 = 0.33$ and $L_8 = 1.0$. When the clustering algorithm is now applied, nodes 1 and 8 declare themselves as cluster-heads and the resulting clusters are cluster-1(1, 2, 3, 4) and cluster-2(5, 6, 7, 8).

When node 1 announces that it is a cluster-head, only nodes 2, 3, and 4 hear it and become affiliated to it, resulting in the creation of cluster-1. Likewise, when node 8 declares that is a cluster-head, only nodes 5, 6 and 7 receive the announcement and become affiliated to it, resulting in the creation of cluster-2. Clearly, the created logical clusters by the clustering algorithm are the same as the physical clusters such that the nodes in the logical clusters are as evenly distributed as they are in the physical clusters. While in this example the logical clusters are exact replicas of the physical clusters, application of a clustering algorithm that solely uses the degree of local connectivity as the criterion for cluster-head selection may not yield exact replicas of physical clusters. Nevertheless, such an algorithm has a high

probability of producing almost equally sized logical clusters, provided that the physical clusters in the underlying network are also almost equally sized.

The effectiveness of the degree of local connectivity depends on the level of connectivity between adjacent clusters. In Figure 4.3, if we let the two clusters move towards each other, the level of connectivity between the clusters becomes so high that the two clusters begin to merge into a single clique. At that point, the degree of local connectivity of each one of the 8 nodes is equally high, and therefore each node becomes equally likely to be selected as a cluster-head. Note that at the point of having a merged clique, a clustering algorithm solely based on the degree of node connectivity (i.e. number of neighbours) would produce almost the same results as the clustering algorithm based on the degree of local connectivity. Also note that the use of the degree of local connectivity as expressed by Equation (4.1) applies to MANET cluster-based topologies only. Applying the formula to a MANET topology which has no physical clusters may not yield the desired results.

The degree of local connectivity is renamed “the node redistribution parameter” based on the fact that its effect in a network is to remove imbalances in node distribution among logical clusters. Hence, Equation (4.1) now becomes:

$$ND_v = \frac{|(M^0 \cap M^1 \cap M^2, \dots, M^{k-1} \cap M^k)|}{k+1} \quad (4.2)$$

where ND_v is the node redistribution parameter for node v . It now implies that a score for a node comes from three factors, namely the node redistribution parameter, the residual battery energy and the degree of mobility. The node redistribution parameter is considered to be the most important among the three factors and therefore it is a dominating factor. The proposed weightings for the node redistribution parameter, the remaining battery energy and the degree of mobility are 0.7, 0.2 and 0.1 respectively. The degree of mobility has the least weighting because it is assumed that once students are in discussion groups there is minimal mobility among them.

The method that is proposed in this thesis for determining the degree of local connectivity by performing set operations on lists of neighbour nodes is obviously not the only one, and perhaps not the most accurate one. An ideal technique would use the geographic distances between the nodes to determine how close they are to one another. Such a technique would need the use of a positioning system such as GPS or an approach based on signal strengths.

The method proposed in this thesis is considered to be suitable for the ELC-cluster-based network based on the following reasons:

- It is likely to be computationally more efficient than techniques based on signal strengths, and therefore it can use more efficiently the limited battery energy of MANET-devices.
- Considering that GPS performs poorly in indoor environments and that the ELC-MANET is proposed for both indoor and outdoor environments, the proposed method is likely to perform better than a technique using GPS.

Degree of node mobility

The degree of mobility of a node with respect to its neighbours is used by many researchers to determine whether such a node can take up the responsibility of the cluster-head or not [35, 107, 111, 121]. Nodes that are less mobile are preferred for the role of the cluster-head since they are likely to maintain stable clusters. Researchers have differed significantly in their way of incorporating the factor of mobility in selecting cluster-heads. Y. Zhang and J. M. Ng [121] have proposed a criterion for selecting nodes for the role of the cluster-head based mainly on the level of mobility of a node relative to its neighbours in a group. The criterion uses the direction, speed and linear displacement of a node relative to its neighbours to compute a parameter called total spatial dependency. Nodes with higher values of total spatial dependency have a large number of neighbours with which they have minimal relative motion, and therefore such nodes have higher chances of being selected to become cluster-heads. The cluster-head selection method by Y. Zhang and J. M. Ng [121] requires location information of each node to be available through a positioning system such as the Global Positioning System.

M. Chatterjee et al. [35] have proposed a method for selecting nodes for the cluster-head role based on computing a score for each node, and have considered mobility as one of the factors in the computation. The mobility component of a score is computed as the average speed of a node till current time t . The coordinates of a node at any time t are needed in the computation of the average speed and this requires the geographic location information of each node should be made available. The authors have proposed the use of a positioning system such as the Global Positioning System for determining location information. They have also

suggested the use of signal strengths to determine relative speed in the absence of a positioning system.

An indirect way of measuring mobility of a node with respect to others has been proposed by S. Adabi et al. [107]. The method uses a parameter called stability which is computed as the total time in which the neighbours of a specific node have spent their time beside the node. A higher value of stability implies that the neighbours of the node under consideration have spent a longer time in its transmission range, and therefore the node has a more stable situation which is desirable for the cluster-head role.

Another indirect way of measuring node mobility, proposed by A. R. Hussein et al. [158], determines the average mobility of a node by examining the number of the neighbours of a node at successive intervals of time. The value obtained is then used to predict the future mobility trends of the node. Assuming $X_v^t, X_v^{t+1}, X_v^{t+2}, \dots, X_v^{t+k}$, are sets of neighbours of node v at times $t, t+1, t+2, \dots, t+k$ respectively, the method by A. R. Hussein et al. [158] finds the set differences $|X_v^{t+1} - X_v^t|, |X_v^{t+2} - X_v^{t+1}|, \dots, |X_v^{t+k} - X_v^{t+(k-1)}|$ and then finds a weighted average value of the set differences. The weighting of the terms (i.e. the set differences) in the computation of the average gives more weighting towards set differences of the later intervals of time such that the set difference of the last two intervals of time (i.e. $|X_v^{t+k} - X_v^{t+(k-1)}|$) has the largest weighting.

A method similar to the one developed by A. R. Hussein et al. [158] is proposed in this thesis for predicting the degree of mobility of a node which is one of the factors for selecting nodes to become cluster-heads. The main similarity between the proposed method and that of A. R. Hussein et al. [158] is that both use the differences in the number of neighbours of node obtained at regular intervals of time to predict the future mobility behaviour of the node.

The two methods differ in two ways. First, while the method by A. R. Hussein et al. [158] focuses on the average mobility determined by computing neighbour set differences obtained from a number of successive intervals of time, the proposed method only considers the neighbour set difference of the interval just before time t under consideration. The latter method is based on the philosophical thinking that the mobility behaviour of a node just before time t is the best predictor of its behaviour just after the time t such that considering events in the interval just before t alone is more likely to give a better prediction than considering events in the interval just before t plus some events from the earlier time

intervals. The second difference is that the method by A. R. Hussein et al. [158] has only considered the set difference $A - B$, assuming A and B are neighbour sets of two adjacent intervals of time, while the proposed method considers both $A - B$ and $B - A$ since the two set differences are not commutative and therefore have different elements which may have different interpretations as the case is in the proposed method.

The proposed method is suitable for the ELC-MANET environments mainly for two reasons:

- It does not require the use of the Global Positioning System which performs poorly in indoor environments.
- It uses lists of neighbour nodes that are manipulated by set operations, and therefore it is likely to be computationally more efficient and accurate than methods based on signal strengths.

We begin the description of the proposed method by defining two sets N_v^1 and N_v^2 , where N_v^1 is the set of neighbours of node v at time t_1 and N_v^2 is the set of neighbours of the same node v at a later time t_2 . The understanding is that during the interval (t_1, t_2) , one or two or all of the following three situations regarding neighbours of node v may have taken place:

- Retaining some or all of the neighbours acquired at time t_1 .
- Losing some or all of the neighbours acquired at time t_1 .
- Adding new neighbours.

Note that a situation in which N_v^1 and N_v^2 are both empty implying that node v is empty at time t_1 and empty again at time t_2 is treated as a special case. Another thing to note is that the mobility of the neighbours of node v within node v 's transmission is not considered since it is assumed that such mobility can not affect connectivity between node v and such moving neighbours.

Assuming that during the interval (t_1, t_2) no new neighbour node is acquired and lost such that its information is not recorded in either set N_v^1 or N_v^2 , and that no neighbour node in N_v^1 leaves the vicinity of node v and comes back before t_2 , the following postulates are considered to be true:

- $|N_v^1 \cap N_v^2|$ gives the number of neighbours retained by node v .
- $|N_v^1 - (N_v^1 \cap N_v^2)|$ gives the number of lost neighbours.

- $|N_v^2 - (N_v^1 \cap N_v^2)|$ represents the number of added neighbours.

The argument here is that the lost neighbours plus the added neighbours (i.e. $|(N_v^1 - (N_v^1 \cap N_v^2)) \cup (N_v^2 - (N_v^1 \cap N_v^2))|$) constitute a set of nodes that have been in motion affecting connectivity of node v . In other words, the lost and added neighbour nodes have been in motion relative to node v , or node v has been in motion relative to them (from the viewpoint of physics). The latter possibility can be interpreted as a situation where node v moves among stationary nodes.

The degree of relative mobility of node v , denoted by M_v , is now obtained as a fraction of the sum of the lost and added neighbour nodes over all the neighbour nodes that interacted with node v during the interval (t_1, t_2) , and is expressed as Equation (4.3):

$$M_v = \frac{|(N_v^1 - (N_v^1 \cap N_v^2)) \cup (N_v^2 - (N_v^1 \cap N_v^2))|}{|(N_v^1 - (N_v^1 \cap N_v^2)) \cup (N_v^2 - (N_v^1 \cap N_v^2)) \cup (N_v^1 \cap N_v^2)|} \quad (4.3)$$

Simplification of Equation (4.3) yields the following formula:

$$M_v = \frac{|(N_v^1 \cup N_v^2) - (N_v^1 \cap N_v^2)|}{|N_v^1 \cup N_v^2|} \quad (4.4)$$

Equation (4.4) is undefined for a situation where both N_v^1 and N_v^2 are empty (i.e. $N_v^1 = N_v^2 = \emptyset$). Such a situation can be interpreted as an isolated node case which implies that node v has been in isolation during the interval (t_1, t_2) . Since such a situation can happen in practice, the proposal is to assign $M_v = 0$ % to an isolated node. By assigning the lowest value of M_v to an isolated node gives the node the chance to elect itself as a cluster-head.

The application of Equation (4.4) to some node mobility scenarios of the ELC-MANET is expressed in the form of propositions which are as follows:

Proposition 4.1: A node v , which is joining a MANET (or the ELC-MANET) such that at time t_1 it has no neighbours (i.e. $N_v^1 = \emptyset$) and at a later time t_2 it has acquired some neighbours (i.e. $N_v^2 \neq \emptyset$), has the highest level of mobility (i.e. $M_v = 1$) during the interval (t_1, t_2) .

Proof: Since $N_v^1 = \emptyset$ and $N_v^2 \neq \emptyset$, $|(N_v^1 \cup N_v^2) - (N_v^1 \cap N_v^2)| = |N_v^2|$ and $|N_v^1 \cup N_v^2| = |N_v^2|$. Hence, Equation (4.4) becomes $M_v = |N_v^2| / |N_v^2| = 1$. ■

Proposition 4.2: A node v , which is leaving a MANET such that at time t_1 it has some neighbours (i.e. $N_v^1 \neq \emptyset$) and at a later time t_2 it has no neighbours (i.e. $N_v^2 = \emptyset$), has the highest level of mobility (i.e. $M_v = 1$) in the interval (t_1, t_2) .

Proof: Since $N_v^1 \neq \emptyset$ and $N_v^2 = \emptyset$, $|(N_v^1 \cup N_v^2) - (N_v^1 \cap N_v^2)| = |N_v^1|$ and $|N_v^1 \cup N_v^2| = |N_v^1|$. Hence, Equation (4.4) becomes $M_v = |N_v^1| / |N_v^1| = 1$. ■

Proposition 4.3: When a node v is moving relative to other nodes (or other nodes are moving relative to a node v) in a MANET such that at time t_1 it has some neighbours (i.e. $N_v^1 \neq \emptyset$) and at a later time t_2 it has only new neighbours that are acquired after time t_1 , the mobility of the node v is at its highest level (i.e. $M_v = 1$) in the interval (t_1, t_2) . (Note that in the ELC-MANET, this scenario is likely to occur frequently in the ELC-random topology environment.)

Proof: In this mobility scenario, $N_v^1 \cap N_v^2 = \emptyset$ and this implies that $|(N_v^1 \cup N_v^2) - (N_v^1 \cap N_v^2)| = |N_v^1 \cup N_v^2|$. Hence Equation (4.4) becomes $M_v = |N_v^1 \cup N_v^2| / |N_v^1 \cup N_v^2| = 1$. ■

Proposition 4.4: If the neighbours of a node v in a MANET at time t_1 are exactly the same as those at a later time t_2 , the mobility of the node v is at its lowest level (i.e. $M_v = 0$) during the interval (t_1, t_2) . (Note that this scenario usually happens to nodes that are all stationary in a neighbourhood or that are all moving at the same speed and in the same direction. In the ELC-MANET, this scenario is likely to happen most of the times when students are attending lessons in a classroom (i.e. ELC-matrix topology environment) and when they are in discussion groups (i.e. ELC-cluster-based topology environment).)

Proof: Since the neighbours at t_1 and t_2 are exactly the same, we have $N_v^1 = N_v^2$. We assume that $|(N_v^1 \cup N_v^2) - (N_v^1 \cap N_v^2)| = |N_v^1 \cup N_v^2| - |N_v^1 \cap N_v^2|$. Since $N_v^1 = N_v^2$, $|N_v^1 \cap N_v^2| = |N_v^1 \cup N_v^2|$. Hence Equation (4.4) yields $M_v = 0$. ■

The main argument behind M_v is that the interaction of the node v with its neighbours during the interval (t_1, t_2) is reliable information to use in predicting its interaction with its neighbours in the next interval. Hence, if node v is considered to be mobile in the interval (t_1, t_2) due to losing and adding a large number of neighbours, it is likely that it would continue to do so in the next interval. Of course, this is a matter of probability and therefore it is possible

that in the next interval its interactions may be in the opposite way. Nevertheless, if (t_1, t_2) is small enough, the probability is small that the interactions of node v with its neighbours may change significantly in the next interval.

Simplicity of Equation (4.4) leads to easy implementation. Node v needs only to know its neighbours at any time t , which is not a problem since such information is obtained from hello messages of CTRDM. The interval (t_1, t_2) is the same interval with which CTRDM uses to refresh the formation of its clusters. Obviously, has to be as small as possible in order to yield meaningful results from M_v .

To make sure that all the parameters contributing to a node score are assigned higher values for cluster-head suitability and lower values for cluster-head unsuitability, the degree of relative mobility is converted into a node stability factor S_v using the following expression:

$$S_v = 1 - M_v \quad (4.5)$$

Residual battery energy

CTRDM uses the remaining battery energy as one of its scoring factors upon which selection of a node for the role of the cluster-head is based. The assumption being made on the residual battery energy is that a procedure of CTRDM responsible for determining residual battery energy is able to obtain the information on the remaining battery energy from the operating system of a MANET-enabled device through the operating system's in-built function similar to a date function. Residual battery energy of a node v is herein denoted by E_v .

Normalisation of scoring factors

Normalisation of scoring factors is considered to be an important concept in this thesis because it brings the factors to a common unit which enables the magnitudes of the factors to be compared easily. Hence, the values of the node redistribution parameter (i.e. ND_v), the node residual battery energy (i.e. E_v), and the node stability (i.e. S_v) are all assumed to be bounded between 0 and 1.

Summary on the CTRDM's clustering method

The proposed CTRDM's clustering scheme is a weight-based method. Selection of nodes for the role of the cluster-head is based on their scores. Given two nodes with different scores,

the node with a higher score is more suitable for the cluster-head role than the node with a lower score. A node score is composed of three factors: namely the node redistribution parameter, ND_v ; the node stability factor, S_v ; and the residual battery energy factor, E_v . Hence, the node score is computed based on the following formula:

$$score = (w_1 \times ND_v) + (w_2 \times S_v) + (w_3 \times E_v) \quad (4.6)$$

where w_1 , w_2 and w_3 are weightings with values proposed as follows: $w_1 = 0.6$, $w_2 = 0.1$ and $w_3 = 0.3$. A possibility of having any two neighbouring nodes declaring themselves as cluster-heads due to having the same score is resolved by selecting the node with a lower ID as the cluster-head.

From the perspective of the research hypothesis and postulate (1) presented under the research hypothesis in chapter 1, the use of the proposed clustering algorithm for CTRDM has a high probability of yielding a CDS which is very close to the MCDS of the ELC-cluster-based topology. The creation of logical clusters by the proposed clustering algorithm is basically a heuristic approach that addresses the MCDS problem. Note that the novel contributions of this research in the proposed clustering algorithm are the introduction of secondary cluster-heads that provide load balancing, the node redistribution parameter (i.e. Equation (4.2)) which also provides load balancing, and the node mobility formula (i.e. Equation (4.4)).

4.4 ELC-Random Topology

4.4.1 Characteristics of the ELC-random topology

The ELC-random topology is the third distinct topology of the ELC-MANET. When students disperse to various locations but within their school campus, they form the ELC-random topology. Here, again, the assumption is that each student holds only one MANET-enabled device. The ELC-random topology is likely to cover a larger geographical area than the ELC-matrix topology which is formed by students only when they are in a classroom environment. A geographical area covered by the ELC-random topology may also be larger than that of the ELC-cluster-based topology since in the former topology students are more likely to move away from the vicinity of their classrooms than in the latter topology.

The ELC-random topology does not only differ from the other two ELC-MANET topologies in terms of the geographical area of coverage but also in terms of the level of mobility of nodes. Since when students disperse after a lesson or a group discussion they are free to move to anywhere they want, it is likely to have more students on the move in the ELC-random topology than in the other two ELC-MANET topologies. Hence, nodes in the ELC-random topology are expected to be more mobile than in the other two topologies.

While wireless connectivity in both the ELC-matrix topology and the ELC-cluster-based topology has some pattern, wireless connectivity in the ELC-random topology has no definite pattern. Nobody has control over the activities of students once they are out of a classroom or a group discussion, and hence the resulting topology formed by the students is highly unpredictable. This unique characteristic of the connectivity of the ELC-random topology with relative to the other two ELC-MANET topologies suggests that the ELC-random topology like each one of the other two topologies requires a route discovery method of its own.

4.4.2 Random Topology Route Discovery Method

The proposed route discovery method of TAPMRP intended for the ELC-random topology is called a Random Topology Route Discovery Method (RTRDM). A clustering technique called passive clustering developed by Y. Yi et al. [100] is recommended for use as the broadcasting method for RTRDM. Passive clustering creates clusters using on-going data traffic. Cluster related information is piggybacked in data packets. Neighbourhood information is made available to nodes through promiscuous packet receptions. Hence, passive clustering generates much less control overhead compared to active clustering. A major drawback of passive clustering is a noticeably long set up latency which may not be acceptable to some time-sensitive applications [95].

From the point of view of the research hypothesis and postulate (3) presented under the research hypothesis in chapter 1 (page 9), the adopted passive clustering technique is capable of producing a CDS which is close to the MCDS of the ELC-random topology at any instant. The use of passive clustering in [100] is fundamentally a heuristic approach that tackles the MCDS problem. The contribution of this research from this section is the application of the passive clustering technique to the ELC-random topology of the ECL-MANET. The

application is based on the reasoning that passive clustering is capable of creating a CDS which is adaptive to the mobility behaviour of nodes.

The reasons for recommending the use of passive clustering for broadcasting in RTRDM are as follows:

- Since it generates less control overhead than active clustering, it can use the limited battery energy and bandwidth of MANETs in a more efficient way.
- It is robust to mobility through creation of clusters which dynamically adapt to mobility behaviour of mobile nodes.
- It likely to perform better than a method based on GPS which performs poorly in indoor environments.

While MTRDM and CTRDM are improvements on the blind flooding method of ODMRP, RTRDM uses a broadcasting technique (i.e. passive clustering) which is already proposed in ODMRP [9]. What is new in this thesis related to RTRDM is that the passive clustering technique proposed in ODMRP is being applied to a special networking environment which is the ELC-random topology environment. RTRDM also consists of a route request phase and a route reply phase just like MTRDM and CTRDM. Only cluster-heads and bridge nodes are allowed to act as forward nodes of broadcast packets in RTRDM. A design specification of RTRDM is presented in chapter 5.

4.5 Summary

This chapter covers the characteristics of the three phases of the ELC-MANET identified in this thesis. The chapter also presents the proposed route discovery methods of the proposed multicast routing protocol (i.e. TAPMRP) for the ELC-MANET. The first phase of the ELC-MANET to be covered is the ELC-matrix topology. The observation in this research is that the ELC-matrix topology is characterised by high network connectivity and uniform distribution of nodes. The high network connectivity of the ELC-matrix topology is due to its confinement to a relatively small geographical area (i.e. the classroom environment). MTRDM is proposed as the route discovery method of TAPMRP suitable for the ELC-matrix topology. Selection of forward nodes of join request packets in MTRDM is based on the partial dominant pruning technique [39].

The next ELC-MANET phase to be considered is the ELC-cluster-based topology which is characterised by high connectivity within clusters and relatively low connectivity between clusters. CTRDM is the proposed route discovery method of TAPMRP for the ELC-cluster-based topology. Broadcasting of control packets (i.e. join request packets) in CTRDM is based on a cluster-based broadcasting method proposed by J. Wu and W. Lou [46], which selects a subset of gateway nodes as forward nodes at each cluster-head using the greedy set cover algorithm.

CTRDM has an underlying clustering platform which is designed based on distributed clustering algorithm proposed by S. Basagni [106]. The algorithm by S. Basagni is extended by incorporating load balancing, specification of node weights, and identification of bridge nodes using a technique called a marking process developed by J. Wu and H. Li [72]. Load balancing is provided using secondary cluster-heads and the node redistribution parameter. Computation of a node weight in the clustering platform of CTRDM is based on three factors, namely the node redistribution parameter, degree of node mobility, and the residual battery energy. The node weight is used as a criterion for selecting nodes for the role of cluster-head.

The ELC-random topology is the last ELC-MANET phase to be covered in this chapter. Lack of a definite pattern of network connectivity is one of the characteristics of the ELC-random topology. Another characteristic is the presence of node mobility which is likely to be higher than in the other two ELC-MANET phases. The ELC-random topology is more likely to cover a larger geographical area than the areas covered by the other ELC-MANET phases.

The route discovery method of TAPMRP proposed to serve the ELC-random topology is RTRDM. RTRDM adopts a passive clustering technique developed by Y. Yi et al. [100]. Passive clustering is capable of adapting dynamically to the mobility of nodes. Only cluster-heads and gateway nodes are allowed to forward join request packets in RTRDM.

CHAPTER 5

DESIGN SPECIFICATION OF TAPMRP

5.1 Introduction

A multicast routing protocol proposed to serve the ELC-MANET environment is the Topology-Aware Polymorphic Multicast Routing Protocol (TAPMRP). Major components of TAPMRP are a route-discovery-method switch and three route discovery methods, namely MTRDM, CTRDM and RTRDM which are partially covered in chapter 4. TAPMRP is an adaptive multicast routing protocol which switches between MTRDM, CTRDM and RTRDM depending on the type of the prevailing ELC-MANET topology. TAPMRP is also a hybrid multicast protocol which has two pure on-demand route discovery methods (i.e. MTRDM and RTRDM) and one route discovery method (i.e. CTRDM) which has a reactive component (i.e. the route request and route reply phases) and a proactive component (i.e. the active clustering platform).

TAPMRP is intended to provide a highly efficient multicast routing method capable of utilising the limited bandwidth of ELC-MANETs and limited battery energy of MANET devices in the most efficient way possible. Hence, TAPMRP is equipped with mechanisms of minimising control overhead and maximising transmission of data packets. The high performance of TAPMRP comes from its ability to respond dynamically to changes in the ELC-MANET topology by way of switching to a route discovery method which best suits the prevailing topology.

The design specification of TAPMRP assumes that a node running TAPMRP already has an IP multicast address. Another assumption is that all the TAPMRP-enabled nodes in a network use one channel for packet transmission. The specification further assumes that all the connectivity links are bi-directional, and that all the MANET devices use omnidirectional antennas.

Like ODMRP, TAPMRP has a unicast capability. This implies that a MANET device using TAPMRP is not required to run a separate unicast protocol for it to deliver services that need

unicast routing. Each of the three route discovery methods of TAPMRP constructs a multicast tree for delivery of multicast services. In a unicast mode of operation, only one path (from a source node to one destination node) of a multicast tree is created and used to deliver packets to only one receiver node. This implies that a unicast mode of operation does not require creation and maintenance of separate routing information but uses the same routing databases (i.e. routing tables and message caches) created and maintained by a multicast mode of operation.

TAPMRP uses a controller node to handle operations that affect all the nodes belonging to a particular multicast group. The controller node runs the route-discovery-method switch of TAPMRP which is responsible for switching of the route discovery methods of TAPMRP. To avoid instability of a multicast group when selecting a new controller node, the concept of “First Declare First Wins” rule is adopted in TAPMRP.

There are four types of nodes that take part in a multicast group formed by TAPMRP: a multicast controller, a source node, a forwarding group node, and a receiver node. A node can take more than one role with the exception that it cannot simultaneously play the role of both a source and a receiver of the same multicast group. For example, a node can simultaneously be a controller node, a forwarding group node, and a receiver node of the same multicast group. Note that in all the three route discovery methods of TAPMRP, a multicast tree used for delivery of packets to receivers is rooted at the source node and not at the multicast controller node. While each multicast group has only one controller node, a multicast group can have as many source nodes as possible.

Unlike in ODMRP where a forwarding group node (or a forward node) can belong to a multicast group, which is different from the one to which the packet being forwarded belongs; all forward nodes of a particular multicast packet in TAPMRP belong to the same multicast group. In other words, a multicast group in TAPMRP is required to provide all the intermediate nodes that are required for successful delivery of its multicast packets. The idea of having forward nodes that may belong to other multicast groups used in ODMRP is avoided in TAPMRP because in practice cooperation of forward nodes that belong to other multicast groups cannot be guaranteed. As a result, two nodes are neighbours in TAPMRP only if they belong to the same multicast group. Nevertheless, any node in TAPMRP including a forward node can belong to more than one multicast group.

The rest of this chapter is organised in the following way. Section 5.2 covers the operations of joining and leaving a multicast group, followed by maintenance of local connectivity in section 5.3. In section 5.4, the multicast controller node is covered in detail. The route-discovery-method switch of TAPMRP is presented in section 5.5. Design specifications of MTRDM, CTRDM and RTRDM are covered in sections 5.6, 5.7, and 5.8 respectively. Section 5.9 presents an explanation on how TAPMRP ensures a seamless switch from one route discovery method to another. The implementation of TAPMRP is presented in section 5.10. The chapter ends with a summary in section 5.11.

5.2 Joining and Leaving a Multicast Group

When a node wants to be a multicast group member, it starts with an attempt to join an existing multicast group by waiting to receive a controller hello message coming from a multicast controller to which the node wants to be affiliated. After a controller timeout period, the node declares itself as a multicast controller. In this case, the node first sends out a new controller announcement followed by periodic controller hello messages.

To a join a multicast group, a node simply responds to a controller hello message, which is also called a Data-Request message. A reply message called a Data-Reply message contains data related to the number of neighbours of the node which is sending the reply. Upon receiving the Data-Reply message, the multicast controller node recognises that it has added one more member to its group. Since a controller hello message is sent out periodically and that every node that belongs to a multicast group whose controller has sent the message is required to respond to the message with a Data-Reply message, there is no generation of an explicit message in TAPMRP for a node to join a multicast group.

Like joining, leaving a multicast group in TAPMRP does not involve an explicit message. When a node wants to leave a multicast group it simply stops responding to controller hello messages. There is no need even for the multicast controller to notify the group members when it wants to leave the group. The multicast controller is free to disappear any time. After detecting the absence of the controller, each of the remaining nodes in the affected multicast group independently invokes a procedure for electing a new controller which eventually yields a universally accepted multicast controller for the group.

5.3 Maintenance of Local Connectivity

Every node is required to periodically send out a hello message to its neighbours. Neighbourhood information is therefore maintained by every node that belongs to a multicast group. A hello message is intended for local connectivity and therefore should not be confused with a controller hello message which is broadcast to an entire network. A hello message has a time to live (TTL) value of 1, which implies that it cannot be propagated beyond a one-hop neighbour. Two sets of information are contained in a hello message: a node ID and a list of all one-hop neighbours. Each time a node receives a hello message from a neighbour node, it updates its list of neighbours. When a node detects absence of a hello message from a particular neighbour, it removes the neighbour from its list. Hence, a node always maintains an up-to-date list of neighbours.

The header of the hello packet has the following fields:

- *Packet_type*
- *Source_IP_address*
- *Sequence_number*
- *Time_to_live (TTL)*

The *packet_type* field contains information which is used to identify the hello packet. The IP address of the sender of the hello packet is contained in the *source_IP_address* field. The *sequence_number* is issued by the node originating the hello packet and is used to uniquely identify each of the hello packets from that node. The *time_to_live* field contains a value of 1 which restricts the packet from propagating beyond a one-hop neighbour node. Not shown on the list of the packet header fields is the information field which contains a list of one-hop neighbours of the node sending the hello packet.

With the exception of RTRDM, the operations of MTRDM and CTRDM depend on the availability of neighbour lists in every node. The exchange of hello messages enables a node u to maintain a list, $N(u)$, of one-hop neighbours. When another node v , receives the list $N(u)$ from u through a hello message, is able to maintain a list of the neighbours of its neighbours (i.e. two-hop neighbours), $N(N(v))$. Hence, every node has the list $N(v)$ and $N(N(v))$. Both MTRDM and CTRDM require the list $N(v)$ and $N(N(v))$ for their operations. RTRDM does not need neighbour information since the passive clustering technique which it uses as its broadcasting scheme does not need exchanging of neighbour lists.

The development of TAPMRP has seriously considered the selection of the interval for hello messages since it has a direct impact on the performance of TAPMRP. The hello message interval of TAPMRP has a fixed value (i.e. a non-adaptive value) regardless of which route discovery method is making use of it. Since the route discovery methods that make use of the interval, namely MTRDM and CTRDM, are intended for the ELC-MANET environments that have minimal mobility of nodes, the hello message interval can be longer than that of most of the existing multicast protocols. The development goal is to make sure that the control overhead generated by TAPMRP as a whole is much less than that generated by the existing multicast protocols including those which do not use hello messages such as ODMRP. The goal is also aware of a general trade off between control overhead and quality of service in which minimising control overhead compromises quality of service, and improving quality of service results in increasing control overhead.

5.4 Multicast Controller Node of TAPMRP

Each multicast group has only one multicast controller node. Selection of a node to become a controller is based on the “First Declare First Wins” rule. The rule implies that a node that first declares itself to be a controller node becomes a “legally” recognised leader of a multicast group such that no other node belonging to the same multicast group can challenge the leader. The controller must make its presence known to all the nodes belonging to its group through sending out of periodic hello messages. When a node detects absence of a controller hello message from the controller within a timeout period, it declares itself as a controller node and announces its leadership status to the rest of the nodes whose multicast controller is missing. Any node that receives the notification from the new leader checks if the former leader is indeed missing. If it is true, it becomes a member of the new multicast controller; otherwise, it destroys the announcement and takes no further action.

Since every node decides independently to be become a controller upon detecting the absence of the current controller node and that the new controller announcement from the first node to detect the absence of the current controller is likely to experience some delay before reaching every node in the affected multicast group, there are likely to be multiple new controller announcements following the disappearance of the current controller. When a node receives a different new controller announcement it compares its time stamp with the time stamp of the announcement it has kept in its database and takes action on the earlier announcement. Note

that the comparison of the announcements takes place only if the announcement kept in the database is still valid (i.e. within a timeout period); otherwise, the stored announcement is automatically replaced by the new announcement.

While the process of electing a multicast controller node can work perfectly with synchronisation of the clocks used by the nodes, it is not necessary to maintain clock synchronisation. Running TAPMRP in an environment where node clocks are not synchronised, as usually is the case in practice, gives more chances to a node with the slowest clock to become a controller than to a node that first makes the announcement. In this case, the “First Declare First Wins” rule is compromised but there is nothing to worry about since the protocol still achieves its goal of having one controller per multicast group at any instant.

Since the absence of a current multicast controller is likely to trigger multiple new controller announcements, it is important to ensure that the broadcasting method employed is as reliable as possible. Hence, the preferred method for broadcasting of the new controller announcements is the blind flooding method. The blind flooding method is not an efficient method of broadcasting but it is preferred due to its high reliability in the presence of node mobility and fast implementation which does not require setting up of clusters as the case is with CTRDM and RTRDM.

Stability of a multicast group from the time a current multicast controller disappears to the time when a new controller is recognised by all the nodes in the affected multicast group is an important issue worth serious consideration. The guarantee provided by TAPMRP for the stability of the affected multicast group during such a situation is conveniently expressed through the following theorem:

Theorem 5.1: The new controller announcements generated independently by nodes belonging to a particular multicast group in TAPMRP when a current multicast controller disappears eventually converge into only one controller announcement whose source is recognised as the new multicast controller for the affected multicast group.

Proof: The proof makes use of the following facts. First, each node decides to become a new controller node independent of the others. Second, execution of the procedure that declares a node as a new controller (i.e. the new-controller-announcement procedure) is triggered by a current-controller-timeout procedure (i.e. a timer procedure or function which runs

periodically). Soon after triggering execution of the new-controller-announcement procedure, the current-controller-timeout procedure terminates. Once the new-controller-announcement procedure has sent out the announcement it also terminates. The termination of the execution of the new-controller-announcement procedure triggers periodic execution of a procedure used for sending out controller hello messages.

Controller hello messages are sent out at an interval shorter than the controller time-out interval. Once the node receives a controller hello message from another controller, it compares the time stamps and if it finds out that the other controller became a multicast leader at an earlier time, this more recent controller immediately stops sending out controller hello messages and also immediately recognises the other controller as the legitimate controller. This whole process (covered in the first two paragraphs of this proof) is the longest possible process which can take place in a node when a current controller disappears. For those nodes whose current-controller-timeout procedure has not yet detected the absence of a current controller, they may simply replace the expired controller with the new one when they receive a new controller announcement.

Upon receiving a new controller announcement or a controller hello message, every node compares the time stamp of the multicast controller stored in its database with the time stamp of the newly received message and accepts the multicast controller with an earlier time stamp. In the case of a tie on the time stamps, the controller with a lower ID wins the competition. This process sieves competing multicast controllers at every node until the earliest declared controller is obtained and retained. Therefore, assuming the earliest declared controller does not move out of the multicast group from the time it released its announcement and that it keeps on sending out controller-hello messages, it will eventually be the only one retained in every node of the affected multicast group. Based on the process described in the first two paragraphs of this proof, the other competing multicast controllers will eventually stop sending out controller hello messages and accept the earliest declared controller as the universally recognised multicast controller. ■

The “First Declare First Wins” rule is obviously not the only way which can be used to select a multicast controller node in TAPMRP. Another possible way is that of using nodes’ identification information such as the lowest ID or the highest ID as a criterion for selecting nodes for the role of the controller. Since the node that first detects the absence of the current controller is likely not the one with the lowest ID (or highest ID), some time may elapse

before the node with the lowest ID (or highest ID) is made aware of the situation. Hence, such a method is likely to have a longer response time than the “First Declare First Wins” rule. The preference of the “First Declare First Wins” rule in TAPMRP is based on the understanding that it offers a short response time.

The multicast controller node makes all the network-wide decisions for the operation of TAPMRP. The route-discovery-method switch which runs in a controller node only is used to make such decisions. The data which the route-discovery-method switch uses for its decision making comes from every node in a multicast group. The controller uses controller hello messages as Data-Request messages such that when a node receives a controller hello message it responds by sending a Data-Reply message back to the controller.

The Data-Request message (or controller hello message) is broadcast to an entire network using the blind flooding method. Sending of the Data-Reply message to the controller uses routing information contained in the routing tables of nodes created during the time of propagating the Data-Request packet (or the route request phase). In other words, the process of transmitting the Data-Request packet and the Data-Reply packet is the same as a route discovery process. This implies that transmission of the Data-Request packet and the Data-Reply packet does not need maintenance of separate routing databases (i.e. routing tables and message caches) in TAPMRP.

Figure 5.1 illustrates the process of generation and transmission of the Data-Request packet and the Data-Reply packet in TAPMRP.

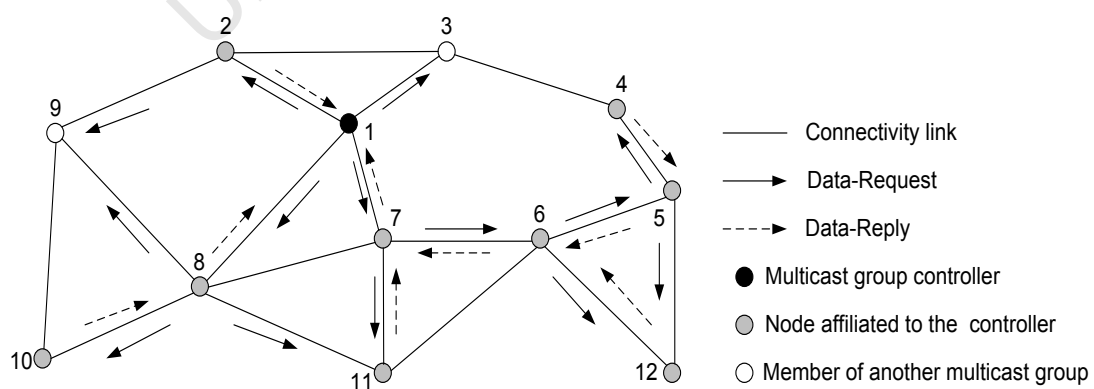


Figure 5.1 Propagation of the Data-Request and Data-Reply Packet in TAPMRP

As shown in Figure 5.1, the propagation of the Data-Request packet to every node in a multicast group is slightly different from the propagation of the Join-Query packet by the blind flooding method in ODMRP. The difference is that in TAPMRP members of other multicast groups are not involved in forwarding a broadcast packet (i.e. the Data-Request packet). For instance, node 3 in Figure 5.1 has received the Data-Request packet from node 1 but it cannot forward it to node 4. All multicast member nodes belonging to a particular controller node in TAPMRP are required to respond to the Data-Request packet with the Data-Reply packet as the case is with node 2, 4, 5, 6, 7, 8, 10, 11, and 12 in Figure 5.1. Node 3 and 9, in Figure 5.1, have not responded to the Data-Request packet with the Data-Reply message because they do not recognise node 1 as their multicast group controller.

While members of a multicast group have different roles to play such as the source node role, the role of forwarding group node, and the role of the multicast receiver node, the multicast controller node does not recognise those roles. What matters to the controller node is that all its members must respond to its periodic Data-Request message with the Data-Reply message, which should contain data indicating the number of one-hop neighbours of the node that has sent the Data-Reply message. Note that one-hop neighbours of a node v in TAPMRP are only those nodes that belong to a multicast group to which node v also belongs.

5.5 Route-discovery-method Switch of TAPMRP

The route-discovery-method switch is the “brain” of TAPMRP. As shown in Figure 5.2, all the three route discovery methods of TAPMRP are under the control of the route-discovery-method switch. Basically, the route-discovery-method switch is a procedure which consists of three sub-procedures. Each of the three sub-procedures has a specific task to perform. The tasks are as follows:

- (1) Computing the standard deviation of the number of neighbours of nodes belonging to a multicast group.
- (2) Using results of (1) to determine a switch from one route discovery method to another.
- (3) Sending out the results of (2) to members of a multicast group.

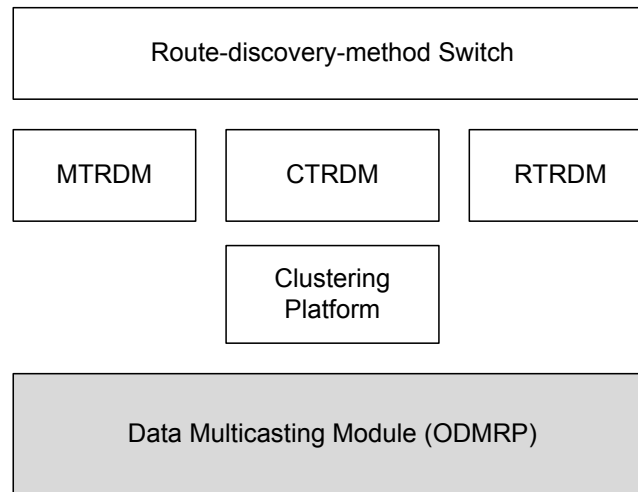


Figure 5.2 Main components of TAPMRP

5.5.1 Computation of the standard deviation of nodes' neighbours

As proposed in chapter 3, the standard deviation of the number of neighbours of nodes is used to detect the three distinct topologies of the ELC-MANET. The computation of the standard deviation is done by the route-discovery-method switch using a function which is hereinafter called a standard-deviation procedure. The standard-deviation procedure uses an online computation technique (refer to Figure 3.1 of chapter 3). The online algorithm is formulated in such a way that, each time data are received from any node, the standard deviation is computed for all the nodes whose data have been received so far. This implies that the route-discovery-method switch is able to make decisions on the standard deviation without waiting for all the nodes of a multicast group to submit data. Even though this is the case, results from the standard-deviation procedure are used by other procedures after a time interval which is long enough to ensure that data from all the nodes of a multicast group have been incorporated in the computation.

Multicast member nodes (i.e. source nodes, forward group nodes and receiver nodes) send their data on the number of their neighbours to the controller node through the Data-Reply packet. Each time the controller node receives the Data-Reply packet, it invokes the execution of the standard-deviation procedure, which extracts appropriate data from the received Data-Reply packet, and computes the standard deviation for the multicast nodes whose data have been received so far since the beginning of the current controller hello interval (covered in the second paragraph of subsection 5.5.2).

5.5.2 Determining a switch to another route discovery method

The route-discovery-method switch uses the results obtained from the standard-deviation procedure to determine a switch to another route discovery method. The procedure for deciding a switch to another route discovery method makes use of the time interval for sending out controller hello messages, which is hereinafter called a controller hello interval.

The route-discovery-method switch expects to receive Data-Reply messages from all the nodes in a particular multicast group within a controller hello interval. Hence, proper selection of the controller hello interval is required in TAPMRP. If the controller hello interval is too short such that it is shorter than the duration of a phase transition, it means a member node cannot record all the nodes it has interacted with during the phase transition. Hence, decisions based on such a situation are likely to be wrong decisions. Having a controller hello interval which is too long results in producing useless decisions since the route-discovery-method switch can miss some of the important events in network. For example, having a controller hello interval which accommodates two consecutive phase transitions of the ELC-MANET is not useful since one of the two phases cannot be served with the right route discovery method. The proposed controller hello interval is 120 seconds.

The route-discovery-method switch does not decide to switch to another route discovery method each time there is a timeout of the controller hello interval. The decision to switch to another route discovery method is made only after two consecutive controller hello timeouts (i.e. 240 seconds). This restriction on the time to make a switch is aimed at minimising instability in the operation of TAPMRP during a prolonged period of phase transition. During such a situation TAPMRP may just be switching from one route discovery method to another during each controller hello timeout. Selecting a switching time which is much longer than 240 seconds may not be all that useful in a situation where students change their activities frequently since the topologies resulting from some of the activities may not be served with the right route discovery method.

When a controller hello interval is over (i.e. at the time of detecting a timeout), the route-discovery-method switch finds out whether or not there was a switch from one route discovery method to another in the last controller hello interval. If there was no switch, the next controller hello message (or Data-Request message) is sent out without a switch decision announcement; if there was a switch, the execution of the procedure for deciding a switch to

another route discovery method is invoked. The procedure is hereinafter called a switching procedure.

The decisions made by the switching procedure are based on the conditions shown in a state transition diagram of Figure 5.3. In Figure 5.3, the operations of the route discovery methods of TAPMRP are illustrated as states of TAPMRP.

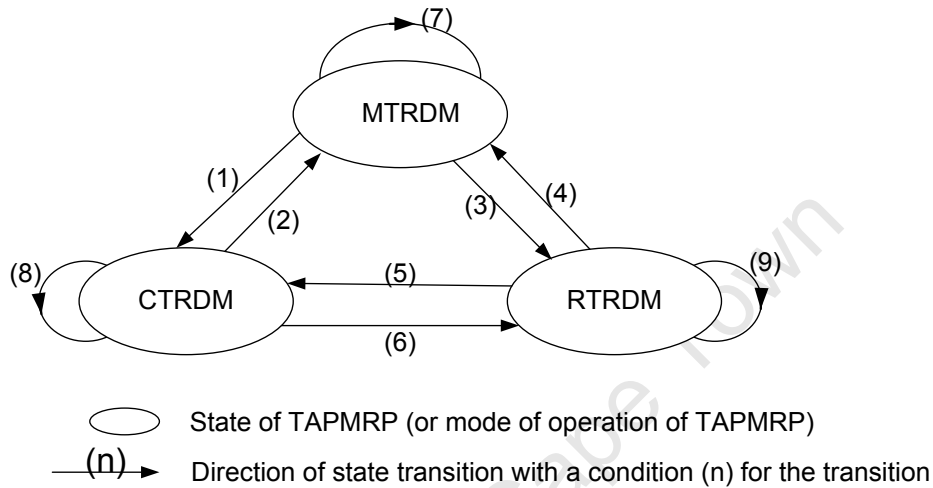


Figure 5.3 State transitions in TAPMRP

The conditions for the state transitions shown in Figure 5.3 are as follows:

- (1) A standard deviation value belonging to the ELC-cluster-based topology.
- (2) A standard deviation value belonging to the ELC-matrix topology.
- (3) A standard deviation value belonging to the ELC-random topology.
- (4) The same as condition (2).
- (5) The same as condition (1).
- (6) The same as condition (3).
- (7) Failure to meet conditions of (1) and (3).
- (8) Failure to meet conditions of (2) and (6).
- (9) Failure to meet conditions of (4) and (5).

Once the decision to switch to another route discovery method is made, the switching procedure then passes on its decision to the procedure which sends out the controller hello messages so that the decision can now be announced to the rest of the nodes in a multicast

group. Note that the decision to switch to another route discovery method includes condition (7), (8), and (9) of the state transitions in Figure 5.3.

Figure 5.4 gives a summary of the execution flow involving procedures of the route-discovery-method switch following the timeout of the controller hello interval. Note that the standard-deviation procedure does not appear in Figure 5.4 because it is executed each time a Data-Reply packet is received from a member node.

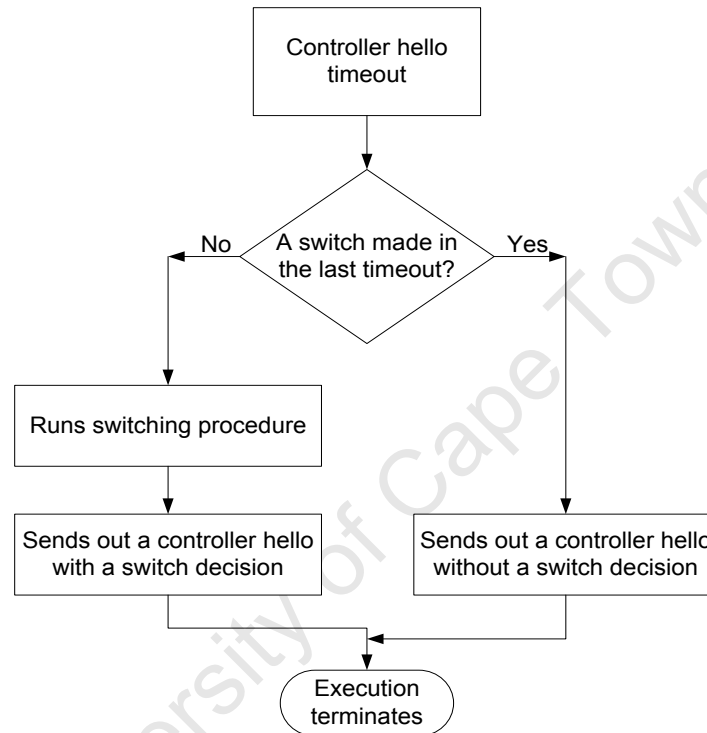


Figure 5.4 Execution flow of the procedures in the route-discovery-method switch

5.5.3 Announcing a switch to another route discovery method

The message that announces a switch to another route discovery method is carried in a controller hello packet. When a member node receives a controller hello message and detects the announcement of a switch to another route discovery method of TAPMRP, it makes the switch immediately independent of the other nodes in a network. Such a switch is mainly made by updating route discovery method information of a multicast group to which the switch announcement belongs. In the case of a switch to CTRDM, the node that has received the switch announcement initiates a clustering procedure even if it may not have data to transmit. Note that switch announcements corresponding to the condition (7) (8) and (9) of

the state transitions in Figure 5.3 do not result in the actual changing of route discovery information.

When a node has data to transmit, it uses the latest route discovery method announced by the controller node for a route discovery operation. Propagation of the resulting Join-Query by intermediate nodes does not solely rely on the latest route discovery method kept by the intermediate nodes. When the route discovery method carried in the Join-Query packet matches with that kept by an intermediate node, the intermediate node propagates the Join-Query using the route discovery method carried in the packet (or the one kept by the intermediate node); when the route discovery methods differ, the intermediate node propagates the Join-Query packet using MTRDM.

5.5.4 The controller hello packet and the data-reply packet

TAPMRP uses the controller hello message for three purposes: 1) to inform the member nodes about the presence of the controller node, 2) for requesting data for use in the computation of the standard deviation of the number of neighbours of nodes, and 3) for announcing a switch of a route discovery method of TAPMRP. The controller hello packet header uses the following fields:

- *Packet_type*
- *Source_IP_address*
- *Multicast_group_IP_address*
- *Sequence_number*
- *Last_hop_IP_address*
- *Hop_count_to_the_controller_node*
- *Time_to_live (TTL)*

The *packet_type* field is used to identify the controller hello packet. Once a member node detects (with the help of the *packet_type* field) that it has received a controller hello packet, it updates its records about the presence of the controller node and it immediately prepares a Data-Reply packet and sends it back to the controller. *Source_IP_address* field contains the IP address of the controller node that has sent the controller hello packet. All the nodes affiliated to the controller node use the IP address contained in the *multicast_group_IP_address*. The intermediate nodes make use of the address in the *last_hop_IP_address* field for building a reverse path to the source node (i.e. the controller node). The

sequence_number field contains a number used to uniquely identify a packet from a particular source.

The Data-Reply packet which a member node sends to the controller node has the following fields in its packet header:

- *Packet_type*
- *Source_IP_address*
- *Destination_IP_address*
- *Sequence_number*
- *Next_hop_address*
- *Hop_count*
- *Time_to_live (TTL)*

The Data-Reply packet is identified by the information contained in the *packet_type* field. The *source_IP_address* field contains the IP address of the member node which has generated the packet. The IP address of the controller node which is the destination node for the packet is contained in the *destination_IP_address* field.

5.6 Design Specification of MTRDM

MTRDM is a pure on-demand route discovery method, which implies that all its procedures are invoked only when a node has data to transmit. The route request phase of MTRDM uses a Join-Query which is piggybacked in a data packet while the route reply phase uses a Join-Reply packet. The formats of the packet headers of both the Join-Query and The Join-Reply are presented in subsection 5.6.1. The operation of MTRDM is covered in subsection 5.6.2 followed by maintenance of routing information in subsection 5.6.3.

5.6.1 Join-Query and Join-Reply packet

The fields of the header of the Join-Query packet are as follows:

- *Packet_type*
- *Multicast_group_IP_address*
- *Source_IP_address*
- *Sequence_number*
- *Previous_hop_IP_address*

- *Hop_count*
- *Time_to_live (TTL)*

The *packet_type* field contains the information which is used to identify the route discovery method (i.e. MTRDM) to which the packet belongs. The *multicast_group_IP_address* is the address of the group to which the source node and the receivers belong. The address of the node that has generated the packet is contained in the *source_IP_address*. The *sequence_number* is used to uniquely identify packets coming from a particular source. Intermediate nodes use the *previous_hop_IP_address* to build a reverse path to the source as the packet propagates in a network. The number of hops traversed so far from the source node is recorded in the *hop_count* field. The *time_to_live* field contains the maximum number of hops of a packet after which the packet is dropped.

The packet header of the Join-Reply packet of MTRDM has the following fields:

- *Packet_type*
- *Acknowledgement_flag*
- *Forwarding_group_flag*
- *Multicast_group_IP_address*
- *Previous_hop_IP_address*
- *Sequence_number*
- *Source_IP_address* (1 or more)
- *Next_hop_IP_address* (1 or more)
- *Address_count*

The *packet_type* field contains information which links the packet to MTRDM. The *acknowledgement_flag* is set when an explicit acknowledgement is required. Setting of the *forwarding_group_flag* implies that the packet has been transmitted by a forwarding group node. The *multicast_group_IP_address* is the address of the multicast group to which the packet belongs. The address of the last node that has processed the packet is contained in the *previous_hop_IP_address* field. The Join-Reply packet can contain one or more *source_IP_addresses* (i.e. addresses of the multicast sources) as a way of minimising packet transmissions in a network. Consequently, the *next_hop_address* field can also contain one or more addresses. The *address_count* field contains the number of combinations of *source_IP_address* and *next_hop_IP_address*.

5.6.2 Operation of MTRDM

Operation of MTRDM is explained with the aid of a flowchart shown in Figure 5.5. As shown in Figure 5.5, MTRDM has a similar design as that of any other route discovery method of an on-demand MANET multicast routing protocol. The uniqueness of MTRDM is in the process block 3 of the flowchart in which forward nodes of the Join-Query packet are selected using the greedy set cover algorithm [7] and the partial dominant pruning technique [39].

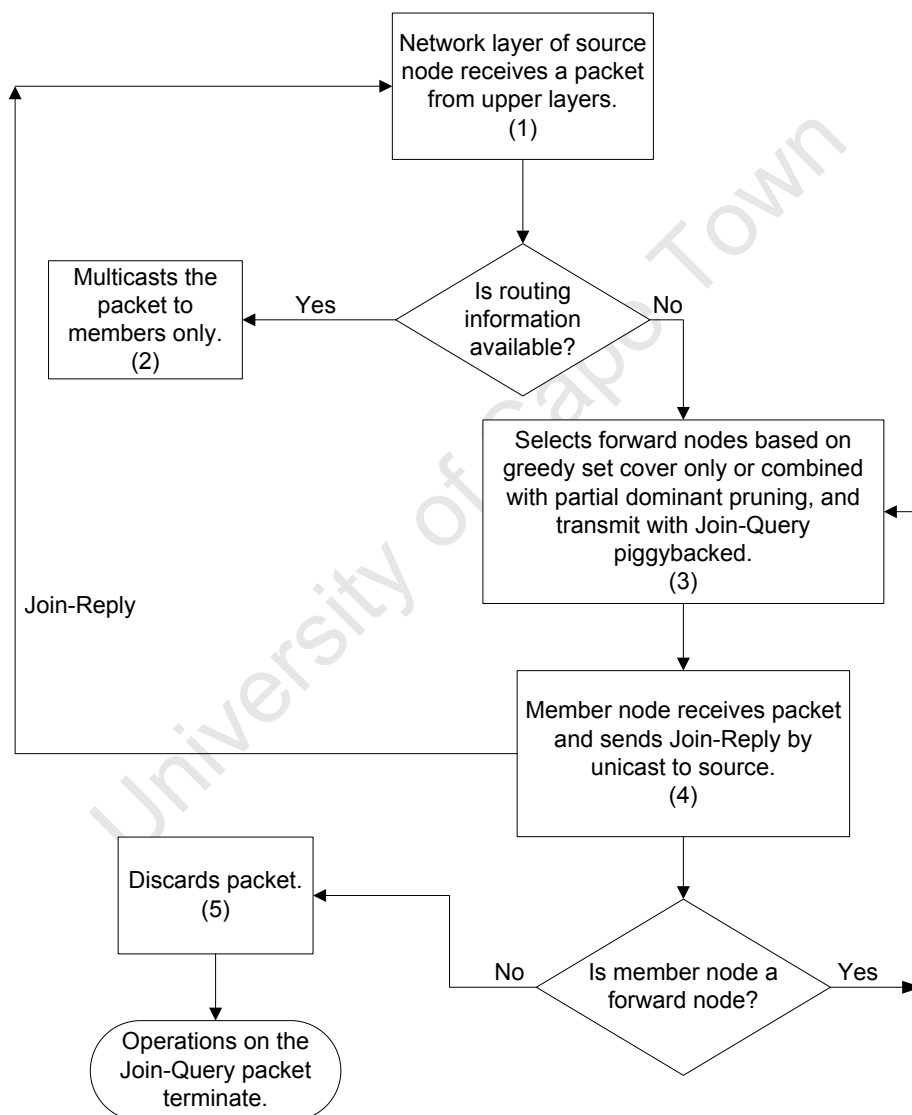


Figure 5.5 Operation of MTRDM during each Join-Query interval

The understanding in this research is that MTRDM, like any other route discovery technique used by a MANET routing protocol, has distributed characteristics in which execution of

procedures at every node is generally event-driven. Therefore, every node executes its procedures in any order as dictated by the occurrence of the events (or messages) and any such order of executions is independent of the executions concurrently taking place at the other nodes in the network. The flowchart in Figure 5.5 is not intended to illustrate the distributed nature of the executions in MTRDM. What Figure 5.5 shows is simply a sequence of major processes of MTRDM that take place anywhere in the network from the time when a source node transmits the Join-Query packet to the time when the source node gets the Join-Reply from any multicast receiver node.

The process block 1 in Figure 5.5 takes place only at a source node. Being part of a routing protocol, a route discovery operation takes place at the network layer as indicated in the process block 1. A search for a route in a network is done for a multicast packet which has already been processed by all the layers above the network layer.

When routing information for a particular multicast packet is available, the source node transmits the packet to the next hop node using a procedure in block 2. The intermediate nodes propagate the packet by using routing information available in their routing databases until each of the multicast receivers in the network gets a copy of the multicast packet.

Process block 3 is invoked at the source node when routing information for a multicast packet is not available. When a routing database does not contain information related to the next hop address of a packet under consideration or the available information has expired, the source node initiates a route discovery process. The source node first selects forward nodes among its one-hop neighbours using the greedy set cover algorithm only, piggybacks a Join-Query together with a list of the selected forward nodes in the packet, and finally transmits the packet. The greedy set cover algorithm operates by selecting a node as a forward node only if any transmission from such a node is not regarded as a redundant broadcast. The source node cannot select forward nodes using the partial dominant pruning technique because none of its neighbours or any node in the network has received the packet before. Selection of forward nodes using the greedy set cover algorithm combined with the partial dominant pruning method takes place at an intermediate node only. The partial dominant pruning algorithm is used for excluding nodes, which have already received the packet, from the list of nodes to be covered by the current transmission before applying the greedy set cover algorithm.

When a member node of a multicast group receives a broadcast packet in which a Join-Query is piggybacked, it first performs two operations as indicated in the process block 4. First, it retrieves the data contained in the packet and sends them to upper layers for further processing. Next, it prepares a Join-Reply packet and sends it to the source node using the address of the neighbour from which it first got the Join-Query as the next hop address of the Join-Reply. The Join-Reply is propagated by the intermediate nodes, using routing information stored in the routing tables during the propagation of its corresponding Join-Query, until it reaches the source node. Thus, transmission of the Join-Reply to the source node is performed by a unicast method. Finally, the member node checks whether or not it is a forward node for the received Join-Query packet.

Process block 5 is invoked when a member node that has received a Join-Query packet finds out that it is not a forward node for the packet. The termination of operations on the Join-Query packet which follows the process in block 5 is just one of the many such terminations that take place during a route discovery operation. Three other examples in which no further process takes place on the Join-Query packet are when a node receives a duplicate Join-Query packet, when the time-to-live (TTL) for the packet is zero, and when the Join-Query packet belongs to another multicast group.

Note that it is the selection of a node as a forward node which makes MTRDM different from the blind flooding method and from the other two route discovery methods of TAPMRP, namely CTRDM and RTRDM. In the blind flooding method, once a node finds out that a packet is not a duplicate and the TTL value of the packet is non-zero, it broadcasts the packet. Thus, there is no special selection of a node as a forward node. The methods used by CTRDM and RTRDM for selecting forward nodes are covered in the subsequent sections.

Like the blind flooding method of ODMRP, MTRDM uses the passive acknowledgement technique for ensuring reliable transmission of the Join-Reply packet. The passive acknowledgement technique makes use of promiscuous reception of packets or overhearing of packet transmissions. When node v transmits a packet to its upstream neighbour of the route to the source node, its downstream neighbour can overhear the transmission of the packet and conclude that the packet was delivered to node v . Since a source node cannot propagate a Join-Reply any further unless it is also a forwarding group node for other sources, it sends an explicit acknowledgement to the previous hop node of the packet. When a node detects the absence of an acknowledgement (passive or explicit) after a timeout

period, it retransmits the Join-Reply packet. The retransmission of the Join-Reply packet is repeated for a specified number of times in accordance with the specification of ODMRP [9] in the event of having no acknowledgement for each retransmission attempt.

5.6.3 Routing information maintained by MTRDM

MTRDM uses shared routing information with the other route discovery methods of TAPMRP. TAPMRP maintains three tables for routing purposes. Every node running TAPMRP maintains a routing table. Propagation of the Join-Reply to a multicast source node uses information stored in the routing table.

Another table maintained by TAPMRP is the forward group table which keeps information about the membership of a multicast group. The forward group table is maintained by a forward group node. In particular, the table stores multicast group addresses for which the node is their forward group node and the time when such information was last refreshed.

A message cache is kept by each node running TAPMRP for the purpose of detecting duplicate packets. As a way of preventing the cache from growing too big, TAPMRP uses the FIFO (First In First Out) queue for the implementation of the message cache. Hence, messages in the cache are not kept permanently. Old (or expired) messages in the cache are removed and new ones are added to it on a regular basis.

5.7 Design Specification of CTRDM

CTRDM is a hybrid route discovery method which has two components: a proactive and a reactive component. The proactive part of CTRDM is provided by the active clustering platform which maintains clusters throughout the execution period of CTRDM. The route request phase and the route reply phase of CTRDM, which are executed only when a node has data to transmit, form the reactive (or on-demand) component of CTRDM. Like MTRDM, the route request phase of CTRDM uses a Join-Query packet while a Join-Reply packet is used for the route reply phase. The packet header fields of the Join-Query and the Join-Reply of CTRDM are similar to those of MTRDM.

The propagation of the Join-Query and the Join-Reply in CTRDM is explained with the help of the flowchart of Figure 5.6. The design of CTRDM, as shown in Figure 5.6, is similar to that of any other route discovery method of an on-demand MANET multicast routing

protocol. One of the unique features of CTRDM is in the process block 3 of the flowchart in which only cluster-heads select forward nodes of the Join-Query packet using the greedy set cover algorithm. Another unique feature of CTRDM is the use of an underlying clustering algorithm which is not shown in the flowchart. Like Figure 5.5, Figure 5.6 shows a sequence of only major processes of CTRDM that take place anywhere in the network from the time when a source node transmits the Join-Query packet to the time when the source node gets the Join-Reply from any multicast receiver node.

Only a source node executes the process block 1 of the flowchart in Figure 5.6. When routing information for a particular multicast packet is available, the cluster-head of the source node or the source node itself if it is a cluster-head transmits the packet to the next hop node using a procedure in block 2. The intermediate nodes propagate the packet by using routing information available in their routing databases until each of the multicast receivers in the network gets a copy of the multicast packet.

Process block 3 is invoked only at a cluster-head when routing information for a multicast packet is not available or has expired. Thus, block 3 initiates a route discovery operation. The cluster-head first selects some bridge nodes as forward nodes among its one-hop neighbours using the greedy set cover algorithm, piggybacks a Join-Query together with a list of the selected forward nodes in the packet, and finally transmits the packet. The greedy set cover algorithm operates by selecting a node as a forward node only if any transmission from such a node is not regarded as a redundant broadcast.

When a multicast member node receives a broadcast packet in which a Join-Query is piggybacked, it first performs two operations as indicated in the process block 4. The first operation is that of retrieving the data contained in the packet and sends them to upper layers for further processing. Next, it prepares a Join-Reply packet and sends it to the source node using the address of the neighbour from which it first got the Join-Query as the next hop address of the Join-Reply. The Join-Reply is propagated by the intermediate nodes until it reaches the source node. Finally, the member node uses information carried in the received Join-Query packet to check whether or not it is a forward node.

In the process block 5, a multicast member node discards a Join-Query packet if it finds out that it is not a forward node for the packet. Note that duplicate packets and those that stay

beyond their time-to-live are also destroyed anywhere in the network. Process block 6 is executed at a bridge node only.

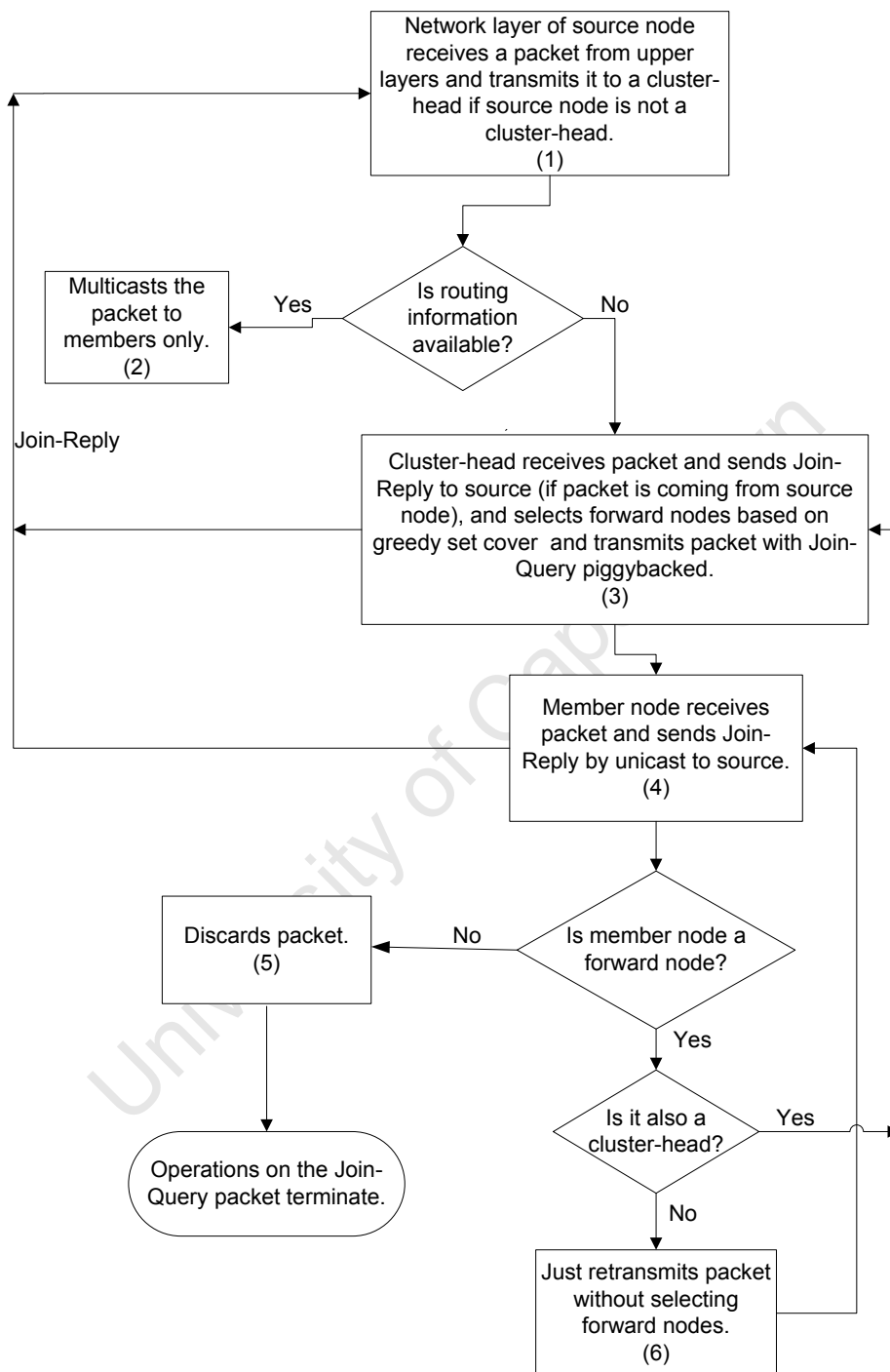


Figure 5.6 Operation of CTRDM during each Join-Query interval

In the forward node selection algorithm of CTRDM, all cluster-heads are selected as forward nodes but not all gateway nodes are selected as forward nodes. CTRDM uses shared routing

databases just like the other two route discovery methods of TAPMRP. When CTRDM is running it can remove and add entries to the shared tables maintained by TAPMRP.

5.8 Specification of RTRDM

RTRDM is a pure on-demand route discovery method just like MTRDM. This means that RTRDM is invoked only when a node has data to transmit. The route request phase of RTRDM uses the Join-Query which has the same packet header fields as those of MTRDM and CTRDM. RTRDM also uses the Join-Reply for the route reply phase which has the same packet header fields as those of MTRDM and CTRDM.

The Join-Query in RTRDM is piggybacked in a data packet and broadcast to an entire network just like in MTRDM and CTRDM. However, RTRDM uses passive clustering as its broadcasting method. Processing of the Join-Query at every node is the same as in MTRDM and CTRDM with the exception that in RTRDM only cluster-heads and gateway nodes are allowed to act as forward nodes for the packet. Note that in the active clustering of CTRDM not all gateway nodes are allowed to forward a Join-Query packet while in the passive clustering of RTRDM all gateway nodes are selected as forward nodes. Propagation of the Join-Reply to a multicast source in RTRDM is exactly the same as in MTRDM and CTRDM. Like MTRDM and CTRDM, RTRDM uses shared routing databases. RTRDM is allowed to remove and add new entries to the tables used for routing purposes maintained by TAPMRP.

The operation of RTRDM has a design similar to that of CTRDM shown in Figure 5.6. The main difference between the designs of RTRDM and CTRDM is that in RTRDM cluster-heads do not select forward nodes as the case is in process block 3 of Figure 5.6 but they just retransmit (or forward) the Join-Query, and that every bridge node in RTRDM is allowed to forward the Join-Query. Details on the design specification of the passive clustering method used by RTRDM are covered in [9, 100, 101].

5.9 Ensuring a Seamless Route Discovery Method Switch

A route-discovery-method-switch announcement from the controller node may not reach every member node at the same time due to delays encountered along the way. The design of TAPMRP takes into consideration the effect of the propagation delay of the switch announcement on the operation of TAPMRP during the period of switching from one route discovery method to another. The design approach is that of ensuring that all the three route

discovery methods are able to operate concurrently without causing problems to on-going traffic in the network. In practice we expect only two route discovery methods to operate concurrently for a short time during the switch-over period. Having three route discovery methods operating concurrently is a rare event, which may happen due to some nodes failing to receive two consecutive announcements from the controller node. However, the design approach considers the case of all the three route discovery methods operating concurrently as a way of preparing for any possible event in a network.

The design of TAPMRP ensures a seamless switch from one route discovery method to another by using a two-step approach. First, when a node generates a control packet it places the route discovery method information related to the packet in a header field of the packet, and this enables each node in the network to treat each control packet according to the route discovery method used by the source of the packet. When the route discovery method carried in the control packet matches with the route discovery method kept by an intermediate node, the packet is forwarded by the intermediate node using the common route discovery method. The intermediate node uses MTRDM for forwarding a control packet if the route discovery method carried in the packet is different from the one kept by the intermediate node. MTRDM is chosen as the default route discovery method because of its implementation simplicity.

The second step of the approach aimed at ensuring a seamless switch between route discovery methods is accomplished through the use of shared routing information (i.e. both the routing databases and the information stored in the databases). This implies that as long as the required routing information is available a packet uses the information for its propagation regardless of which route discovery method provided the information. Hence the withdraw of a particular route discovery method has minimal effect on the on-going control packets as well as data packets since the information provided by the withdrawing route discovery method is left untouched in the routing databases until after a timeout period.

In general, transmission of data packets in TAPMRP does not depend on a particular route discovery method. Transmission of data packets uses shared routing databases. When routing information is not available, a route request process is initiated using a specific route discovery method. All the route discovery methods of TAPMRP broadcast their Join-Queries by piggybacking them in data packets. Therefore, the transmission of the data packets which

piggyback control information is dependent on the route discovery method used for the transmission.

5.10 Implementation of TAPMRP

The development of TAPMRP ended with an implementation which involved development of code. The code was used for simulations which were aimed at testing the feasibility of using TAPMRP's ideas in practice. GloMoSim-2.03 [5] was used for the simulations which were based on extending the functionality of ODMRP [9]. The simulations focussed on all the major ideas of TAPMRP which were developed in this research. The tested ideas include the use of the standard deviation of the number of neighbours of nodes in a network as a mechanism for detecting the three phases of the ELC-MANET, the introduction of secondary cluster-heads in a one-hop clustering scheme, the use of the node redistribution parameter (i.e. Equation (4.2)), and the use of the node mobility formula (i.e. Equation (4.4)).

The implementation of TAPMRP makes use of two types of a periodic hello message: a neighbour hello message and a controller hello message. The neighbour hello message is generated and transmitted by a function called *TapmrpSendHello()*, while the controller hello message is generated and transmitted by *TapmrpSendControllerHello()*. *TapmrpSendHello()* is executed at every node in a network and is used to send a hello message to one-hop neighbours only after every 30 seconds of the GloMoSim-2.03 clock. The interval of 30 seconds between two consecutive neighbour hello messages which may appear to be relatively too long was chosen based on the understanding that mobility of nodes in the ELC-MANET is generally low, and therefore it is better to have a relatively low frequency of generating neighbour hello messages in order to avoid excessive control overhead.

Figure 5.7 shows the functions of TAPMRP that are executed following the invocation of *TapmrpSendHello()* by an in-built timer function of GloMoSim-2.03. Before a hello message is sent out, *TapmrpSendHello()* computes a node weight by using the node redistribution parameter value and the node stability value as inputs. The node redistribution value is determined by *TapmrpSendHello()* itself through computation of Equation (4.2) while the node stability value is obtained from a value returned by *TapmrpDetermineMobilityLevel()* which implements Equation (4.4). The resulting node weight and a list of one-hop neighbours of the node running *TapmrpSendHello()* are then put in a hello message, which is sent out at the end of the execution of *TapmrpSendHello()*. Upon receiving a hello message, a node runs

TapmrpHandleHello() which updates a list of its one-hop neighbours and a list of its neighbours located within two-hops.

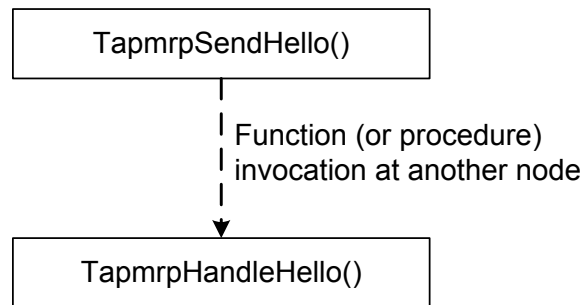


Figure 5.7 Block diagram of TAPMRP’s functions related to the neighbour hello message

The execution of *TapmrpSendControllerHello()* by a controller node sends out a controller hello message to a multicast group after every 120 seconds of the GloMoSim-2.03 clock. The reason for choosing a controller hello interval of 120 seconds is similar to that of the neighbour hello message interval. Figure 5.8 shows the execution flow involving the main functions of TAPMRP that are executed following the invocation of *TapmrpSendControllerHello()* by an in-built timer function of GloMoSim-2.03.

Upon receiving a controller hello message, a node executes *TapmrpHandleControllerHello()*, which in turn runs *TapmrpSendControllerReply()*, *TapmrpDetermineMobilityLevel()*, and *TapmrpInitClusterCTRDM()* if CTRDM is the currently announced route discovery method by the controller node. Every node in a network, on receiving a controller reply message, runs *TapmrpHandleControllerReply()*. However, only a controller node runs *TapmrpComputeStdev()* following the execution of *TapmrpHandleControllerReply()*. It is *TapmrpHandleControllerReply()* which implements the route-discovery-method switch of TAPMRP. Results returned by *TapmrpComputeStdev()* are used for making route discovery switching decisions by the controller node.

Execution of *TapmrpInitClusterCTRDM()* invokes a series of functions as shown in Figure 5.8 that creates a clustering platform. Appointment of secondary cluster-heads by a primary cluster-head is implemented in *TapmrpChMessageCTRDM()*, and is done before a cluster-head announcement message is sent out. Note that the clustering platform component of TAPMRP shown in Figure 5.2 is implemented by the functions shown in Figure 5.8. Also

note that the functions shown in Figure 5.7 and 5.8 are not part of the flowchart processes shown in Figure 5.5 and 5.6 which deal with the Join-Query.

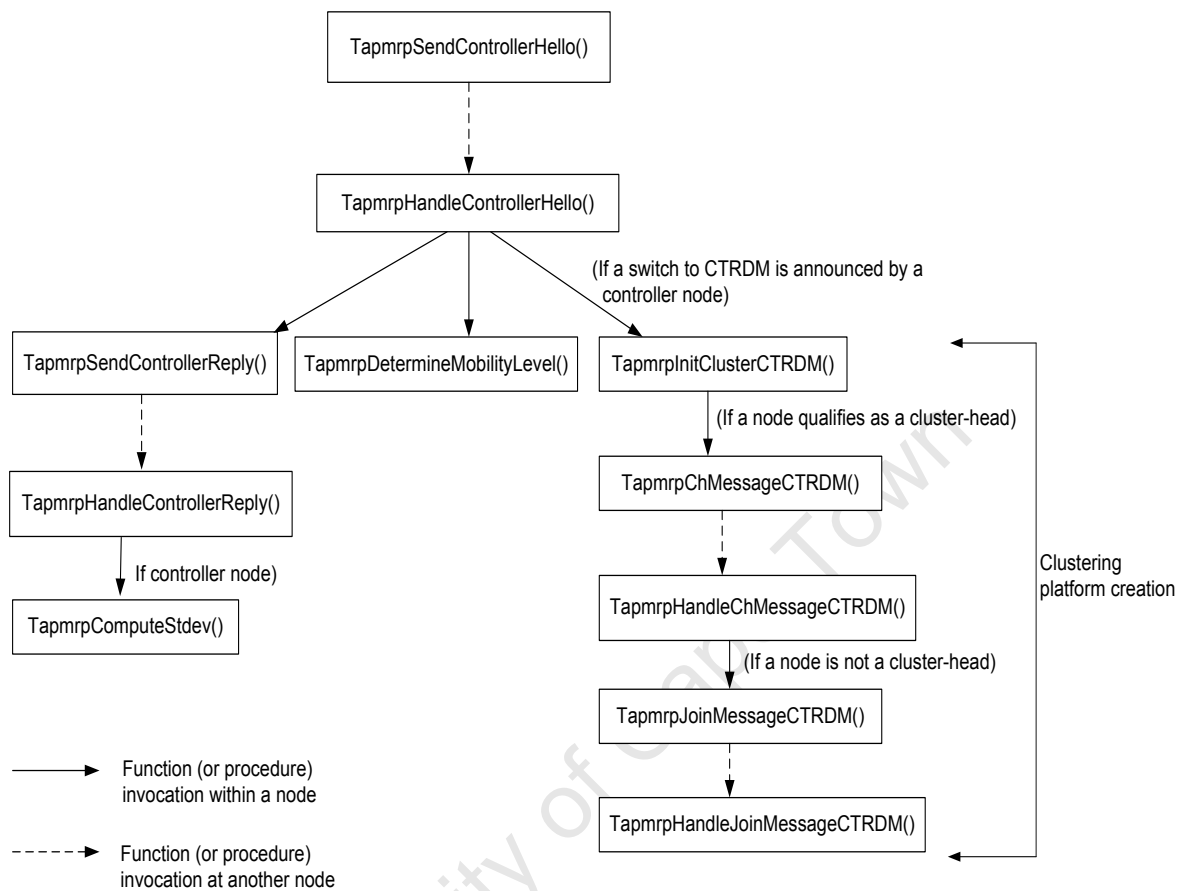


Figure 5.8 Block diagram of TAPMRP's functions related to the controller hello message

When a source node of multicast packets wants to send out a Join-Query for route discovery, it starts by executing *RoutingOdmrpSendQuery()* shown in Figure 5.9. Before the Join-Query is sent out, a source node selects a set of forward nodes from its list of one-hop neighbours using *TapmrpSelectFwdNodesMTRDM()* if MTRDM is the current route discovery method used by the source node. If CTRDM is the current route discovery method, the source node, if it is not a cluster-head, simply transmits the Join-Query to its cluster-head without selecting forward nodes. The cluster-head then selects forward nodes using *TapmrpSelectFwdNodesCTRDM()*. RTRDM does not have an explicit way of selecting forward nodes by way of using a special function which implements a particular pruning technique. Instead, RTRDM uses every cluster-head and bridge node as a forward node. In Figure 5.9, RTRDM uses a piece of code incorporated in *RoutingOdmrpHandleJoinQuery()* to identify cluster-heads and bridge nodes by a passive clustering technique.

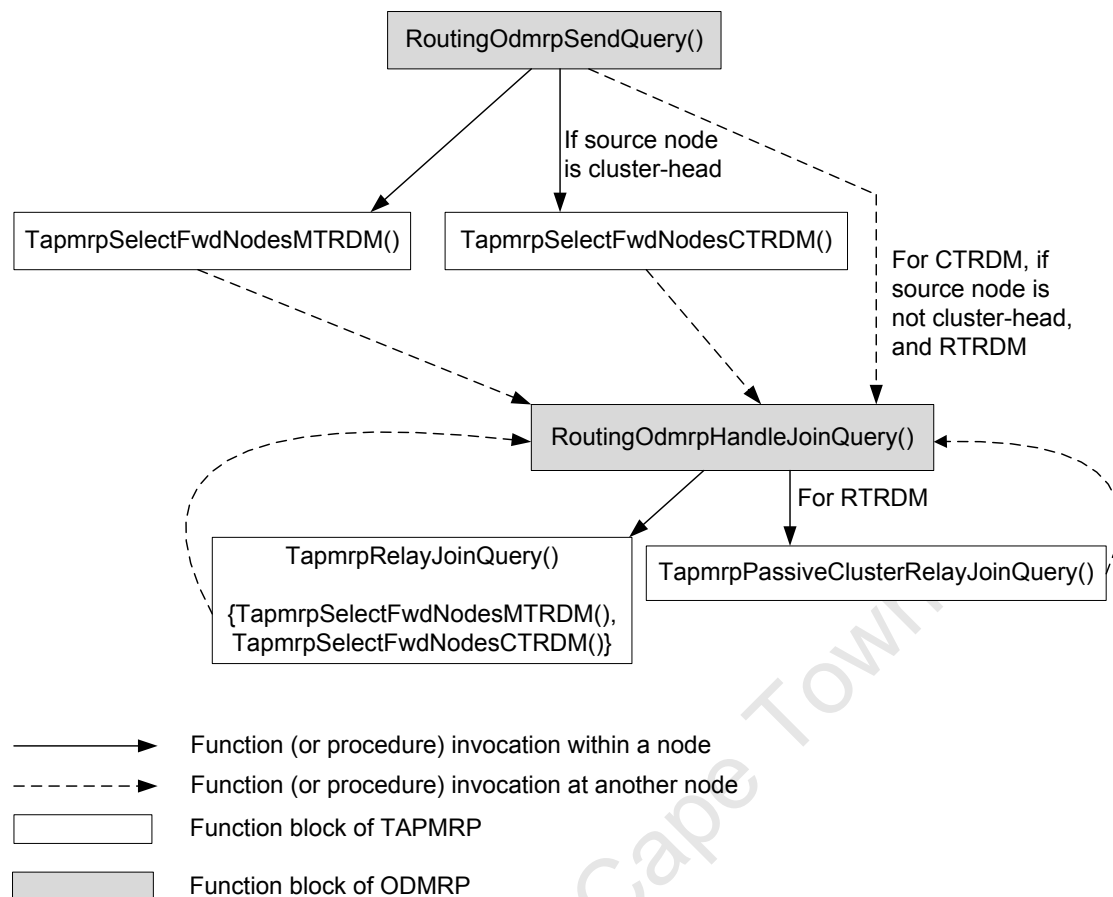


Figure 5.9 Block diagram of functions of TAPMRP related to the Join-Query

Note that the process block 3 of the flowchart of Figure 5.5 is implemented by *TapmnpSelectFwdNodesMTRDM()* in Figure 5.9, while the process block 4 that deals with the sending of the Join-Reply to the source node is implemented by some lines of code in *RoutingOdmrpHandleJoinQuery()* of Figure 5.9. Similarly, the process block 3 of the flowchart of Figure 5.6 is handled by *TapmnpSelectFwdNodesCTRDM()* in Figure 5.9, while the process block 4 is done by the same lines of code in *RoutingOdmrpHandleJoinQuery()* that implement the process block 4 of Figure 5.5. Also note that the process block 2 appearing in the flowcharts of Figure 5.5 and 5.6 is handled by the data multicasting module of ODMRP (refer to Figure 5.2), which is not part of the code developed for TAPMRP.

The experimental designs of the simulations are presented in the methodology chapter of this thesis. The results of the simulations are covered in the results chapter that follow the methodology. Note that the simulations focussed on a one multicast group scenario in which all the nodes of a simulated network were considered as members of one multicast group. The compact disc which accompanies this thesis contains the entire code of the simulations. The

Appendix of this thesis presents pseudo code of some of the functions of TAPMRP which were used for the simulations. Almost all the functions of TAPMRP shown in Figure 5.8 and 5.9 appear in the Appendix in the form of pseudo code.

5.11 Summary

Presented in this chapter is a design specification of TAPMRP which is the proposed solution to the problem of this research undertaking. TAPMRP is an adaptable hybrid multicast routing protocol. The adaptability of TAPMRP comes from its ability to switch dynamically between MTRDM, CTRDM and RTRDM depending on the prevailing ELC-MANET topology. TAPMRP is a hybrid protocol because it has two pure on-demand route discovery methods, namely MTRDM and RTRDM, and a route discovery method (i.e. CTRDM) which has a proactive component (i.e. the active clustering platform).

A multicast group served by TAPMRP has a multicast controller node which acts as the leader of the multicast group. The first node to become a member of a multicast group takes up the role of the multicast controller. The multicast controller node periodically sends out controller hello messages. A node wanting to join the multicast group simply responds to the controller hello messages by sending reply messages back to the controller node. When a node wants leave the multicast group, it stops responding to the controller hello messages.

Maintenance of local connectivity through exchanging of neighbour lists accomplished through hello messages is very important in TAPMRP. Every node in TAPMRP is required to periodically send a list of its one-hop neighbours to its neighbours. This exchange of information enables every node to build a list of its two-hop neighbours. Both MTRDM and CTRDM require two-hop neighbour information for their operation. However, RTRDM does not require exchanging of neighbour information because it uses passive clustering which can operate without relying on maintenance of neighbourhood information.

Most of the network-wide operations in TAPMRP are coordinated by the route-discovery-method switch, which resides in a multicast controller node. The route-discovery-method switch has three major sub-procedures. The first sub-procedure is the standard-deviation procedure which collects data related to the number of neighbours of a node from every multicast member and computes the standard deviation of the number of neighbours of nodes in the network. The results from the standard-deviation procedure are used in the second sub-

procedure of the route-discovery-method switch for determining whether a switch to another route discovery method of TAPMRP should take place or not. The third sub-procedure of the route-discovery-method switch is responsible for announcing the decision made by the second sub-procedure to the rest of the members of a multicast group.

This chapter also covers the design specifications of MTRDM, CTRDM and RTRDM. All the three route discovery methods of TAPMRP use the same formats for the Join-Query and the Join-Reply. The three route discovery methods also use the same routing tables for propagation of the join-Reply packet to the multicast source node. The main difference between the three route discovery methods is in the broadcasting methods used for transmitting the Join-Query packet. MTRDM uses a broadcasting method based on the partial dominant pruning technique [39] for selection of forward nodes. CTRDM uses a cluster-based broadcasting technique in which cluster-heads and a subset of gateway nodes are used as forward nodes for the Join-Query packet. The broadcasting method used in RTRDM makes use of passive clustering in which all cluster-heads and gateway nodes act as forward nodes for a broadcast packet.

The design of TAPMRP has taken into consideration the need to provide a seamless switch from one route discovery method to another. The approach used to provide a seamless switch is that of ensuring that all the three route discovery methods, namely MTRDM, CTRDM and RTRDM, or any two of them, are able to operate concurrently without negatively affecting the flow of traffic in a network.

The chapter ends with a description of the implementation of TAPMRP. The implementation involved development of code which was used to test TAPMRP in the form of computer simulations that were conducted in GloMoSim-2.03. The simulations focussed on all the major ideas of TAPMRP which were generated in this research.

CHAPTER 6

METHODOLOGY

6.1 Introduction

The experimental part of this research was undertaken in three stages. The work in the first stage was on the validation of the existence and the significance of the research problem. In the second stage, the research focused on the separation of the ELC-MANET into three distinct phases each of which is characterised by a distinct topology. The third stage of the research experiments dealt with the testing of TAPMRP which is proposed to serve the phased MANET of the ELC environment.

All the major ideas generated in this research were tested by computer simulations using GloMoSim-2.03 [5]. GloMoSim is a network simulator which was specifically developed for wireless networks. The computer simulation language used by GloMoSim is PARSEC [159] which is a discrete-event simulation language based on C programming language. PARSEC stands for PARallel Simulation Environment for Complex systems.

The experimental design of each simulation made use of the assumptions used by GloMoSim-2.03. Four of such assumptions are as follows: all the nodes have the same transmission range, all the nodes are homogeneous (i.e. of the same type), a shared transmission medium is used by all the nodes in a network, and transmission of packets is bidirectional.

The simulations made use of the receiver sensitivity threshold settings for varying the connectivity level of a network. The reasoning in this research is that the same effect of varying connectivity could have been achieved by varying the transmission range and also by varying the transmitting power of the nodes. Another point to note is that a particular connectivity level, whether it is low or high, can occur at any receiver sensitivity threshold setting because connectivity level, among others, depends on both the receiver sensitivity threshold setting and the geographical distance between any two communicating nodes. Hence, it is possible to have low connectivity at -90dBm or -36dBm; or high connectivity at -

90dBm or -36dBm. This reasoning on network connectivity is intended to provide the explanation for the use of receiver sensitivity threshold values in some simulations that are possibly much higher or much lower than those that are associated with typical MANET devices in practice.

The ranges of the receiver sensitivity threshold settings used in the simulations were imposed on the simulations by a combination of the geographical network areas and the distances between nodes that were used. For example, a network area of $80\text{m} \times 80\text{m}$ used by the ELC-cluster-based topology resulted in having around -65.5dBm as the lowest receiver sensitivity threshold setting that yielded a completely connected network in which each node had all the other nodes as its one-hop neighbours, and -59.5dBm as the highest setting that yielded a fully connected network beyond which the network became disconnected.

The network areas used in the simulations, namely $20\text{m} \times 20\text{m}$ for the ELC-matrix topology, $80\text{m} \times 80\text{m}$ for ELC-cluster-based topology, and $100\text{m} \times 100\text{m}$ for the ELC-random topology are considered in this research as the network areas that can occur on average in the ELC environment. While the area of the ELC-matrix topology may not extend much further than the suggested average value, areas of the ELC-cluster-based topology and the ELC-random topology can extend much further in practice. For example, the ELC-random topology on a university campus can cover a diameter of more than 500m, and in this case the lowest setting of the receiver sensitivity can be much lower than -65.5dBm.

Note that in the simulations of this research, the number of forward nodes of a broadcast packet (i.e. a control packet) is considered as the main performance metric which is used to address the problem as well as the hypothesis of this research presented in chapter 1. The conventional performance metrics such as latency, jitter, and throughput are of secondary importance in this research because the understanding is that the use of such metrics has more significance in the data traffic, which is outside the scope of this research, than in the control (or signalling) traffic, which is within the scope of this research. The simulations use the packet delivery ratio to measure performance associated with data traffic where it is necessary to consider data traffic.

This chapter starts with the methodology used to validate the research problem in section 6.2. Section 6.3 covers the experimental designs that were used to formulate a mechanism for detecting the three phases of the ELC-MANET. Thereafter, the chapter presents

methodologies used to test various ideas related to the proposed route discovery methods of TAPMRP.

6.2 Validation of the Research Problem

The validation test on the research problem was aimed at confirming by simulation the existence and the significance of the problem of this research undertaking. The use of a computer simulation to validate the research problem was based on the reasoning that broadcast redundancy during a route discovery operation is a major problem which makes existing multicast protocols less efficient in ELC-MANETs. Most of the current multicast routing protocols including ODMRP use blind flooding for broadcasting Join-Query packets during the route request phase. The blind flooding method produces a lot of redundant retransmissions which waste energy and bandwidth.

The simulation test was conducted in two stages. Some of the simulation settings were common in the two stages of the simulation. Examples of the common settings were as follows:

- Nodes were placed using the random placement method provided by GloMoSim-2.03.
- Transmitter power was set at 15 decibel-milliWatt (dBm), which is about 32 milliWatts.
- Dimensions of the area of the simulated network were fixed at 100m × 100m.
- The radio propagation model used was the two-ray (or ground reflection) model.
- Bandwidth capacity of the transmission medium was set at 2Mbps.

In the first stage of the simulation, the network size was fixed at 50 nodes. The receiver sensitivity threshold was varied from -70dBm to -56dBm, resulting in indirectly varying the transmission range of the nodes. One of the 50 nodes of the simulated network was arbitrarily selected to act as a multicast source node of CBR traffic while the rest of the 49 nodes were receivers. The session of CBR traffic flow was fixed at a maximum duration of 3 seconds of the GloMoSim clock, which resulted in the generation of only one Join-Query packet in all the runs of the first stage of the simulation. A simulation measurement taken corresponding to each receiver sensitivity threshold setting was the number of nodes in the network that acted as forward nodes for the Join-Query packet.

The second stage of the simulation involved variation of the network size from 10 nodes up to 50 nodes with a spacing of 5 nodes between adjacent network sizes while the receiver sensitivity threshold was fixed at -70dBm. The setting of the receiver sensitivity threshold at -70dBm resulted in the formation of highly connected networks in which a source node of CBR traffic in each of such simulated networks had the rest of the nodes in the network as its one-hop neighbours. The simulation preferred such highly connected network scenarios because they clearly showed cases in which retransmissions performed by the blind flooding method of broadcasting were seen to be unnecessary. The maximum duration of CBR traffic flow was set at 3 seconds just like in the first stage of the simulation. The data collected during the second stage of the simulation were the number of forward nodes of the Join-Query packet for each network size that was tested.

6.3 Detection of the ELC-MANET Phases

The aim of the second stage of the experiments of this research was to formulate a computationally efficient mechanism for detecting the three phases of the ELC-MANET. As mentioned in chapter 3, the proposed mechanism for detecting dynamically the three phases of ELC-MANET is based on the computation of the standard deviation of the number of neighbours of nodes in the ELC-MANET. The standard deviation mechanism was formulated using a computer simulation which was conducted in GloMoSim-2.03 [5]. The methodology employed by the simulation involved the construction of model topologies of each of the three phases of the ELC-MANET. The standard deviation of the number of neighbours of nodes in each topology was determined with respect to its variation with the transmission range and the network size.

The methodology used to formulate the proposed mechanism for identifying the topologies of the ELC-MANET is covered in subsection 6.3.1. Subsection 6.3.2 presents a procedure used to test the validity of the online algorithm for computing the standard deviation proposed in this thesis (refer to subsection 3.3.2 of chapter 3 and subsection 5.5.1 of chapter 5). The last subsection of this section presents a justification for separating the ELC-MANET into phases.

6.3.1 Formulation of the standard-deviation mechanism

To validate the hypotheses stated in section 3.3 of chapter 3, the following facts were considered:

- The ideal validation procedure is to consider *stdev* for all possible matrix, cluster-based and random topologies in ELC.
- But it is practically impossible to produce *stdev* for all possible cluster-based and random topologies even for a small network.
- However, for small networks, the symmetry of the matrix topology makes it possible to study *stdev* of all the possible matrix topologies.
- *stdev* depends on the transmission range of nodes and the size of the network.

Dependence of *stdev* on the transmission range

The dependence of *stdev* on the transmission range was explored by using a matrix topology of size 55 with nodes arranged as illustrated in Figure 6.1 and an arbitrary random topology also having 55 nodes. The methodology was based on the major assumption already built in GloMoSim that every node in a simulation network has the same transmission range and receiver sensitivity. Transmitter power was fixed at 15 decibel-milliWatts (dBm) (a default value) while receiver sensitivity was varied from -52dBm to -38dBm, effectively varying the transmission range.

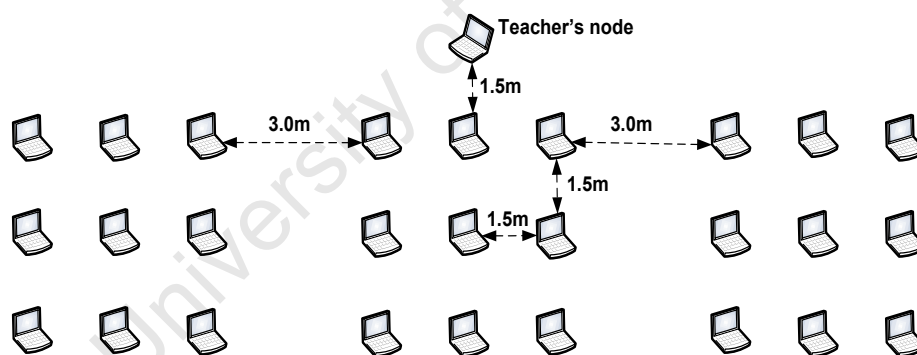


Figure 6.1 Example: $(3 \times 9) + 1$ ELC-matrix topology.

Variation of *stdev* with the network size

The research into *stdev*'s dependence on the network size started with the definitions of model ELC matrix and ELC cluster-based topologies. The ELC matrix topology was defined as an $M \times N$ matrix of nodes with the distance between nodes as shown in Figure 6.1, which is an example of such a matrix.

Eighteen network sizes spaced at an interval of three nodes between consecutive nodes were considered: 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52 and 55. For each

network size k , all the $(M \times N) + 1$ matrices formed from $k - 1$ were investigated and the matrix which gave the highest standard deviation of the number of neighbours was taken to be a representative matrix for the network size k and its *stdev* became the *stdev* for the network size in the data analysis. For instance, the network size $k = 31$ had the following as its $(M \times N) + 1$ matrices: $(1 \times 30) + 1$, $(2 \times 15) + 1$, $(3 \times 10) + 1$ and $(5 \times 6) + 1$. To determine the highest *stdev* for a particular matrix such as $(2 \times 15) + 1$, the transmission range of the nodes was varied over the whole range of values which could give practically useful results.

The investigation of *stdev*'s variation with the network size for the ELC cluster-based topologies started with the definition of the topologies. Each cluster-based topology was allowed to have only three clusters, and all the three were different with the first being circular, the second one matrix and the third one random in shape. An example of the ELC cluster-based topology is shown in Figure 6.2. Even though the model ELC cluster-based topology has only three clusters, the research assumes that the results obtained are applicable to the ELC cluster-based topology with any number of clusters.

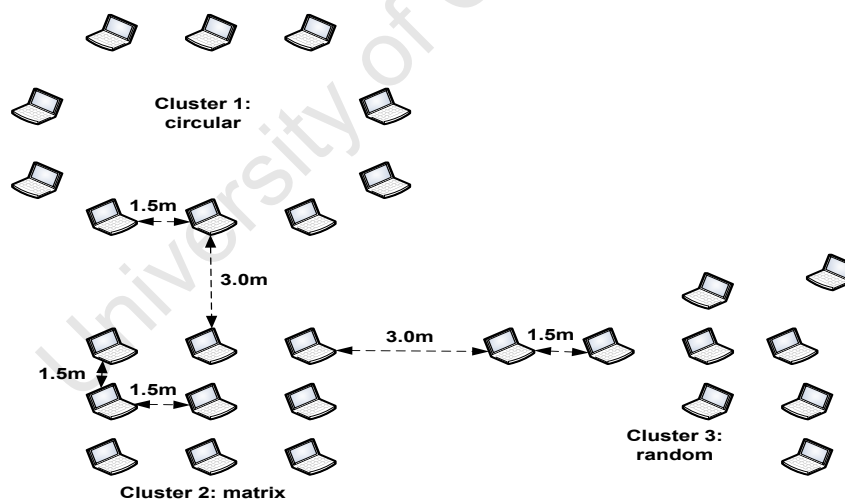


Figure 6.2 Example: ELC-cluster-based topology with 28 nodes.

The lowest network size considered by the model network was 10, while higher network sizes were obtained by adding a node to each cluster resulting in a three-node interval between consecutive network sizes. The ten-node cluster-based topology is shown in Figure 6.3. The minimum distance between clusters was maintained at 3.0m as indicated in Figure 6.2 and 6.3.

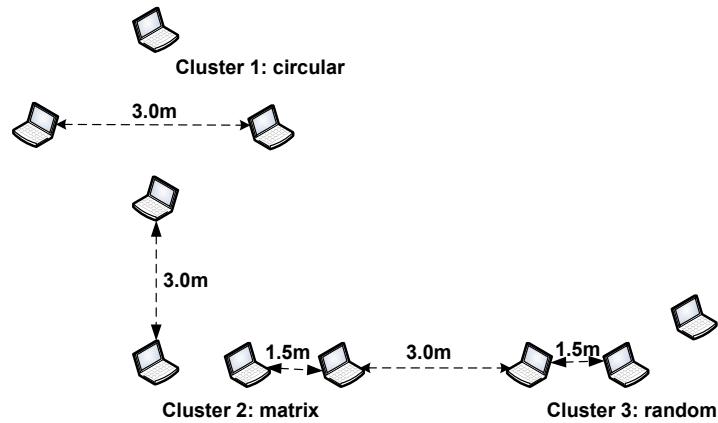


Figure 6.3 Example: ELC-cluster-based topology with 10 nodes.

In total, the following network sizes of the ELC cluster-based topologies were investigated: 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, and 55. A cluster-based topology for the network size of 28 is shown in Figure 6.2 as an example of a cluster-based topology with a number of nodes greater than 10. Again, in this part of research the highest *stdev* of a particular topology was obtained by varying the transmission range.

Figure 6.4 is an example of the ELC-random topology. Random topologies were not considered in this part of research on the dependence of *stdev* on the network size because the values of *stdev* for the cluster-based topologies were considered as forming a line of demarcation between random topologies and cluster-based topologies (covered in the results chapter). The demarcation line meant that cluster-based topologies have their highest values of *stdev* close to the line while random topologies have their lowest values also close to the same line but on the other side.

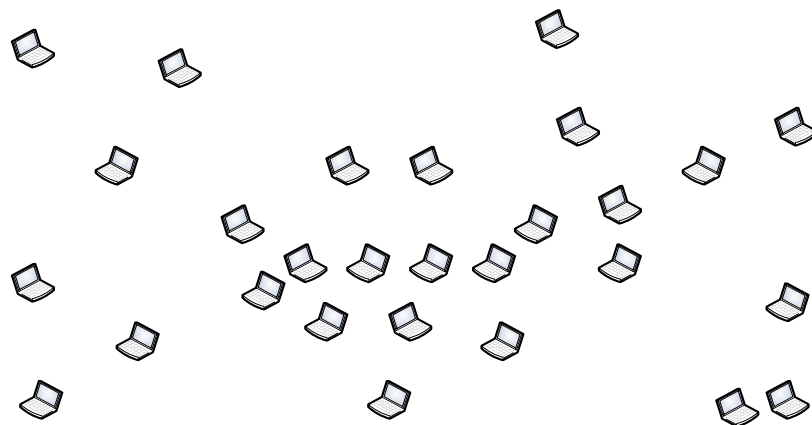


Figure 6.4 Example: ELC-random topology with 28 nodes.

6.3.2 Online algorithm for computing standard deviation

In this section, we look at the validity of the online method used by TAPMRP to compute the standard deviation of the number of neighbours of nodes in the ELC-MANET. As explained in subsection 3.3.2 of chapter 3, the justification behind the usefulness of the algorithm is based on the understanding that what matters most is that the algorithm should be capable of generating the same standard deviation values for the same network topology and size while operating under different situations in terms of factors such as place, purpose, and devices. Based on this understanding, a simulation to confirm the validity of the online algorithm involved running the algorithm on two different computers: a laptop computer and a desktop computer. The validation test had two aims. The first aim was to find correlation between the standard deviation values obtained from the two different computers. The second aim of the test was to determine the magnitude of the error associated with the proposed online algorithm.

The specifications of the laptop and the desktop computer used in the investigation are shown in Table 6.1. Note that a desktop computer is not a good example of a MANET device since it is not portable; however, it was used in this part of research only to play the role of having a device different from the laptop computer.

Table 6.1 Specifications of a laptop and a desktop computer

Component	Laptop specifications	Desktop specifications
Operating System	Windows 7 Home Basic	Windows XP Professional version 2002 Service Pack 2
Processor	Celeron(R) Dual-Core CPU T3100@ 1.90GHz	Pentium(R) 4 CPU 2.80GHz
RAM	2.00GB	1.98GB
System type	32-bit Operating System	32-bit Operating System
Manufacturer	Samsung Electronics	Philips

The validation approach used was that of testing the online algorithm before implementing it in TAPMRP. Hence, the algorithm was implemented as a standalone PARSEC [159] program. The data used to test the algorithm were arbitrarily chosen from one of the large data sets generated in subsection 6.3.1. The data set had 55 elements and is presented in Table 6.2.

Table 6.2 Data set for testing the standard deviation online algorithm

Data set
{20, 21, 21, 23, 25, 32, 35, 40, 44, 50, 46, 44, 44, 44, 40, 35, 31, 27, 24, 40, 47, 43, 36, 39, 42, 35, 36, 40, 32, 33, 38, 29, 29, 35, 25, 25, 32, 44, 47, 43, 39, 43, 32, 35, 37, 37, 30, 32, 31, 25, 22, 22, 21, 18, 16}

Note that each element in the data set gives the number of neighbours of a particular node in a network. The validation test was conducted by operating the algorithm on 10 different data subsets drawn from the data set given in Table 6.2. The first data subset used consisted of the first 10 elements of the data set, while the second subset consisted of the first 15 elements of the same data set. The rest of the data subsets used in the tests were obtained in a way similar to that of the first and second subsets in which elements were drawn from the data set and placed in the subsets based on the placement order used in their original set (i.e. the set presented in Table 6.2). The difference in the number of elements between any two consecutive subsets was 5. The tenth subset consisted of all the elements in the data set.

For each data subset used in the algorithm, the measurements taken corresponding to the subset were the value of the standard deviation obtained while running the algorithm on the laptop computer, the value of the standard deviation obtained while running the algorithm on the desktop computer, the value of the standard deviation computed using the Microsoft Excel spreadsheet, and the error in the standard deviation obtained from the laptop computation with respect to the corresponding standard deviation value obtained using the Microsoft Excel spreadsheet.

6.3.3 Justification for separating the ELC-MANET into phases

After separating the ELC-MANET into three distinct phases, a question arises naturally, and the question is, “why is it necessary to detect the phases of the ELC-MANET?” The answer to this question is what led to the development of TAPMRP. Hence, the simulation of this section had two aims. The first aim was to compare the performance of ODMRP in all the three ELC-MANET phases with each phase having its own characteristics that are identified in this thesis, while the second aim was to see the effect of the differences in the ELC-MANET topologies only on the performance of ODMRP.

The simulation test was conducted in two steps. In the first step, some of the settings of the simulation were kept constant in all the three ELC-MANET topologies considered, while other settings differed from one ELC-MANET topology to another depending on the characteristics of the ELC-MANET topology under consideration. Two of the important settings that were kept constant were the network size, which was fixed at 55 nodes in all the runs of the simulation; and the receiver sensitivity threshold settings which varied from -67dBm to -58dBm with a spacing of 0.5dBm between any two consecutive settings but the settings were the same in all the three ELC-MANET topologies. The highest setting of the receiver sensitivity threshold was chosen to be -58dBm because it was the maximum setting which produced a fully connected network in all the three ELC-MANET topologies. All the three simulated networks began to be completely connected such that any node had all the other nodes in a network as its one-hop neighbours at the lowest setting of about -65.5dBm. Based on this observation, the lowest endpoint of the settings of the receiver sensitivity threshold was decided to be -67dBm.

Settings that were applied to specific ELC-MANET topologies in the first step of the simulation were as follows:

- Geographical areas for the simulated networks were set at $20\text{m} \times 20\text{m}$ for the ELC-matrix topology, $80\text{m} \times 80\text{m}$ for the ELC-cluster-based topology, and $100\text{m} \times 100\text{m}$ for the ELC-random topology.
- The ELC-matrix topology was constructed using the $(6 \times 9) + 1$ arrangement of nodes.
- The ELC-cluster-based topology had four clusters out of which two were random, one circular and one matrix in arrangement; the clusters were almost of the same size.

Figure 6.5 shows the approximate dimensions of the clusters in the simulated network. The average distance between any two nodes within a cluster was 1.5m. In the ELC-random topology, 20 nodes were configured to move randomly at a constant speed of about 1.5m/s which was considered to be the average speed at which students can walk while using the ELC-MANET.

The simulation was conducted by sending CBR traffic from one node, which was arbitrarily chosen to be a source node, to the rest of the nodes (i.e. 54 nodes) in a simulated network. The CBR traffic was sent continuously for a maximum duration of 10 seconds (refer to

chapter 8, section 8.5) of the GloMoSim clock in each run of the experiment. The simulation had 10 runs with each run using a different seed value for the generation of various random numbers used by GloMoSim. The packet delivery ratio was the only performance metric used in the simulation.

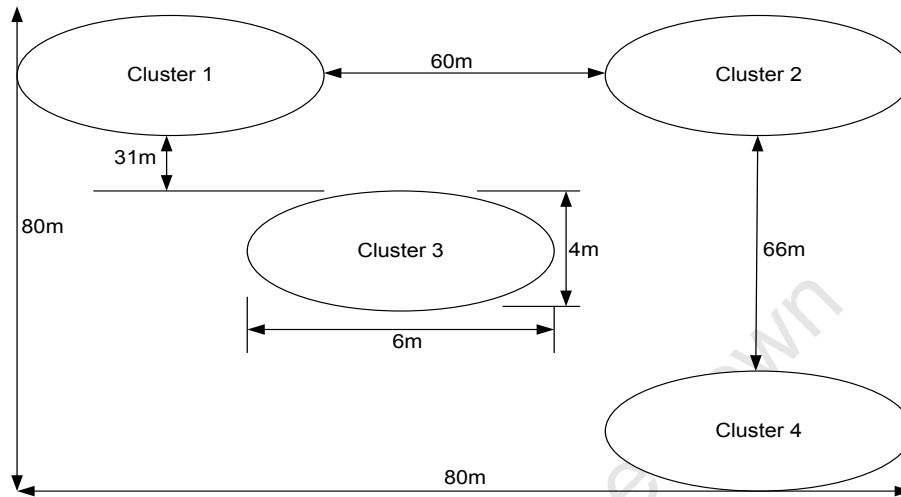


Figure 6.5 Approximate dimensions of clusters in the simulated network

The use of only the packet delivery ratio as a performance metric is considered to be sufficient based on two reasons. First, the packet delivery ratio also gives the level of packet loss which can be expressed as a percentage (i.e. $\text{packet loss percentage} = (1 - \text{packet delivery ratio}) \times 100$). Second, aspects of latency associated with the traffic are indirectly reflected in the packet delivery ratio through the level of packet loss. In this simulation as well as in the simulations of the rest of this research, the network traffic was generated using a Constant Bit Rate (CBR) generator, which does not adapt to prevailing network conditions; and therefore since each packet buffer in the intermediate nodes has a finite capacity, unreasonable latency in the CBR traffic is likely to result in packet loss. Note that the packet ratio is used as the only performance metric for data traffic in the rest of the simulations of this research based on the reasons given in section 6.1 and on the two reasons given here.

In the second step of the simulation, only the ELC-matrix and ELC-cluster-based topologies were considered. The ELC-random topology was not considered based on the understanding that being a random topology it has an infinite number of different network scenarios that can be created even for a small network size, and therefore the performance of a generic multicast

routing protocol in the ELC-random topology with respect to the characteristics of the topology only is likely to have no specific trend compared to the performance of the same protocol in the other two ELC-MANET topologies (i.e. ELC-matrix and ELC-cluster-based topologies).

All the simulation settings for both the ELC-matrix and ELC-cluster-based topologies used in the first step of the simulation were maintained in the second step except for the network area and the receiver sensitivity threshold settings. The geographical network area was fixed at $20\text{m} \times 20\text{m}$ for both topologies. The receiver sensitivity threshold settings were varied from -54dBm to -36dBm with each setting of the receiver sensitivity threshold being the same for both topologies.

The simulation was conducted by sending CBR traffic continuously from a source node for a maximum duration of 10 seconds in each run of the experiment just like in the first step of the simulation. The simulation had 10 runs with each run having a different seed value for use by the random number generator of GloMoSim. The second step of the simulation also used the packet delivery ratio as the only performance metric just like in the first step.

6.4 Matrix Topology Route Discovery Method

From this section onwards we will be looking at the simulation tests conducted in the third stage of the experiments of this research. The simulations were focussed on testing TAPMRP. The approach taken for the testing of TAPMRP is what may be referred to as a bottom-up approach. This means that the simulation tests start with the smaller components of TAPMRP and then move on to larger components, and finally deals with the components which affect the operations of the entire TAPMRP. Hence, simulation tests on the route discovery methods of TAPMRP (i.e. MTRDM, CTRDM, and RTRDM) are covered before the tests on TAPMRP as a whole.

The simulation test of MTRDM had three aims:

- To confirm if MTRDM uses fewer forward nodes than the blind flooding method of ODMRP.
- To compare the performance of MTRDM in terms of the packet delivery ratio with that of the blind flooding method of ODMRP, and

- To investigate the effect of mobility of nodes on the performance of MTRDM.

The three aims resulted in having a simulation test of two stages. The first stage of the simulation began with the configuration of the ELC-matrix topology of the nodes, which was constructed using a $(6 \times 9) + 1$ arrangement. The geographical area for the simulated was set at $20\text{m} \times 20\text{m}$. The receiver sensitivity threshold settings were varied from -54dBm to -36dBm . The upper endpoint of -36dBm was chosen because it was the maximum sensitivity value which yielded a completely connected network; any value higher than -36dBm produced a disconnected network. The connectivity of the network was increasing in the direction of decreasing receiver sensitivity threshold values such that at -52dBm each node in the network had any other node as its one-hop neighbour. Based on this connectivity observation, the lower endpoint was decided to be -54dBm .

One of the nodes was arbitrarily chosen as a source node, which was used to continuously send CBR traffic to the rest of the nodes in the network for a maximum duration of 10 seconds of the GloMoSim clock in each run of the experiment. Each setting of the receiver sensitivity threshold had ten runs of the simulation with each run having a different seed value for the generation of various random numbers used by GloMoSim.

One of the measurements taken was the number of forward nodes of the Join-Query packet in the simulated network with respect to the receiver sensitivity threshold settings. The other measurement was the packet delivery ratio which was also considered with respect to the receiver sensitivity threshold values.

After running the simulation of MTRDM, the testing process proceeded with the simulation of the blind flooding method of ODMRP for purposes of performance comparison. The blind flooding method was also run in the topology that was previously configured for MTRDM. Measurements taken during the experiment of the blind flooding method were the same as those taken when conducting the simulation of MTRDM, namely, the number of forward nodes and the packet delivery ratio.

The second stage of the simulation test was on the effect of mobility on the performance of MTRDM. The network setup used in this stage was the same as that used in the first stage. Mobility of nodes was configured using the random waypoint model which is available in GloMoSim. Three mobility speeds were experimented. The first speed to be tested was 0m/s

which represented a situation in which students are stationary. The second speed was 1.5m/s which was set as the maximum speed based on the understanding that mobility of nodes in a classroom environment is normally in the form of walking and therefore a speed of 1.5m/s, though possibly higher than the normal maximum walking speed of a person, is reasonable enough to represent the speed of moving students in the ELC environment. The last speed that was experimented was 10m/s which was arbitrarily chosen as a representative of any speed reasonably higher than 1.5m/s (refer to chapter 8, section 8.5, for implications of settings).

The simulation was conducted by sending CBR traffic continuously from a source node for a maximum duration of 10 seconds in each run of the simulation just like in the first step of the simulation. The packet delivery ratio was used as the only performance metric in the second stage of the simulation.

6.5 Cluster-based Topology Route Discovery Method

CTRDM was tested by simulations just like MTRDM. However, CTRDM had more tests than MTRDM. Simulation tests on CTRDM were conducted at two levels. The first level involved testing of the ideas proposed in this research for the active clustering component of CTRDM, while the second level dealt with the testing of CTRDM as a whole. Procedures for testing the ideas proposed for the active clustering component of CTRDM are covered in subsections 6.5.1, 6.5.2 and 6.5.3 followed by the methodology for testing CTRDM as a whole in subsection 6.5.4.

This research proposes three ideas for the active clustering of CTRDM which were tested by simulations, and these are as follows:

- The introduction of secondary cluster-heads aimed at providing load balancing.
- The use of the node redistribution parameter to provide more load balancing.
- The use of neighbour information obtained during a controller hello interval to predict mobility behaviour of a node in the next controller hello interval.

6.5.1 Introduction of secondary cluster-heads

The aim of the simulation test on the introduction of the secondary cluster-heads was to verify the effectiveness of using secondary cluster-heads. The ELC-cluster-based network

setup was used for the simulation. The simulated network was the same as the one used in subsection 6.3.3 whose approximate dimensions are shown in Figure 6.5. The size of the simulated network was fixed at 55 nodes and its geographical area was $80\text{m} \times 80\text{m}$. The transmission range of the nodes, which was the same, was set indirectly using the receiver sensitivity threshold. The settings of the receiver sensitivity threshold were varied from -67dBm to -59.5dBm .

The simulation was conducted in two stages. In the first stage, the simulation used a one-hop clustering algorithm (covered in chapter 5) that used the degree of connectivity for cluster-head selection and had no limit on the maximum size of a cluster. The algorithm allowed only one cluster-head (a primary cluster-head) per cluster (a logical one). The interval of the hello messages was set at 30 seconds of the GloMoSim clock while the refresh interval of the clusters was set at 120 seconds of the same clock.

Three nodes belonging to the same cluster were arbitrarily chosen as source nodes. Each of the source nodes continuously sent CBR traffic to the rest of the nodes in the network for a maximum duration of 10 seconds of the GloMoSim clock in each run of the experiment. The source nodes sent the traffic in a consecutive manner and not simultaneously. Each setting of the receiver sensitivity threshold had ten runs of the simulation with each run having a different seed value for the generation of various random numbers used by GloMoSim. Performance metrics used were the number of forward nodes, the workload of each forward node in terms of the number of retransmissions of the Join-Query packet performed, and the packet delivery ratio.

In the second stage of the simulation, the clustering algorithm was modified so as to perform two important tasks at this stage. The first task was to create clusters (logical ones) by setting their maximum size at 15 nodes, while the second task was the introduction of secondary cluster-heads. The rest of the simulation was conducted just like in the first stage.

6.5.2 Node redistribution parameter

The simulation test on the node redistribution parameter were aimed at confirming the effectiveness of using the node redistribution parameter (refer to chapter 4). The simulation used a network setup which was the same as the one used for testing the effect of the introduction of secondary cluster-heads. Like in the simulation on secondary cluster-heads,

settings of the transmission range were done indirectly using the receiver sensitivity threshold settings which were varied from -67dBm to -59.5dBm.

The simulation was performed in two steps. In the first step, the simulation used a one-hop clustering algorithm which used the degree of connectivity for cluster-head selection. The interval of the hello messages was set at 30 seconds of the GloMoSim clock while the interval of refreshing clusters was set at 120 seconds of the same clock. Three nodes were chosen from different clusters to be multicast sources. The three source nodes consecutively, and not simultaneously, sent CBR traffic continuously for 10 seconds over the network in each run of the simulation. Each setting of the receiver sensitivity threshold had ten runs of the simulation, and each one of them used a different seed value for the generation of various random numbers used by GloMoSim. Measurements taken during the simulation were the number of forward nodes, the workload of each forward node in terms of the number of retransmissions of the Join-Query packets performed, and the packet delivery ratio.

The clustering algorithm was modified in the second step of the simulation in order to select cluster-heads using the node redistribution parameter as the criterion. The rest of the simulation was conducted using the procedure outlined in the first step.

6.5.3 Effect of the node mobility formula

The aim of the validation test on the prediction of node mobility was to verify the effectiveness of Equation (4.4) derived in chapter 4. The network setup used in the simulation of this subsection was the same as the one used in subsections 6.5.1 and 6.5.2 except for the inclusion of mobility behaviour of some nodes in the network. Instead of using the random waypoint mobility model (the default model in GloMoSim) which could have completely distorted the cluster-based arrangement of nodes, the mobility behaviour used in this simulation was created by arbitrarily selecting three nodes from each cluster which were configured to move between clusters at an average speed of 1.5m/s. Thus, in total, there were 12 mobile nodes. The simulation made use of the receiver sensitivity threshold settings as an indirect way of having settings of the transmission range, and these were varied from -67dBm to -59.5dBm.

The simulation had two stages. The first stage involved the use of a clustering algorithm which selected cluster-heads based on the degree of connectivity as the criterion. The simulated network made use of CBR traffic which was continuously generated and

transmitted for a maximum duration of 10 seconds of the GloMoSim clock by one node which was chosen for that purpose. The simulation had ten runs and each one of them had a different seed value for the generation of random numbers used by GloMoSim. The packet delivery ratio was used as the only performance metric.

In the second stage of the simulation, the clustering algorithm was modified so as to create clusters using the numerical values computed by the mobility formula (i.e. Equation (4.4)) as the criterion for selecting nodes for the role of cluster-head. The packet delivery ratio was also the only measurement taken in this stage. Both stages made use of mobile nodes.

6.5.4 Testing of CTRDM as a whole

Simulation of CTRDM had three aims: 1) To verify if CTRDM requires fewer forward nodes for propagation of control packets than the blind flooding method of ODMRP, 2) To compare the performance of CTRDM with that of the blind flooding method in terms of the packet delivery ratio, and 3) To investigate the effect of mobility associated with the ELC-cluster-based network scenarios on the number of forward nodes required for complete broadcast operations of control packets and on the packet delivery ratio of CTRDM.

The network configuration used for the simulation was the same as the one used in subsections 6.5.1 through 6.5.3. Like in the other subsections, the network size was fixed at 55 nodes and the settings of the transmission range of the nodes were made indirectly using the receiver sensitivity threshold settings which were varied from -67dBm to -59.5dBm.

There were two steps into which the simulation was divided. The first step started by simulating CTRDM using of a clustering algorithm which selected nodes for the role of cluster-head using a criterion which was based on the computation of a weight. The weight for each node was determined as follows:

$$weight = (0.6 \times node\ redistribution\ parameter) + (0.4 \times stability\ factor) \quad (6.1)$$

Note that the term “stability factor” that appears in Equation (6.1) is given by Equation (4.5) in chapter 4 and comes from the Equation (4.4) which is used to predict node mobility. The computation of the weight in the simulation did not include the factor for the residual battery energy proposed in this thesis, and hence the weightings of the factors proposed in the thesis were not used (refer to Equation (4.6)). The domain for the values of the weight in the

simulation was $[0, 100]$ instead of $[0, 1]$ used in the thesis due to the need to achieve computational efficiency.

The simulation in the first step was conducted by using one of nodes in the simulated network as a source node which sent CBR traffic continuously to the rest the network for 10 seconds of the GloMoSim clock. For each receiver sensitivity threshold setting, there were ten runs of the same experiment with each run having a different seed value for use by the GloMoSim random number generator. Measurements taken during each run of the simulation were the number of forward nodes of the Join-Query packets and the packet delivery ratio.

The first step of the simulation proceeded by repeating the whole simulation using the blind flooding of ODMRP as the route discovery method instead of CTRDM. The blind flooding method did not make use of the clustering of nodes. Performance metrics used were the number of forwards and the packet delivery ratio just like in the case of CTRDM.

In the second step, some nodes of the simulated network were made mobile by using a user-defined configuration described in subsection 6.5.3. CTRDM was simulated just like in the first step. This second step also made use of the blind flooding method for performance comparison and it was simulated following the procedure of the first step.

6.6 Random Topology Route Discovery Method

A simulation test on RTRDM was based on a partial implementation of the passive clustering method proposed by Y. Yi et al. [100], and had two aims. The first aim was verify if the passive clustering method is capable of using fewer forward nodes than the blind flooding method of ODMRP, and the second aim was to investigate the performance of the passive clustering method in terms of the packet delivery ratio. The implementation of the clustering method did not use the lengthy periodic calculations of cluster member states for adapting to mobility proposed in [100], but instead it used the resetting of any state back to the initial state after performing every forwarding action.

The simulation made use of a random topology configuration which was set using an in-built function provided by GloMoSim-2.03. The network size was fixed at 55 nodes and the network area was set at $100\text{m} \times 100\text{m}$. Like in the other simulations of this research, the transmission of the nodes was the same for all the nodes and was set indirectly using the

receiver sensitivity threshold. Settings of the receiver sensitivity threshold were varied from -67dBm to -59.5dBm. Mobility of the nodes was configured using the random waypoint model provided by GloMoSim. The average speed of the nodes was set at 1.5m/s. The speed of 1.5m/s was considered as the realistic average speed of MANET users expected in the ELC-random topology environment.

There were two steps in the simulation. The first step involved the simulation of RTRDM which made use of a passive clustering method. The simulated network used one source node which transmitted CBR traffic continuously to the rest of the nodes in the network for a maximum duration of 10 seconds of the GloMoSim clock. The CBR traffic generator's inter-departure time between packets was set at 10 milliseconds. The relay of Join-Query packets (i.e. control packets) at every forward node in the network made use of a delay jitter period which was set at 60 milliseconds of the GloMoSim clock. The delay jitter of 60ms was chosen because it was the minimum jitter which yielded a significant decrease in the number of forward nodes compared to the number of forward nodes used by the blind flooding method. Note that Join-Query packets were broadcast by the source node after every 3 seconds. Each receiver sensitivity threshold setting had ten runs of the simulation with each run having a different seed value for use by the random number generator of GloMoSim. Measurements taken during the simulation were the number of forward nodes and the packet delivery ratio.

The second step of the simulation involved the blind flooding method of ODMRP. The network setup used by the blind flooding method was the same as the one used by RTRDM. Basically, the blind flooding method followed the same simulation procedure used by RTRDM. However, the blind flooding method did not make use of clustering of nodes and a delay jitter for propagation of Join-Query packets as the case was with RTRDM.

6.7 Simulation Tests on TAPMRP

Validation tests were not only conducted on the route discovery methods of TAPMRP but also on operations of TAPMRP which affect TAPMRP as a whole. The tests were focussed on the ideas proposed in chapter 5 on the operations of TAPMRP. One of such ideas is the convergence of TAPMRP from the time when a controller node disappears (or fails) from a multicast group to the time when the multicast group recognises only one new controller node. The second idea to be tested was the effort by TAPMRP to provide a seamless switch

from one route discovery method to another. A simulation test was also conducted on the probability of making correct switching decisions from one route discovery method to another. Finally, the advantage of using of TAPMRP over ODMRP was analysed theoretically using information gathered from all the simulations conducted in this research.

6.7.1 Convergence of TAPMRP after a controller node failure

The meaning of the term “convergence” in this context is the termination of a process of electing a new controller node in TAPMRP such that at the time of the termination there is only one new controller node elected for the affected multicast group. The simulation test on the convergence of TAPMRP after a controller node failure had two aims. The first aim was to find out how rapidly the process of electing a new controller node converges from the time when a current controller node disappears to the time when only one new controller node is available in the affected multicast group. Another aim was to see how the performance of a network is affected during the time of electing a new controller node. Note that the test does not include the quality of the selection since the “First Declare First Wins” rule is adopted for selecting a controller node (section 5.4, 2nd paragraph of page 100).

The simulation had two stages. The speed of convergence of the new controller election process was the focus of the first stage. The first stage was further divided into two parts. The first part of the first stage used a network setup of 55 nodes which were confined to a simulation network area of $100\text{m} \times 100\text{m}$. The nodes were placed using a random model provided by GloMoSim. The transmission range of the nodes was set indirectly using the receiver sensitivity threshold. Settings of the receiver sensitivity threshold were varied from -67dBm to -59.5dBm.

The first part of the first stage of the simulation was conducted by first running TAPMRP for 500 seconds of the GloMoSim clock and then its controller node was deliberately failed such that for the rest of the simulation which in total took 3000 seconds (or 50 minutes) there were no more controller hello messages from the initial controller node. TAPMRP was programmed to trigger announcements of new controller nodes after 250 seconds, which is approximately twice the controller hello interval of 120 seconds, from the time of detecting the absence of the current controller node.

The taking of measurements of the simulation took advantage of the observation that new controller announcements from the nodes were triggered almost at the same time such that

collectively the announcements had a periodic interval of approximately 120 seconds which is the same as a controller hello interval of a single controller node. Hence, the speed of convergence of the new controller election process was measured in terms of the number of controller hello intervals in which the number of new controller node announcements was more than one. Thus, simulation instances with fewer controller hello intervals with multiple announcements were considered to have converged more rapidly than instances with more controller hello intervals with multiple announcements. Note that using absolute time measurements such as seconds or minutes for the election convergence was not going to be informative enough since the GloMoSim clock differs from the real time clock.

The second part of the first stage of simulation differed from the first part by considering a range of network sizes instead of just one network size as the case was with the first part. The network sizes considered were 10, 15, 20 up to 80 with a spacing of 5 nodes between any two adjacent sizes. This second part of the first stage was aimed at investigating the relationship between the speed of election convergence and the network size. Simulation procedure and measurements used in the second part were basically the same as those used in the first part.

The second stage of the simulation concentrated on the performance of TAPMRP during the period of electing a new controller node. The network setup used had 55 nodes with a random arrangement just like in the first part of the first stage. One of the nodes was arbitrarily chosen as a source node and it was used to send CBR traffic to the rest of the network continuously beginning from 700 seconds from the start of the simulation up to 1700 seconds. The CBR traffic duration covered almost the entire duration of controller election process in all the instances of the simulation that were run. The packet delivery ratio was the only performance metric used in this simulation. For purposes of performance comparison, the entire simulation of the second stage was repeated in a TAPMRP scenario of having only one controller node.

6.7.2 Seamless route discovery method switch

The aim of the simulation test on the design effort of TAPMRP to provide a seamless switch involving route discovery methods was to compare the performance of TAPMRP during the switch-over period and after switching when there is only one route discovery method in use. The test was conducted on the six important transitions that can arise from the three route discovery methods of TAPMRP. The six important transitions are as follows: from MTRDM

to CTRDM and vice versa, from MTRDM to RTRDM and vice versa, and from CTRDM to RTRDM and vice versa. The remaining three transitions, namely from MTRDM to MTRDM, from CTRDM to CTRDM, and from RTRDM to RTRDM were not considered based on the understanding that their nature of transition only involves the same route discovery method and therefore it cannot negatively affect the performance of TAPMRP.

The simulation test involved all the three ELC topologies. The network size in all the three ELC topologies was fixed at 55 nodes, and the network area was also fixed at $100\text{m} \times 100\text{m}$. The receiver sensitivity threshold setting for all the three ELC topologies considered was also fixed at -62.0dBm . It was not possible to directly configure a network scenario in GloMoSim-2.03 that could dynamically transform into another topology during a single run of a simulation. Hence, the simulation proceeded by configuring each ELC topology in a separate simulation and initialising TAPMRP with a route discovery method that was not corresponding to the ELC topology under consideration. During the run of the simulation, TAPMRP was able to switch its route discovery method from the initialised method to the method that was corresponding to the ELC topology under consideration. As a result, the aims of the simulation test were fulfilled.

The simulation of TAPMRP in each ELC topology was conducted by allowing one source node to send CBR traffic to the rest of the nodes in the network continuously for 120 seconds. Out of the 120 second period, 60 seconds covered the period just before switching to another route discovery method while the remaining 60 seconds covered the period just after switching to a new route discovery method. The packet delivery ratio during the 120 seconds of changing from one route discovery method to another was determined as the performance metric of the simulation. Thereafter, the simulation was repeated and the source node sent CBR traffic for again 120 seconds but this time the whole 120 seconds of CBR traffic covered a period during the use of a new route discovery method. The packet delivery ratio was also determined in this type of repeated simulation run. Note that during this repeat of the simulation the seed value used by the random number generator of GloMoSim was kept the same.

6.7.3 Probability of making correct switch decisions

Simulations of this subsection were aimed at determining the probability of making a correct switch decision when the ELC-MANET changes its topology from one form to another. All

the three ELC-MANET topologies, namely the ELC-matrix topology, the ELC-cluster-based topology and the ELC-random topology were considered in the simulations. Like in the simulation test of subsection 6.7.2, it was not possible to configure in GloMoSim-2.03 a direct and clear dynamic transformation of a topology from one form to another during a single run of a simulation. Therefore, each ELC topology had its own simulation in which TAPMRP was initialised with a route discovery method that was different from the one that was corresponding to the topology under consideration. During a simulation, depending on the computed value of the standard deviation of the number of neighbours of nodes, TAPMRP was able to switch to the route discovery method that was corresponding to the topology under consideration. Hence, it was possible to fulfil the aim of the simulations.

The network size in all the three ELC network topologies was fixed at 55 nodes. Each ELC network had its own network area dimensions. The network area of the ELC-matrix network was $20\text{m} \times 20\text{m}$, the area of the ELC-cluster-based network was $80\text{m} \times 80\text{m}$, and that of the ELC-random network was $100\text{m} \times 100\text{m}$. The transmission range of the nodes was set indirectly using the receiver sensitivity threshold. The receiver sensitivity threshold settings of the ELC-matrix topology varied from -54dBm to -40dBm with a spacing of 1dBm , while the settings for the ELC-cluster-based topology and the ELC-random topology were the same and they ranged from -67dBm to -59dBm with a spacing of 0.5dBm .

The simulations considered only three route discovery method switches, namely CTRDM to MTRDM, MTRDM to CTRDM and MTRDM to RTRDM. The reasoning behind the selection of the three route discovery method switches was that the results from these three switches can be generalised to all the other switches because what determines a switch to a particular route discovery method are the connectivity characteristics of the prevailing topology and not the connectivity characteristics of the topology before the current one or the route discovery method used before making a switch.

Each of the receiver sensitivity threshold setting used in the simulations had ten runs of the same simulation. Each simulation run had a different seed value for use by the random number generator of GloMoSim. The probability of making a correct switch decision was determined as a fraction of the number of simulation runs in which correct switch decisions were made over the total number of the simulation runs which was ten.

6.7.4 Benefit analysis of using TAPMRP

Even though TAPMRP is designed as an improvement on ODMRP, it differs significantly from ODMRP. TAPMRP is a hybrid protocol which has a proactive component as well as a reactive component while ODMRP is a pure on-demand (or reactive) multicast routing protocol which operates only when there are data packets to be transmitted. ODMRP piggybacks the Join-Query in multicast data packets which implies that ODMRP does not use explicit control messages for the route discovery operation. Operation of ODMRP does not rely on the use of periodic hello messages as the case is with other multicast routing protocols of MANETs. This means that ODMRP was designed with the intention of minimising control overhead as much as possible.

While ODMRP has a design which minimises control overhead, TAPMRP makes use of a number of explicit control messages. For instance, the controller node periodically broadcasts a hello message to other nodes in the network. The clustering component of TAPMRP also makes use of explicit control messages. Based on these observations, a detailed analysis of the benefit of using TAPMRP over ODMRP was carried out. The analysis made use of the settings of the simulations that were conducted in this research. Refer to the results chapter for details of the analysis.

6.8 Summary

The aim of this chapter is to present the procedures used to validate by computer simulations the existence of the research problem and the ideas generated in this research which constitute a proposed solution to the research problem. All the simulation tests were conducted in GloMoSim-2.03 [5], a network simulator used by researchers for wireless local area networks. One of the features of GloMoSim-2.03 which was very useful in this research was the possibility of creating user-defined topologies of nodes. The availability of the implementation of ODMRP in GloMoSim-2.03 [5] was also very useful in the simulation tests since TAPMRP and its three route discovery methods were tested by incorporating them in ODMRP.

A validation test on the existence and the significance of the research problem was conducted on ODMRP. The simulation test mainly involved the investigation of the broadcast

redundancy generated by ODMRP under different networking scenarios. CBR traffic was used for the simulation.

The proposed mechanism for detecting the ELC-topologies based on the standard deviation of the number of neighbours of nodes in a network was formulated using a computer simulation. The simulation involved construction of the models of the ELC-matrix topology, ELC-cluster-based topology and ELC-random topology. The simulation proceeded in each of the three model ELC-topologies by investigating the dependence of the standard deviation of the number of neighbours of nodes in a network on the network size and the transmission range. The simulation also included the validation of the proposed online algorithm used to compute the standard deviation of the proposed mechanism for detecting the ELC-topologies. The last simulation test on detecting the ELC-topologies was on the justification for separating the ELC-MANET into the three ELC-topologies.

Each of the three route discovery methods of TAPMRP, namely MTRDM, CTRDM and RTRDM were tested by simulations. The simulations tests were generally aimed at confirming if each of the route discovery methods is indeed the best method in terms of performance for its intended ELC-topology. The presence of the active clustering technique in CTRDM had its own set of simulation tests. The first test on CTRDM's clustering method was on the effectiveness of the introduction of secondary cluster-heads. The introduction of the node redistribution parameter in the clustering method of CTRDM was also tested by simulation. The third and also the last simulation test of CTRDM on clustering was on the effectiveness of using Equation (4.4) for predicting node mobility.

The last simulation tests of this research were on the operations of TAPMRP which affect TAPMRP as a whole. The first set of such tests was on the convergence (i.e. process termination) of the process of selecting a new controller node in TAPMRP, and the main aim was to determine how fast the process converges (or terminates). The second simulation test on TAPMRP's operations was on TAPMRP's effort to provide a seamless route discovery method switch. The aim of this test was to find out the performance of TAPMRP during the time of switching from one route discovery method to another and after making a switch to a new route discovery method. Determining the probability of making the correct switch decision that corresponds to the right ELC-topology change was also part the simulation tests on TAPMRP's operations. The chapter ends with an introduction to a benefit analysis of using TAPMRP over ODMRP.

CHAPTER 7

RESULTS AND ANALYSIS

7.1 Introduction

This chapter presents the results obtained from the GloMoSim simulations which were conducted with the intention of testing the ideas generated during this research undertaking. The analysis of each set of the results obtained is also covered together with the explanations for any behaviour shown by the results.

Each section (or subsection) of this chapter is based on a simulation design outlined in a corresponding section (or subsection) in the methodology chapter. For example, section 7.3 on the detection of the ELC-MANET phases corresponds to section 6.3 in the methodology chapter which is also on the same topic. Similarly, subsection 7.3.1 on the formulation of the standard-deviation mechanism is based on a simulation design presented in subsection 6.3.1 in the methodology chapter. Thus, the decimal numbering of corresponding headings in the two chapters (i.e. chapter 6 and 7) only differ in the first digit which represents a chapter.

As mentioned in the introduction section of chapter 6 (i.e. section 6.1), the number of forward nodes of a control packet which is broadcast in a network is the main performance metric of the simulations of this research, and therefore most of the results presented in this chapter are analysed with respect to the number of forward nodes. Performance associated with data traffic is analysed with respect to the packet delivery ratio which is considered to be sufficient based on the reasons given in section 6.1 and subsection 6.3.3.

The organisation of this chapter is as follows. Section 7.2 covers the results on the validation of the research problem, and these are followed by the simulation results on the proposed method for detecting the phases of the ELC-MANET in section 7.3. The sections which follow section 7.3 present simulation results on the operations of the various components of TAPMRP. Section 7.8 presents a discussion on the overall results. A summary in section 7.9 concludes this chapter.

7.2 Validation of the Research Problem

The data obtained during the simulation aimed at the validation of the existence and the significance of the research problem were plotted, and the resulting graphs are shown in Figure 7.1 and 7.2. The horizontal axis in Figure 7.1 is expressed in terms of the receiver sensitivity threshold which gives a measure of the transmission range of nodes. The longest transmission range used in the simulation is represented by -70dBm while -56dBm represents the shortest transmission range of the simulation. Hence, in moving from -70dBm towards -56dBm the transmission range of the nodes in the simulated network decreases.

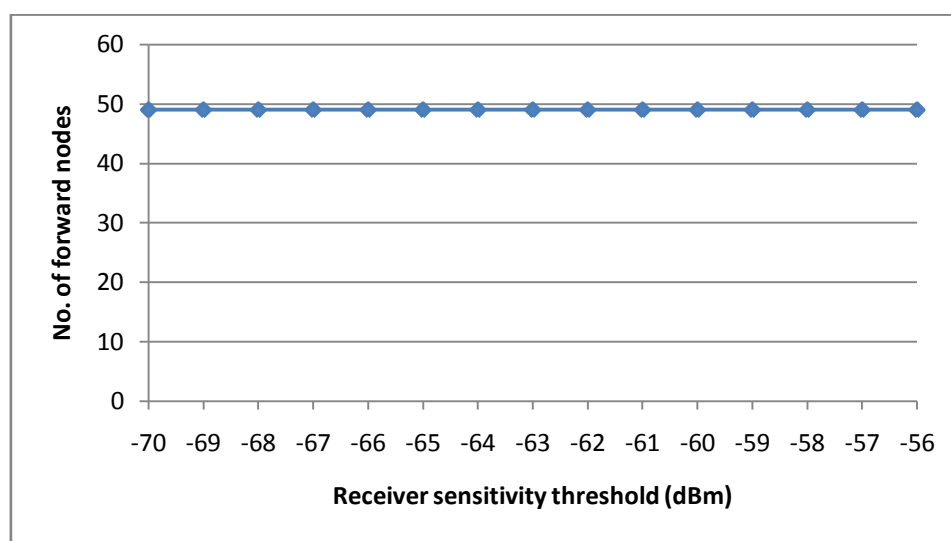


Figure 7.1 Forward nodes in ODMRP with respect to the transmission range

Figure 7.1 clearly shows that the number of forward nodes in a broadcast operation of Join-Query packets in ODMRP is independent of the transmission range of the nodes. This implies that the blind flooding method which ODMRP uses for broadcasting the Join-Query packet does not take into consideration the connectivity of the network for possible reduction of broadcast redundancy.

Another characteristic of blind flooding shown in Figure 7.1 is that all the nodes in a network except for the source node participate in forwarding a broadcast packet. In the simulated network of this simulation there were 50 nodes in total, and out of the 50 nodes, 49 nodes acted as forward nodes; this implies that only the source node did not take part in forwarding the Join-Query. The worst case scenario of blind flooding in this simulation was observed from the receiver sensitivity threshold of -70dBm to -68dBm in which a source node had the

rest of the nodes in the network (i.e. 49 nodes) as its one-hop neighbours. Clearly, in such a situation there was no need for the neighbours of the source node to rebroadcast the Join-Query since all the other nodes in the network had already received a copy of the Join-Query direct from the source node itself.

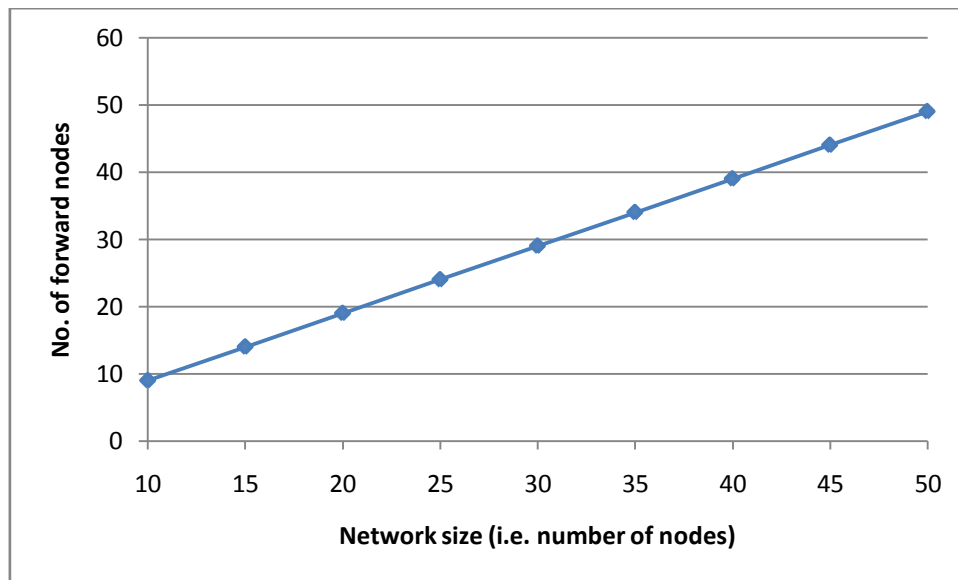


Figure 7.2 Dependence of forward nodes on the network size in ODMRP

Figure 7.2 illustrates the relationship between the number of forward nodes resulting from the blind flooding method of broadcasting and the size of a network. The graph in Figure 7.2 indicates the number of forward nodes of the Join-Query packet in a network of size n is given by $n - 1$.

The one node which does not take part in forwarding a broadcast packet is the source node of the broadcast packet being propagated in the network. Based on the results shown in Figure 7.2, it can be concluded that when using the blind flooding method for broadcasting a node in a network of any size is not spared from the possibility of performing redundant retransmissions unless it is a source node of a broadcast packet being propagated. In fact, all the forward nodes shown in Figure 7.2 performed a redundant retransmission of the Join-Query packet because the source node for each network size n shown in the figure had $n - 1$ as its 1-hop neighbours.

Figure 7.1 and 7.2 provide evidence that indeed blind flooding leads to broadcast redundancy and that no node in a network of any size is spared from wasting its energy on redundant

retransmissions unless it is a source of a broadcast packet being propagated in the network. Broadcast redundancy is therefore considered to be a significant problem in this research based on the following reasons, which are related to the observations in Figure 7.1 and 7.2:

- Each rebroadcast of a packet uses some energy.
- The number of redundant retransmissions generated by blind flooding, which is used by most of the existing multicast routing protocols including ODMRP for broadcasting, increases as the size of the network increases resulting in more wastage of energy by the network as a whole.
- Join-Query packets, which are broadcast packets, are generated and transmitted periodically during a multicast session, and therefore the level of broadcast redundancy in a network increases as the duration of a multicast session and the number of source nodes generating Join-Query packets increase in the network; this can easily lead to the broadcast storm problem observed in [8].
- Devices of the ELC-MANET are powered by battery energy which is limited in supply, and therefore making efficient use of the energy is of paramount importance.

7.3 Detection of the ELC-MANET Phases

The first set of simulation results obtained under this section was on the validation of the proposed mechanism that uses the standard deviation of neighbours of nodes to detect the ELC-MANET topologies (refer to subsection 6.3.1 of chapter 6). This first set of results and its analysis are presented in subsection 7.3.1. The next results presented in this section are on the proposed online algorithm for computing the standard deviation values (refer to subsection 6.3.2 of chapter 6 for details), and these are covered in subsection 7.3.2. The last subsection of this section presents results and their analysis on the justification for separating the ELC-MANET phases proposed in this research (for details, refer to subsection 6.3.3 of chapter 6).

7.3.1 Formulation of the standard-deviation mechanism

Results on the formulation of the proposed standard-deviation mechanism are in two categories. The first category of the results is on the variation of the standard deviation (*stdev*) of neighbours of nodes with the transmission range, while the second category is on the dependence of *stdev* on the network size.

Dependence of *stdev* on the transmission range

The data obtained from the simulation on the dependence of *stdev* on the transmission range were plotted as shown in Figure 7.3. The graph indicates that as the transmission range decreases in going from -52dBm towards -38dBm, *stdev* values for both the matrix and the random topology increase rapidly until maximum values are reached at -45dBm for the matrix topology and -44dBm for the random topology, and thereafter the values start to decrease steadily. At the receiver sensitivity threshold setting of -52dBm and below, every node recognised the presence of almost all the other nodes in the simulated network (in both the matrix and random topology). This high network connectivity at -52dBm and below resulted in *stdev* values of zero. Above -38dBm, the sensitivity of the receiver was too low to detect a signal even from the nearest neighbour, therefore, no network was created.

The observation on the dependence of *stdev* on the transmission range necessitated that each measurement of *stdev*'s variation with the network size (results of the next subsection of this section) be based on varying the transmission range (covered in detail in subsection 6.3.1 of chapter 6). Consequently, only peak values of *stdev* were recorded.

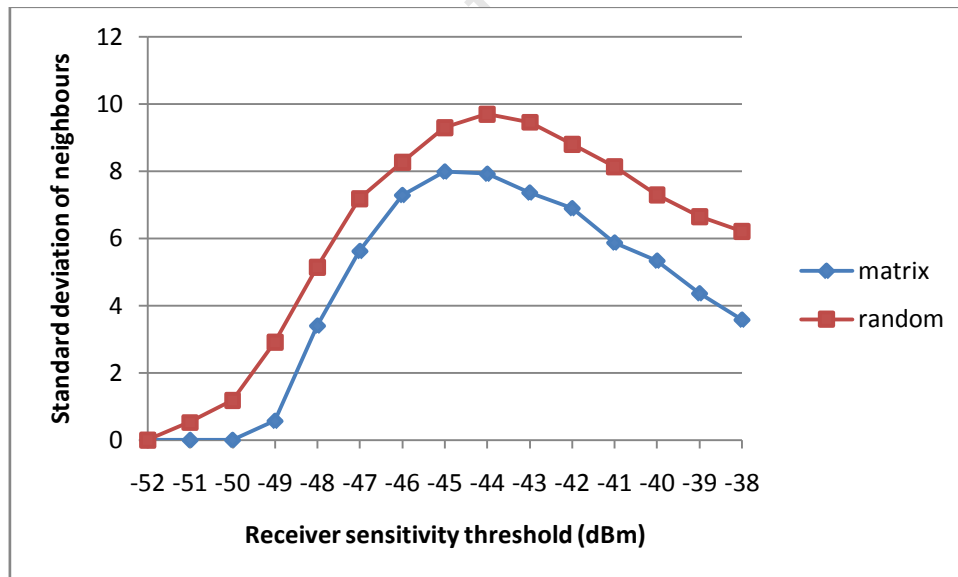


Figure 7.3 Variation of *stdev* with the transmission range

Variation of *stdev* with the network size

Results from the investigation into the variation of *stdev* with the network size were in two sets. The first set of the results was on the matrix topologies while the second set was on the

cluster-based topologies. The values of *stdev* under the matrix topologies came from the $(M \times N) + 1$ matrix which gave the highest value for a given network size. The inclusion of a random cluster in the model of ELC-cluster-based topologies was responsible for ensuring that the values of *stdev* obtained were the highest possible values from ELC-cluster-based topologies (refer to subsection 6.3.1 of chapter 6).

Two polynomial fit curves through *stdev* results obtained from the network sizes of 10 up to 55 were generated as shown in Figure 7.4 using Matlab software. The curve fitting software used the linear least-squares method for the fits. The following linear equations were obtained:

$$y_{\text{matrix}} = 0.1416x + 0.2259 \quad (7.1)$$

$$y_{\text{cluster-based}} = 0.1713x + 0.0946 \quad (7.2)$$

The two equations, namely Equation (7.1) and (7.2), were obtained using 95% confidence bounds. Equation (7.1) is for ELC-matrix topologies and is represented in Figure 7.4 by a solid line. A dashed line on the graph of Figure 7.4 represents Equation (7.2) and belongs to ELC-cluster-based topologies.

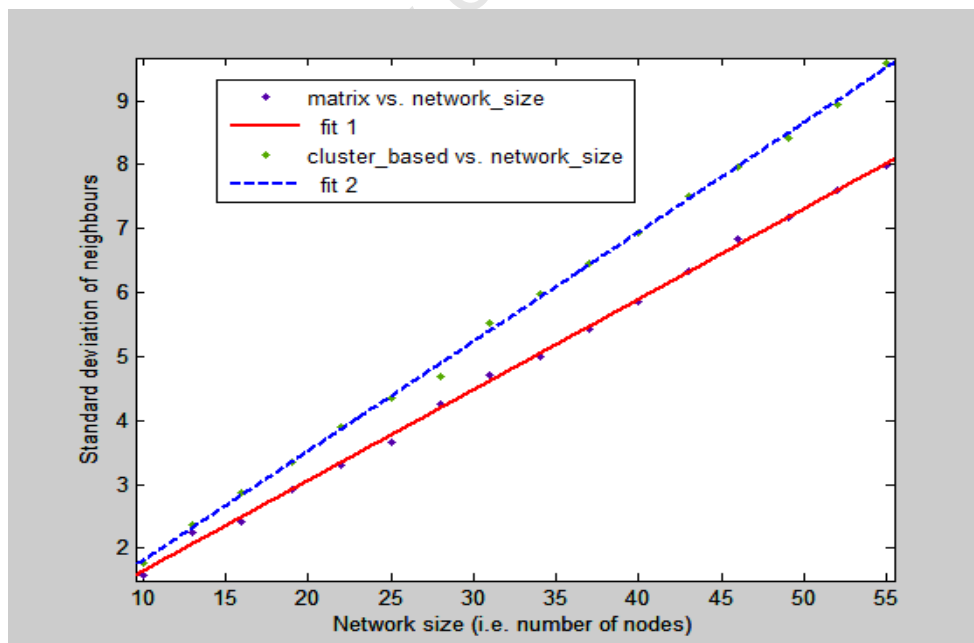


Figure 7.4 Demarcation lines for ELC-MANET topologies

The two graph lines of Figure 7.4 show that ELC-cluster-based topologies have higher values of *stdev* than ELC-matrix topologies for a given network size. Figure 7.4 also shows that the values of *stdev* increase as the network size increases, which is expected since as the network size increases nodes tend to have more neighbours.

Interpretation of the results: The interpretation of the graph of Figure 7.4 is that the solid line representing Equation (7.1) provides a demarcation line between ELC-matrix topologies and ELC-cluster-based topologies. Almost all the ELC-matrix topologies lie below the line y_{matrix} , while the region between the solid graph line and the dashed line is where most of the cluster-based topologies are located. Most of the ELC-random topologies exist above the line $y_{\text{cluster-based}}$.

Considering that network connectivity in the ELC-MANET environment uses radio signals, it implies that the three regions of ELC topologies obtained in Figure 7.4 are probabilistic in nature. Therefore, there are still overlaps of mainly two adjacent regions along the two graph lines. In the case of the ELC-matrix region, since the values of *stdev* used for ELC-matrix topologies were the highest possible values, it means that the probability of the ELC-matrix region extending into the ELC-cluster-based region along the line y_{matrix} is very small. Hence, the existence of the ELC-matrix topology can be detected with a very high probability. In fact, out of the three ELC topologies, the matrix topology is likely to have the highest probability of being identified.

The inclusion of a random cluster in the ELC-cluster-based topologies ensures that the probability of the ELC-cluster-based region extending into the ELC-random region is small along the line $y_{\text{cluster-based}}$. However, theoretically, the probability of the ELC-cluster-based region extending beyond the y_{matrix} into the matrix region is still high. Therefore, the probability of detecting the ELC-cluster-based topology is not as high as that of the ELC-matrix topology. For a given network size, the range of the *stdev* values assigned to the ELC-random topology is much longer (i.e. from a particular value up to infinity) than that assigned to the ELC-cluster-based topology; therefore, the ELC-random topology is likely to have a higher probability of being detected than the ELC-cluster-based topology.

The main argument on the usefulness of the proposed mechanism is that in the first place, the mechanism relies on prior knowledge that the ELC-MANET network has only three distinct topologies. Hence, the probability of detecting any of those three topologies without the use

of any extra information is $1/3$, which is well above zero. The use of the standard deviation values given by Equation (7.1) and (7.2) increases the probability of identifying each of the three ELC-topologies to a value above $1/3$. In the case of the ELC-matrix topology, the probability of detecting it is much higher than $1/3$.

Note that equating Equation (7.1) to (7.2) and then solve for x yields a value of x which is approximately equal to 4.42. This implies that for any network size below 4.42, Equation (7.2) yields lower *stdev* values than those produced by Equation (7.1). Since such results have no practical use, the use of Equation (7.2) for providing *stdev* values for the ELC-cluster-based topologies is only valid for network sizes of 5 nodes and above. Theoretically, the validity of Equation (7.2) is for network sizes above 4.42 nodes, but since we cannot have a fractional node it makes sense to shift upwards the validity point of Equation (7.2) to a network size of 5 nodes. For network sizes below 5 nodes only Equation (7.1) is valid, and in this situation Equation (7.1) provides a demarcation line between ELC-matrix topologies and ELC-random topologies while ELC-cluster-based topologies are assumed to be non-existent.

7.3.2 Online algorithm for computing standard deviation

The data obtained from the validation test on the proposed online algorithm for computing standard deviation (refer to subsection 6.3.2 of chapter 6) are presented in Table 7.1. Column 1 of Table 7.1 presents only the data-set sizes which were used for testing the algorithm, while the data (or elements) contained in the data sets are presented in subsection 6.3.2 of chapter 6. Presented in column 2 are the standard deviation values obtained by using the Microsoft Excel spreadsheet. Column 3 contains the standard deviation values obtained by using the proposed online algorithm while running on a laptop computer. Standard deviation values obtained by running the online algorithm on a desktop computer are presented in column 4. Column 5 shows the error in the standard deviation values obtained by the online algorithm with respect to the standard deviation values computed using the Microsoft Excel spreadsheet. The formula used to compute the entries of column 5 is as follows:

$$Error (\%) = \frac{|Stdev_Excel - Stdev_Online_Laptop| \times 100 \%}{Stdev_Excel} \quad (7.3)$$

The results shown in Table 7.1 show that the standard deviation values obtained using the proposed online algorithm while running on a laptop computer are exactly equal (up to 6 decimal places) to the corresponding values obtained while running the algorithm on a

desktop computer. While the standard deviation values presented in Table 7.1 have a precision of up to 6 decimal places for the sake of presentation convenience, the actual values obtained in the experiments showed equality between corresponding values of column 3 and 4 with a precision of up to 20 decimal places; and there was no sign of having different values beyond that point for the data sets, which had non-zero digits beyond 20 decimal places. This observation shows that the proposed online algorithm is capable of producing consistent results in different computing devices, and therefore it is a valid algorithm to be used in TAPMRP for determining the standard deviation of neighbours of nodes in a network.

Table 7.1 Standard deviation values obtained using different approaches

Dataset size	Stdev-Excel	Stdev-online, laptop	Stdev-online, desktop	Error (%)
10	10.795575	10.795575	10.795575	0.000000
15	10.653414	10.653415	10.653415	0.000009
20	9.750034	9.750035	9.750035	0.000010
25	9.302867	9.302865	9.302865	0.000021
30	8.543936	8.543933	8.543933	0.000035
35	8.242644	8.242644	8.242644	0.000000
40	8.322968	8.322965	8.322965	0.000036
45	7.955558	7.955562	7.955562	0.000050
50	7.764414	7.764415	7.764415	0.000013
55	8.662859	8.662862	8.662862	0.000035

A look at the error (column 5) associated with the proposed online algorithm shows that it is very small in magnitude. In fact, the largest error in column 5 is 0.00005% which is too small to negatively affect the usefulness (or the application) of the standard deviation values of the neighbours of nodes in the model ELC-topologies considered in this research. This observed small magnitude in the error presented in column 5 further confirms that the proposed online algorithm is indeed valid for use in TAPMRP.

7.3.3 Justification for separating the ELC-MANET into phases

The results of the first step of the simulation test on the justification for separating the ELC-MANET into phases as proposed in this thesis were plotted as shown in Figure 7.5. The packet delivery ratio that appears on the vertical axis of the graph in Figure 7.5 is defined as follows:

$$Packet_delivery_ratio = \frac{total_packets_received}{(total_packets_sent \times total_multicast_receivers)} \quad (7.4)$$

The term “total_packets_received” in Equation (7.4) is the sum of all the packets including those that piggybacked Join-Queries received by all the multicast receivers of a simulated network. The multiplication of the term “total_packets_sent” in Equation (7.4) by the term “total_multicast_receivers” comes from the fact that the packets from the source node were transmitted to multicast receivers by multicasting which effectively meant that the packets were duplicated a number of times equal to the total number of the multicast receivers which is denoted by “total_multicast_receivers” in Equation (7.4). Each value of the average packet delivery ratio used to plot the graph of Figure 7.5 was obtained by adding 10 corresponding values of the packet delivery ratio obtained in the 10 runs of the simulation and then divided the total by 10. Note that each receiver sensitivity threshold setting had ten runs of the simulation for each ELC-MANET topology.

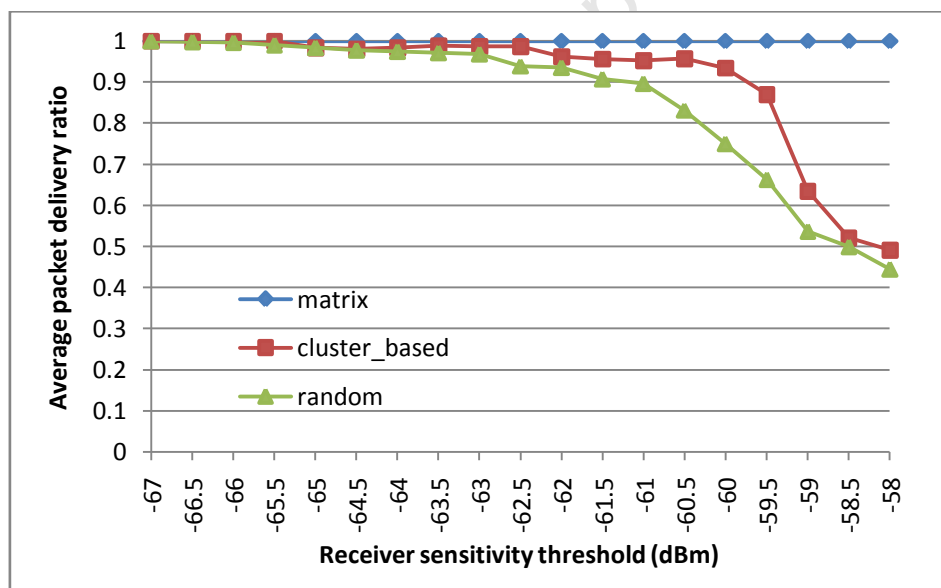


Figure 7.5 ODMRP’s performance in the three ELC-MANET topologies

Figure 7.5 indicates that the average packet delivery ratio in the ELC-matrix topology is 1.0 for the whole range of the receiver sensitivity threshold values that was considered. The ELC-matrix topology has such excellent performance possibly because at the maximum setting of the receiver sensitivity threshold value of -58dBm which yielded a fully connected network for all the three ELC-MANET topologies, the ELC-matrix network already had very high connectivity such that each node had all the other nodes (i.e. 54 nodes) in the network as

its one-hop neighbours. These results confirm the observation in this thesis that the ELC-matrix topology has higher connectivity than the other two ELC-MANET topologies. This high connectivity of the ELC-matrix topology is due to the smaller geographical network area associated with the ELC-matrix topology compared to the network areas of the other two ELC-MANET topologies. Note that what makes the ELC-matrix topology to have a relatively small network area is not the nature of the topology but its location in a classroom area which has a relatively small geographical area compared to the areas normally occupied by the other two ELC-MANET topologies.

At -67dBm in Figure 7.5, all the three ELC-MANET topologies were completely connected such that each node in a network had all the other nodes as its one-hop neighbours (i.e. 54 one-hop neighbours). As the transmission range was decreasing through increasing the receiver sensitivity threshold values, the average number of neighbours per node in the ELC-cluster-based network was decreasing such that at -58dBm the average number of neighbours per node was 18.8. This trend in the average number of neighbours shows that network connectivity level was decreasing in the direction of increasing receiver sensitivity threshold settings (i.e. from -67dBm towards -58dBm) and it is this connectivity trend which is possibly responsible for the performance trend of the ELC-cluster-based topology in Figure 7.5. The understanding here is that when connectivity level is low packets do not have enough energy for them to be detected by the destination nodes, resulting in loss of some packets. Hence, the rate of packet loss increases with the decrease in network connectivity.

The performance of the ELC-random topology is generally lower than that of the other two ELC-MANET topologies in the graph of Figure 7.5 possibly due to two reasons. The first reason is that the connectivity level of the ELC-random topology is the weakest of the three topologies at any given receiver sensitivity threshold setting mainly due to the fact that the ELC-random topology has the largest network area. The second reason is node mobility which made nodes miss packets.

Figure 7.6 shows the effect of the differences in the topologies only between the ELC-matrix and the ELC-cluster-based topology on the performance of ODMRP in the two topologies. Note that it may not be sensible to attribute the behaviour of the results shown in Figure 7.5 to the differences in the nature of the topologies of the three considered ELC-MANET phases because the effect of network connectivity resulting from the differences in the geographical

areas of the networks used in the simulation significantly outweigh the effect of the differences in the nature of the topologies on the results.

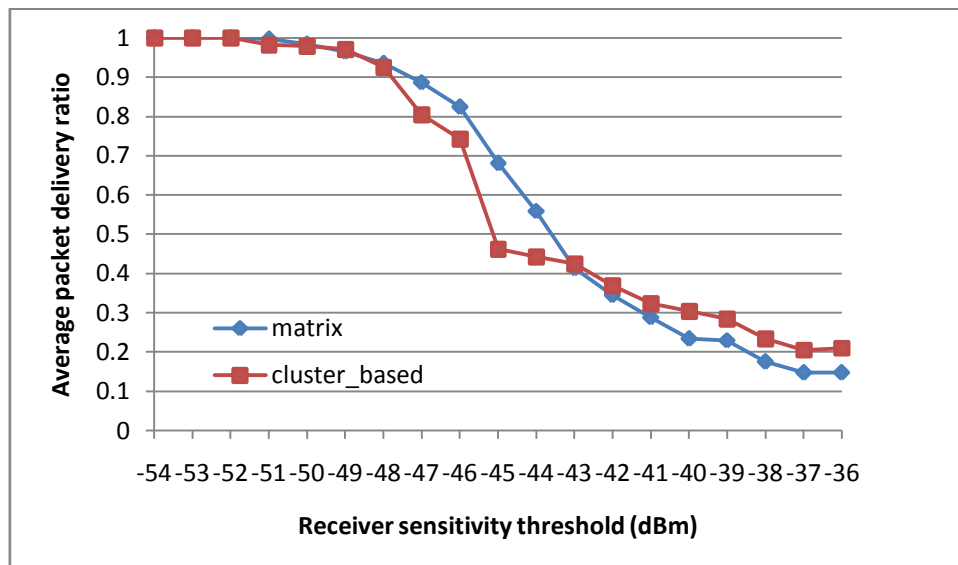


Figure 7.6 Effect of ELC-MANET topology differences on ODMRP’s performance

Between -52dBm and -43dBm in Figure 7.6, the ELC-matrix topology performs better than the ELC-cluster-based topology. A possible reason for this performance observation is that connectivity links associated with bridge (or gateway) nodes of clusters in the ELC-cluster-based network act as single points of failure for the clusters which they serve resulting in less overall packet delivery ratio for the network compared to the corresponding overall packet delivery ratio of the ELC-matrix topology in which such single points of failure do not exist.

From -43dBm up to -36dBm in Figure 7.6, the performance trend has changed such that the ELC-cluster-based topology now performs better than the ELC-matrix topology. A possible explanation for this observation is based on the understanding that connectivity is higher within a cluster than anywhere else outside the cluster. Therefore, the high packet delivery ratio from the nodes belonging to the cluster to which the source node also belongs contributes significantly to the overall packet delivery ratio of the network; while in the ELC-matrix network nodes are almost evenly distributed resulting in weak connectivity everywhere in the network, which in turn results in less overall packet delivery ratio than that of the ELC-cluster-based network.

The observation in this thesis is that the differences in the performance of ODMRP in the two ELC-MANET topologies that yielded the results of Figure 7.6 are quite significant. For instance, at -45dBm the difference in the average packet delivery ratios of the two topologies is about 0.22 and expressing this as a percentage of the average delivery ratio of the ELC-matrix topology, we have $(0.22 \div 0.68 \times 100\%)$ which gives approximately 32.4%.

Note that the variation of the connectivity level that resulted from the receiver sensitivity threshold settings shown in Figure 7.6 was due to the average distance of 1.5m between adjacent nodes in the simulated network. Increasing the distance between adjacent nodes could have resulted in yielding the same results of Figure 7.6 but at receiver sensitivity threshold settings with lower values, for example -60dBm as the maximum, than those used in Figure 7.6. This implies that actual values of the receiver sensitivity settings as shown in Figure 7.6, and also in any other graph of this thesis, are not important but the connectivity level that result from them. In practice, the ELC-matrix topology can have an average distance of more than 1.5m between adjacent nodes.

This research considers the differences in the performance of ODMRP shown in Figure 7.5 and 7.6 to be significant enough to necessitate the separation of the three distinct phases of the ELC-MANET with the intention of optimising performance, not only in terms of the packet delivery ratio but also in terms of the number of forward nodes required by a route discovery process, in each one of them. Further justification for separating the three distinct topologies of the ELC-MANET is provided by the performance, in terms of the required number of forward nodes of control packets, of the proposed route discovery methods for the distinct topologies (or phases) covered in the subsequent paragraphs.

7.4 Matrix Topology Route Discovery Method

The first set of data aimed at comparing the number of forward nodes used by MTRDM and ODMRP were plotted as shown in Figure 7.7. The vertical axis of Figure 7.7 gives the average percentage of forward nodes in a network and is herein defined as follows:

$$\text{Average_percentage_of_forward_nodes} = \frac{\text{Average_number_of_forward_nodes} \times 100\%}{\text{Possible_number_of_forward_nodes}} \quad (7.5)$$

The term “average_number_of_forward_nodes” appearing in Equation (7.5) was obtained by dividing by ten the sum of the number of forward nodes obtained in the ten runs of the experiment for each setting of the receiver sensitivity threshold (refer to section 6.4 of chapter 6 for details). “Possible_number_of_forward_nodes” that appears in Equation (7.5) is given by $n - 1$, where n is the size of the network under consideration and -1 is used to exclude a source node from the list of possible forward nodes. The simulated network which yielded the results of Figure 7.7 had 55 nodes and therefore the possible number of forward nodes according to Equation (7.5) was 54.

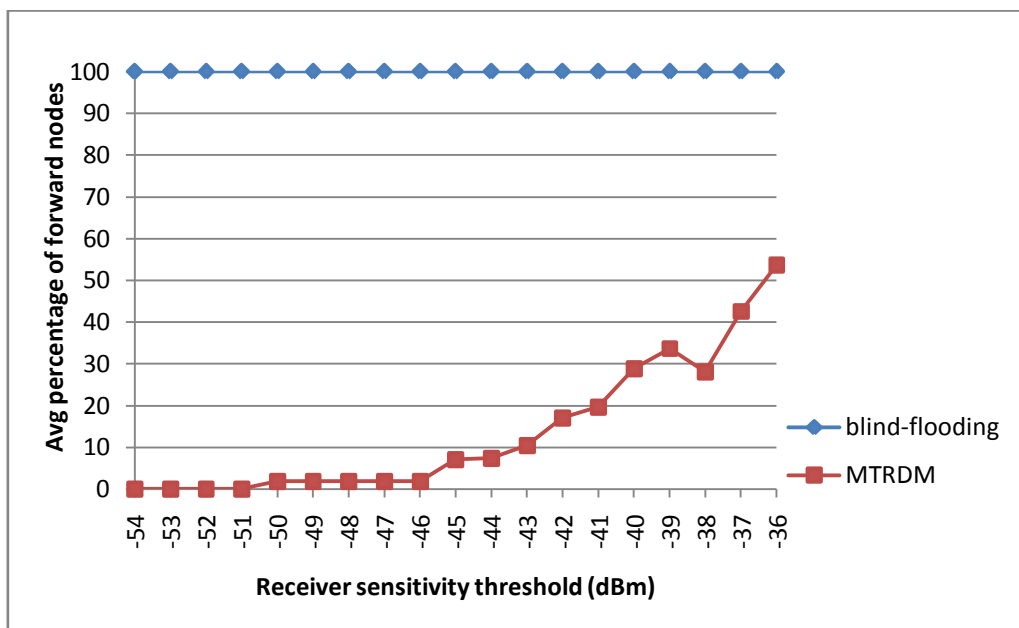


Figure 7.7 Number of forward nodes in MTRDM and the blind flooding method of ODMRP

The graph of Figure 7.7 clearly shows MTRDM uses many fewer forward nodes than the blind flooding method for a complete network coverage of the Join-Query packet at almost all the settings of the receiver sensitivity threshold (or transmission range) that were considered. At -54dBm the average number of neighbours per node in the network was 54 which was the maximum possible number of neighbours per node. This implies that at -54dBm the network connectivity was very high. As the settings of receiver sensitivity were increasing in value from -54dBm towards -36dBm, the network connectivity was becoming weaker and weaker such that, for example, at -45dBm and -40dBm the average number of neighbours per node was 30.8 and 10.7 respectively. Hence, the number of forward nodes required by MTRDM increases as network connectivity decreases.

The results of Figure 7.7 further show that the number of forward nodes required by the blind flooding method for complete network coverage of broadcasting the Join-Query packet is independent of the connectivity characteristics of the underlying network. The blind flooding method uses all the nodes in the network as forward nodes except the source node of the Join-Query under consideration. Clearly, the blind flooding method wastes energy for performing retransmissions which are redundant. For instance, between -54dBm and -51dBm in Figure 7.7 the blind flooding method required that all the nodes in the network other than the source node should rebroadcast the Join-Query packet which is not necessary since all the nodes in the network are directly connected to one another. Hence, compared to the blind flooding method, MTRDM is much more efficient in terms of energy usage for transmissions.

Figure 7.8 shows a performance comparison between MTRDM and the blind flooding method of ODMRP based on the packet delivery ratio. The results of Figure 7.8 indicate that, in general, MTRDM has a slightly better packet delivery ratio than the blind flooding method at a given transmission range (or receiver sensitivity threshold setting). Between -47dBm and -39dBm MTRDM has higher values of the packet delivery ratio possibly due to a reduction in packet collisions in the shared transmission medium resulting from the pruning effect of forward nodes for Join-Query packets. The fluctuation of the graph line for MTRDM is possibly due to the topology of the nodes in relation to the pruning technique of forward nodes used by the method.

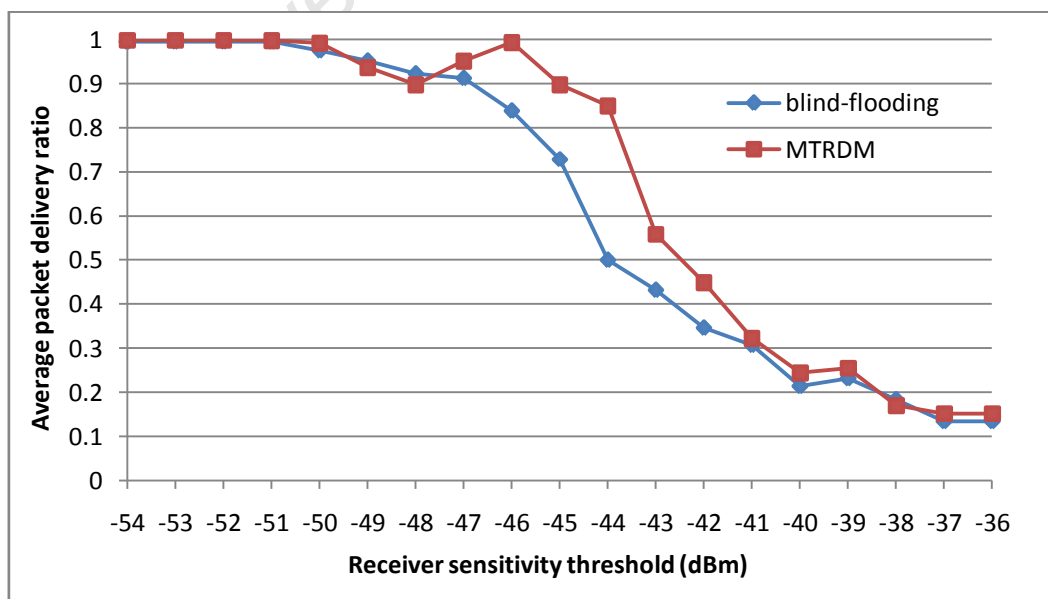


Figure 7.8 Performance comparison between MTRDM and the blind flooding method

The observation from the results of Figure 7.8 is that MTRDM is not only better than the blind flooding method in terms of the number of the required forward nodes but also in terms of the packet delivery ratio. In fact, even if the packet delivery ratios of the two methods were exactly the same for a given transmission range, MTRDM would still be more suitable for the ELC environment because of its savings on energy which may not be directly reflected in a packet delivery ratio.

The graph of Figure 7.9 compares the packet delivery ratio of MTRDM for three speeds of node mobility in the ELC environment. According to Figure 7.9, on average stationary nodes (i.e. nodes moving at 0m/s) have the highest packet delivery ratio for a given transmission range. Between -48dBm and -41dBm, the values of the packet delivery ratio of both 1.5m/s and 10m/s are lower than those of 0m/s most probably due to the mobility of the nodes which made them miss the packets.

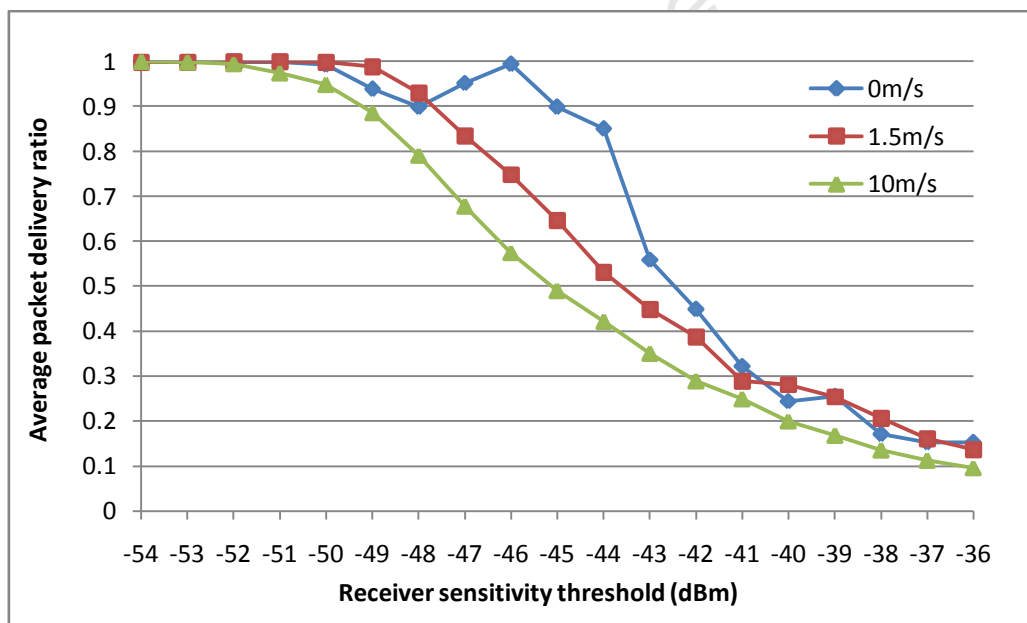


Figure 7.9 Effect of node mobility on the performance of MTRDM

In moving from a receiver sensitivity of -48dBm towards -54dBm in Figure 7.9, the values of the packet delivery ratio for all the three speeds converge to the maximum possible value of 1.0. This observation indicates that the effect of node mobility on the performance of MTRDM decreases as network connectivity increases. Since the ELC environment for which MTRDM is intended is characterised by high network connectivity, MTRDM is still suitable for the environment from the viewpoint of node mobility. From the results of Figure 7.9 we

can deduce that MTRDM can perform poorly in the ELC-random topology where mobility is relatively high and connectivity is relatively low. In addition, MTRDM may not perform excellently in the ELC-cluster-based topology due to a combination of the presence of some mobility and low connectivity.

7.5 Cluster-based Topology Route Discovery Method

7.5.1 Effect of introducing secondary cluster-heads

The first set of data collected in connection with the need to observe the effect of introducing secondary cluster-heads in a one-hop clustering algorithm was plotted as shown in Figure 7.10. The graph shown in Figure 7.10 compares the number of forward nodes required for a complete broadcast of signalling packets generated by three different source nodes in a network in the two clustering algorithms considered in this simulation. The clustering algorithm which had only one cluster-head (primary cluster-head) per cluster is represented in Figure 7.10 by the graph line labelled “without sec. CHs”. The other clustering algorithm had secondary cluster-heads and is represented in Figure 7.10 by the graph line labelled “with sec. CHs”. The explanation for the term “average percentage of forward nodes” appearing on the vertical axis of the graph of Figure 7.10 is given by Equation (7.5).

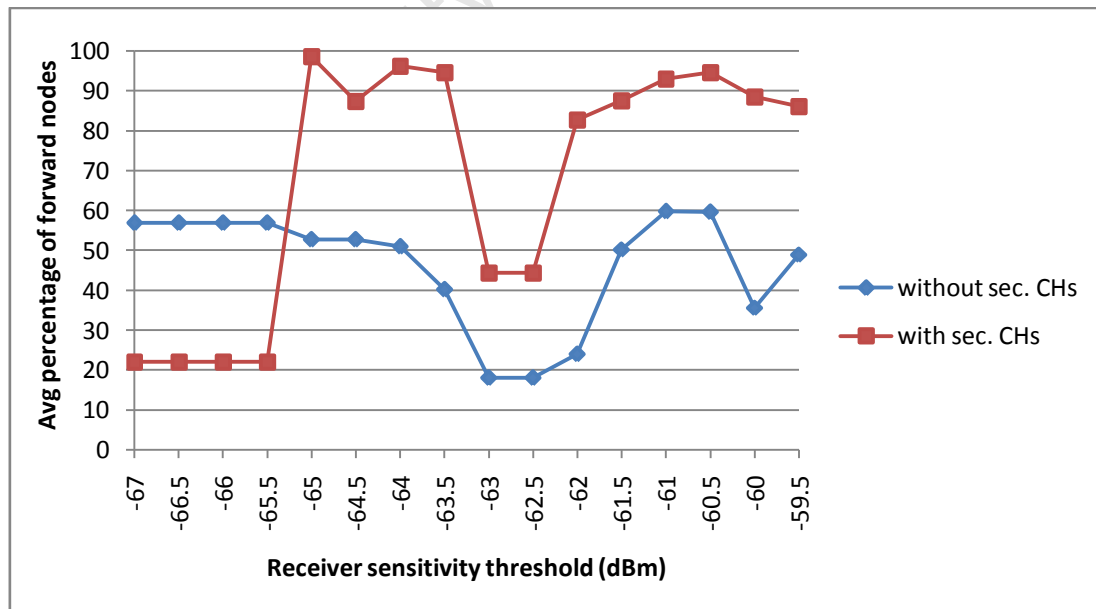


Figure 7.10 Number of forward nodes of CTRDM’s clustering algorithms on secondary cluster-heads

The level of connectivity of the network in Figure 7.10 decreases as the receiver sensitivity settings change from -67dBm towards -59.5dBm. For instance, at -62.5dBm and -59.5dBm the average number of neighbours per node in the network was 46.4 and 28.9 respectively. Between -67dBm and -65.5dBm the network was completely connected such that any two nodes in the network were direct neighbours and each node had 54 nodes as its one-hop neighbours.

Ideally, between -67dBm and -65.5dBm, we expect both clustering algorithms to have no forward nodes (i.e. both graph lines to have zero values of forward nodes) because the connectivity is so high such that every node has any other node in the network as a neighbour. The forward nodes that appear between -67dBm and -65.5dBm are those nodes that failed to receive clustering signals. In fact, in this research, nodes that fail to receive a clustering signal act as cluster-heads, which also act as forward nodes. Between -65dBm and -59.5dBm, the average number of forward nodes for the clustering algorithm with secondary cluster-heads is significantly more than that of the clustering algorithm without secondary cluster-heads for each receiver sensitivity setting. The reason why a clustering algorithm without secondary cluster-heads has more forward nodes than the other algorithm between -67dBm and -65.5dBm is mostly due to a probabilistic chance arising from the positioning of the cluster-heads in the network.

The use of more forward nodes in the clustering algorithm with secondary cluster-heads seems to defeat the purpose of using a clustering approach in CTRDM which is on reducing the number of forward nodes. The use of a maximum of two secondary cluster-heads ensures that the secondary cluster-heads should provide just enough load balancing without compromising too much on the broadcast (or multicast) efficiency which the proposed method is expected to offer.

Figure 7.11 presents a graph which compares the workload processed by each forward node in the two clustering algorithms. The vertical axis of the graph of Figure 7.11 represents the number of retransmissions of signalling packets performed by a forward node as a percentage of all the signalling packets generated by the three sources in the simulated network. The term “average” that appears on the vertical axis implies that each value of the percentage is a representative of the corresponding values obtained in the ten runs of the simulation.

The results of Figure 7.11 confirm that the introduction of secondary cluster-heads helps in spreading the task of propagating signalling packets over a number of forward nodes in a network. Note that the task of transmitting the multicast data packets which follow the route discovery operations is also spread over a number of forward nodes in the network. The transmission of data packets is spread due to the fact that a primary cluster-head and its two secondary cluster-heads act separately as forward nodes in three different multicast trees. The high fluctuations in the graph lines of Figure 7.11 are likely due to the combined effect of the probabilistic nature of wireless environment and the irregular arrangement of nodes the ELC-cluster-based topology. Note that between -67dBm and -65.5dBm in Figure 7.11 the algorithm without secondary cluster-heads has higher workload percentages than the other algorithm as it is expected because the three sources used one cluster-head while in the other algorithm the sources were affiliated to different cluster-heads who shared the workload.

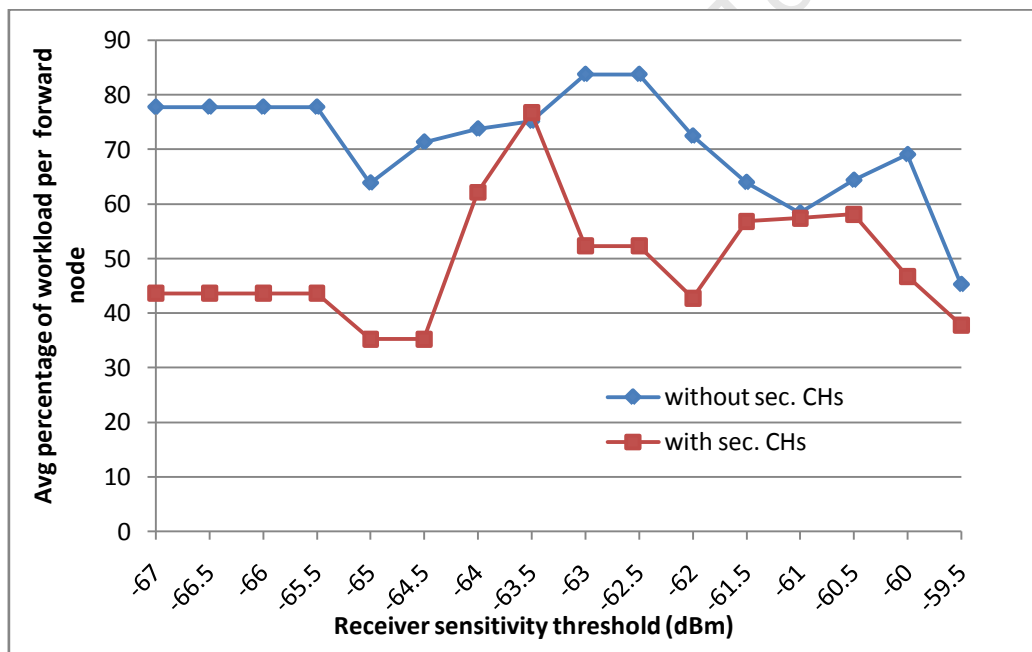


Figure 7.11 Workload per forward node in CTRDM’s clustering algorithms on secondary cluster-heads

Performance, in terms of the packet delivery ratio, of the two clustering algorithms of CTRDM considered in this subsection is compared in Figure 7.12. The observation is that the two clustering algorithms of CTRDM have almost the same performance which implies that the introduction of secondary cluster-heads has no negative effect on the packet delivery ratio with respect to the situation of having no secondary cluster-heads.

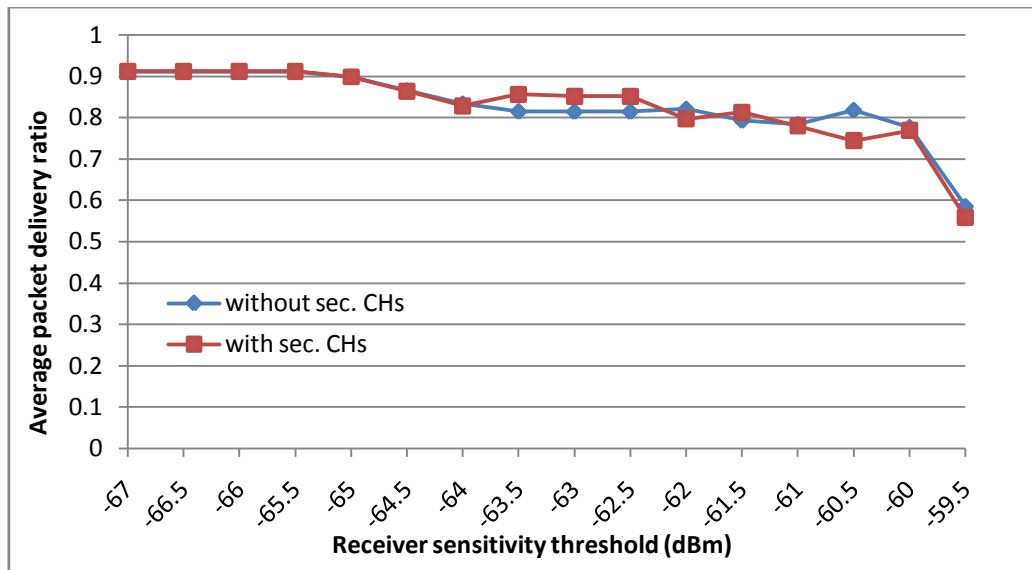


Figure 7.12 Performance of CTRDM’s clustering algorithms on secondary cluster-heads

7.5.2 Effect of the node redistribution parameter

In Figure 7.13, the graph shows a comparison between the number of forward nodes required in the CTRDM’s clustering algorithm that used degree connectivity for cluster-head selection and the number of forward nodes required in the other CTRDM’s clustering algorithm that made use of the node redistribution parameter for selecting nodes for the cluster-head role. Each value of the percentage of forward nodes indicated on vertical axis of the graph of Figure 7.13 is the one that was needed for a complete broadcast of Join-Query packets from all the three source nodes at a given transmission range in both clustering algorithms. Note that the meaning of the term “average percentage of forward nodes” appearing on the vertical axis of the graph of Figure 7.13 is given by Equation (7.5).

In Figure 7.13, like in all the figures of this section, connectivity of the network was decreasing as the receiver sensitivity settings were changed from -67dBm towards -59.5dBm. Between -65dBm and -59.5dBm the number of forward nodes required in the CTRDM’s clustering algorithm that used the node redistribution parameter is higher than in the other clustering algorithm because the node redistribution parameter has the effect of increasing the number of logical clusters in a network which in turn increases the number of forward nodes. The increase of the number of logical clusters due to the effect of the node redistribution parameter does not exceed the number of physical clusters available in the network; this condition ensures that the use of the node redistribution parameter does not compromise too

much on the main goal of this research which is on reducing the number of forward nodes of signalling packets required in multicasting.

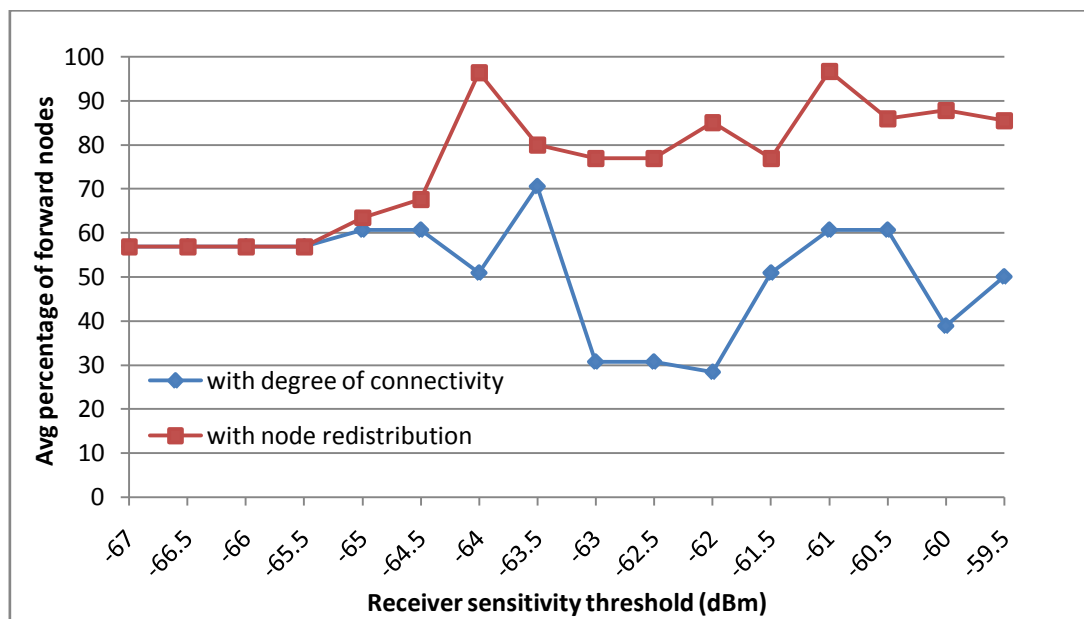


Figure 7.13 Forward nodes in CTRDM’s clustering algorithms on the node redistribution parameter

Between -67dBm and -65.5dBm in Figure 7.13, the number of forward nodes required in the two clustering algorithms of CTRDM on the node redistribution parameter is the same due to the fact that the connectivity of the network is so high such that every node has all the other nodes as its neighbours, which in turn results in removing the effect of the node redistribution parameter. In fact under ideal situations there should be no forward nodes between -67dBm and -65.5dBm. In this practical situation the presence of forward nodes arises from nodes that fail to receive clustering signals. Note that in this research a node that fails to receive a clustering signal is treated as a cluster-head that also acts as a forward node.

The workload per forward node in the two clustering algorithms of CTRDM on the node redistribution parameter is compared in the graph of Figure 7.14. In Figure 7.14, the vertical axis of the graph represents the number of retransmissions of Join-Query packets performed by a forward node as a percentage of all the Join-Query packets generated by the three sources in the simulated network. Each value of the percentage given by the vertical axis of the graph of Figure 7.14 is a representative of the corresponding values obtained in the ten runs of the simulation, and hence it is associated with the term “average” that appears on the axis.

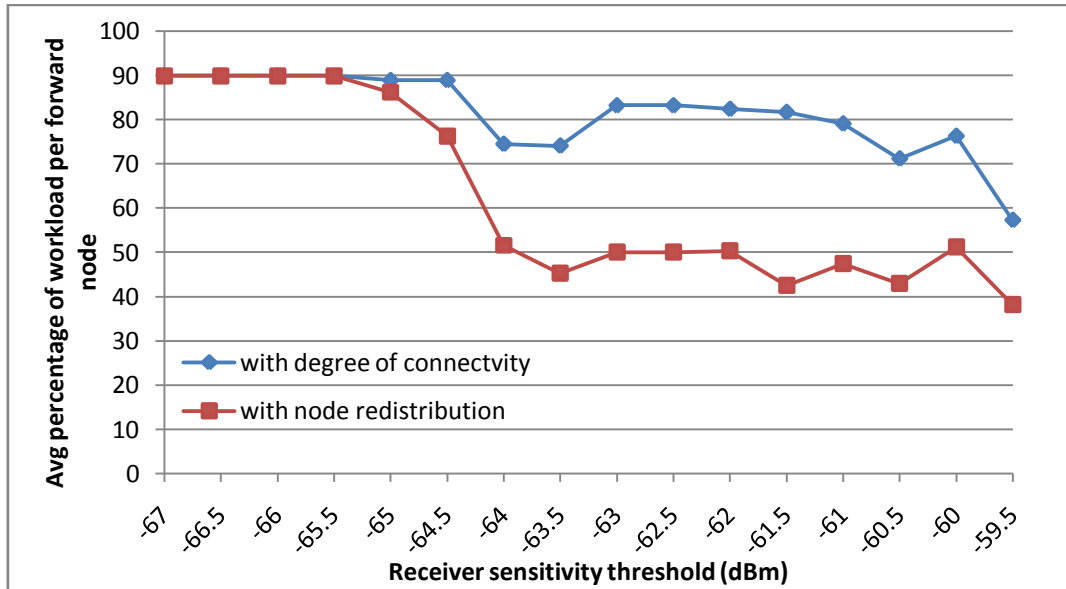


Figure 7.14 Workload per forward node for the two CTRDM's clustering algorithms on the node redistribution parameter

The results of Figure 7.14 confirm that the use of the node redistribution parameter helps in distributing the task of propagating control packets over a number of forward nodes in a network. Note that the transmission of the multicast data packets that follow the route discovery operations has a high probability of being distributed also over a number of forward nodes in the network; this is due to the fact that the cluster-heads of the source nodes have a high probability of acting separately as forward nodes in different multicast trees. On average, the difference in the percentage of the workload between the two versions of the cluster-based method in the three-source scenario of Figure 7.14, excluding the portion between -67dBm and -65.5dBm, is about 25%, which is quite significant. The difference is big enough to provide significant load balancing, which can lead to a reduction in the occurrences of node failures arising from faster depletion of battery energy. Note that load balancing increases with the increase in the number of sources belonging to different clusters.

The graph of Figure 7.15 compares the performance, in terms of the packet delivery ratio, of the two clustering algorithms of CTRDM on the node redistribution parameter. The conclusion from the behaviour of the graph of Figure 7.15 is that the two clustering algorithms yield almost the same packet delivery ratio at a given transmission range. Hence, with respect to the use of the degree of connectivity in clustering, the use of the node redistribution parameter does not negatively affect the performance of a multicast service in terms of the packet delivery ratio.

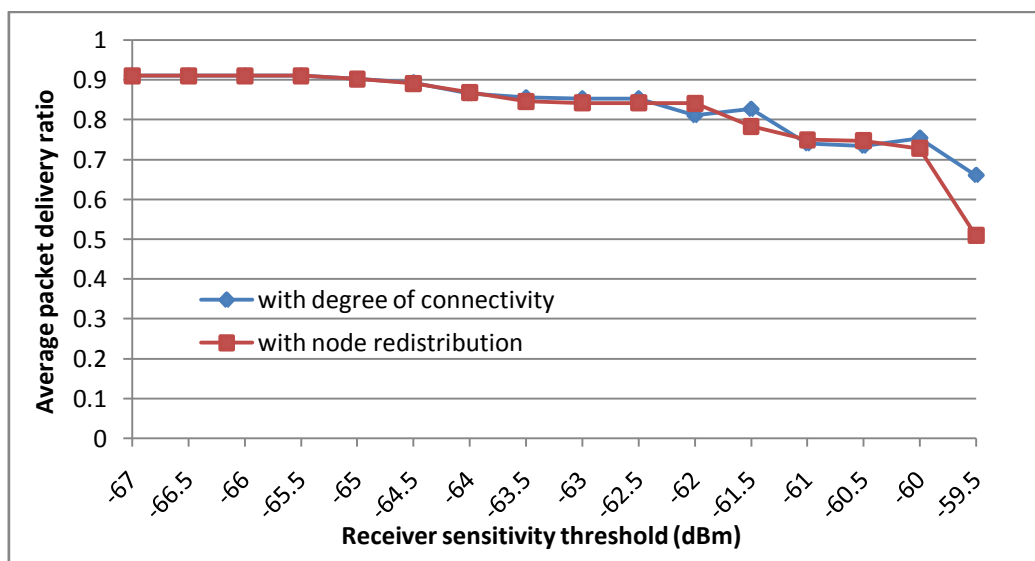


Figure 7.15 Performance of the two CTRDM's clustering algorithms on the node redistribution parameter

7.5.3 Effect of the node mobility formula

In Figure 7.16, the graph compares the performance of the two clustering algorithms of CTRDM on node mobility in terms of the packet delivery ratio. From -62dBm to -59.5dBm connectivity of the network is relatively low such that the mobility formula (i.e. Equation (4.4)) is able to detect accurately mobility of nodes in and out of one another's transmission range resulting in high probability of selecting the least mobile nodes for the cluster-head role. Thus, between -62dBm and -59.5dBm, the graph line in Figure 7.16 representing the clustering algorithm that used the mobility equation for cluster-head selection is higher than the graph line representing the other clustering algorithm that made use of degree of connectivity most likely due the selection of stable cluster-heads that acted as forward nodes for multicast packets.

Between -64.5dBm and -62.5dBm, the connectivity of the network is relatively high such that there is much overlapping of transmission ranges of nodes which result in the mobility equation failing to detect accurately mobility of nodes. Another consequence of relatively high connectivity is that the impact of mobility of forward nodes on the packet delivery ratio is likely to be very small. Based on these two observations, between -64.5dBm and -62.5dBm, the two graph lines of Figure 7.16 are expected to be very close to each other. However, this has not been the case. A sudden drop in graph line representing the clustering algorithm that made use of the mobility equation between -63.5dBm and -62.5dBm is

possibly due to the cluster-based configuration of the network which created a bottleneck (or bottlenecks) to the flow of traffic in the network. Such a bottleneck (or bottlenecks) was by chance overcome by the other clustering algorithm (the one that used degree of connectivity for cluster-head selection) which perhaps used mobile forward nodes to create alternative links for the flow of traffic.

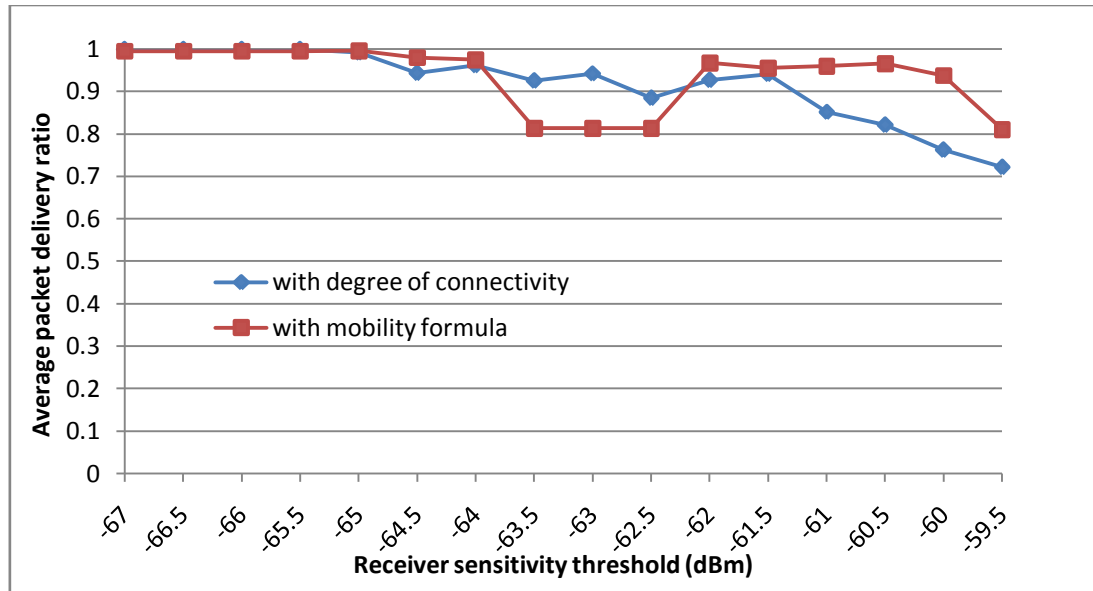


Figure 7.16 Performance of the two clustering algorithms of CTRDM on node mobility

When connectivity of the simulated network was so high between -67dBm and -65.5dBm such that any node in a network had any other node as its neighbour, the use of Equation (4.4) for detecting node mobility became completely ineffective. Based on the results of Figure 7.16 we can conclude that when the network is completely connected as the case is between -67dBm and -65.5dBm, it does not matter which clustering algorithm is used for cluster-head selection because both clustering algorithms of CTRDM on node mobility yield the same maximum possible packet delivery ratio of 1.0.

The overall conclusion on the use of the mobility formula as a criterion for cluster-head selection is that it is effective and yields positive results when network connectivity is relatively low. According Figure 7.16, the connectivity level in which the mobility equation is effective can be approximated as starting from the receiver sensitivity threshold setting of about -63dBm up to -59.5dBm and beyond (i.e. from -63dBm and then move in the direction of decreasing connectivity). Note that at -63dBm and -59.5dBm the average number of neighbours per node in the simulated network was 46.4 and 28.9 respectively. This relatively

low connectivity level (i.e. from -63dBm up to -59.5dBm and beyond as the case is in Figure 7.16) is the one which is likely to be associated with the ELC-cluster-based topology considered in this thesis, and therefore the mobility equation is relevant to the ELC-cluster-based topology.

When connectivity is relatively high such as the one from -63dBm down to -67dBm and beyond in Figure 7.16, the switching mechanism of the route discovery methods of TAPMRP can most likely deploy MTRDM to serve such an environment regardless of the arrangement of nodes. This does not mean that the ELC-cluster-based topology becomes the ELC-matrix topology when connectivity is very high, but it only implies that the connectivity level has reached the point of being associated with the ELC-matrix topology defined in this research. In fact, creating logical clusters (by a clustering algorithm) in an environment of high connectivity is a waste of network resources including energy because multicast packets can be delivered reliably and more efficiently by MTRDM without the use of clusters.

7.5.4 Performance of CTRDM as a whole

The simulation test on CTRDM started with the collection of data plotted in Figure 7.17 on the number of forward nodes used for broadcasting of Join-Query packets in CTRDM and the blind flooding method of ODMRP. The explanation of the vertical axis of the graph of Figure 7.17 is given by Equation (7.5). The number of forward nodes shown in Figure 7.17 used by CTRDM for complete broadcast operations of control packets is much less than that used by the blind flooding method to achieve the same goal due to the fact that in CTRDM forwarding of control packets is done by cluster-heads and selected bridge nodes only while in the blind flooding method every node in the network is a forward node.

Basing on the range of the receiver sensitivity settings used in Figure 7.17, on average CTRDM uses approximately only 23% of the forward nodes required by the blind flooding method. Without regarding the overhead contributed by the hello messages required in clustering and effects of mobility, it can be translated that CTRDM saves about 77% of the energy that is used in the retransmissions of signalling packets (i.e. the Join-Query packets) by the blind flooding method. Considering that the interval of hello messages can be tuned in accordance with the level of mobility of a network and that CTRDM is intended for cluster-based topologies with relatively low mobility, CTRDM can still save a significant amount of energy compared to the blind flooding method.

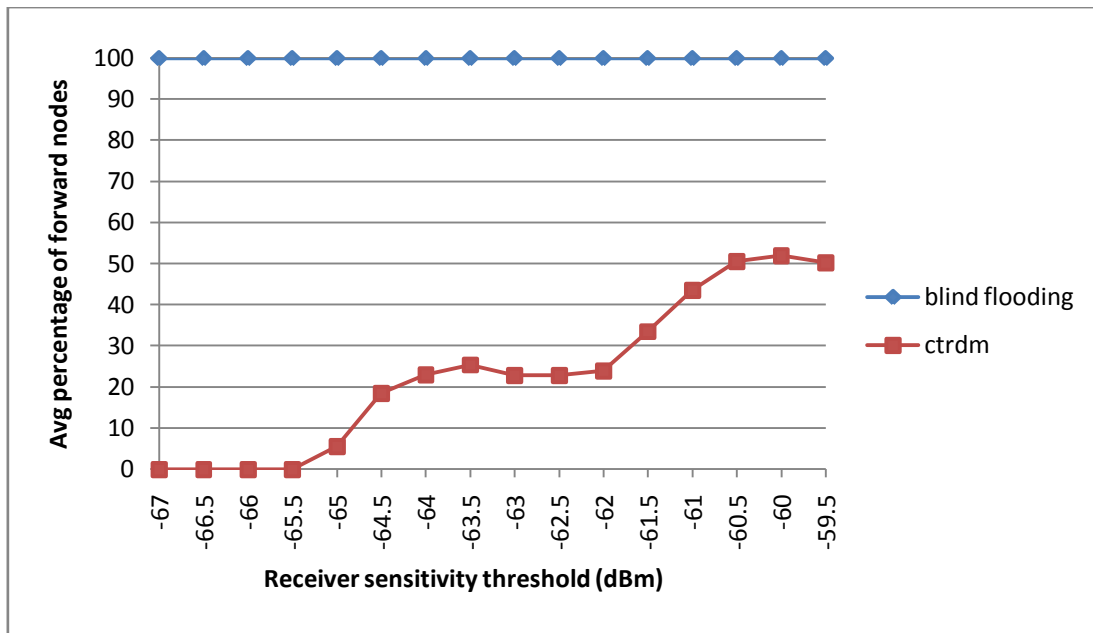


Figure 7.17 Number of forward nodes in CTRDM and the blind flooding method

The graph of Figure 7.18 compares the performance, in terms of the packet delivery ratio, of CTRDM with that of the blind flooding method of ODMRP. The conclusion from Figure 7.18 is that CTRDM has almost the same packet delivery ratio performance as that of the blind flooding method. This observation is most likely to the use of nodes that fail to receive clustering signals as cluster-heads that also act as forward nodes. By using such nodes as cluster-heads ensures that every node in the network is part of a clustering system, which in turn results in a reliable multicast service as evidenced by the results of Figure 7.18. The price that is paid for using forward nodes arising from nodes that fail to receive clustering signals is that there are generally more forward nodes in the network than in the case of not using such nodes. However, the price paid is significantly outweighed by the benefits of having a reliable multicast (or broadcast) service.

The effects of low mobility that characterise the ELC-cluster-based topologies defined in this thesis on the performance of CTRDM are shown in Figure 7.19 and 7.20. The graph of Figure 7.19 shows a comparison on the number of forward nodes required for a complete broadcast of the Join-Query packets in involving CTRDM with mobility, CTRDM without mobility and the blind flooding method of ODMRP with mobility. As explained in subsection 6.5.3 and 6.5.4 of the methodology chapter, only 12 nodes were allowed in the simulation to continuously move between clusters at an average speed of 1.5m/s.

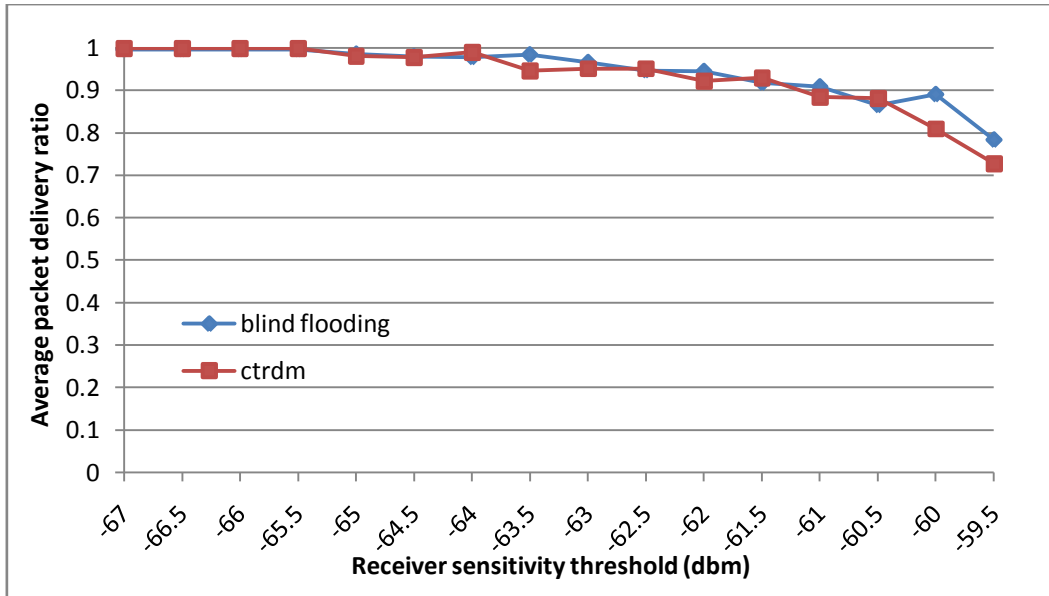


Figure 7.18 Performance of CTRDM and the blind flooding method

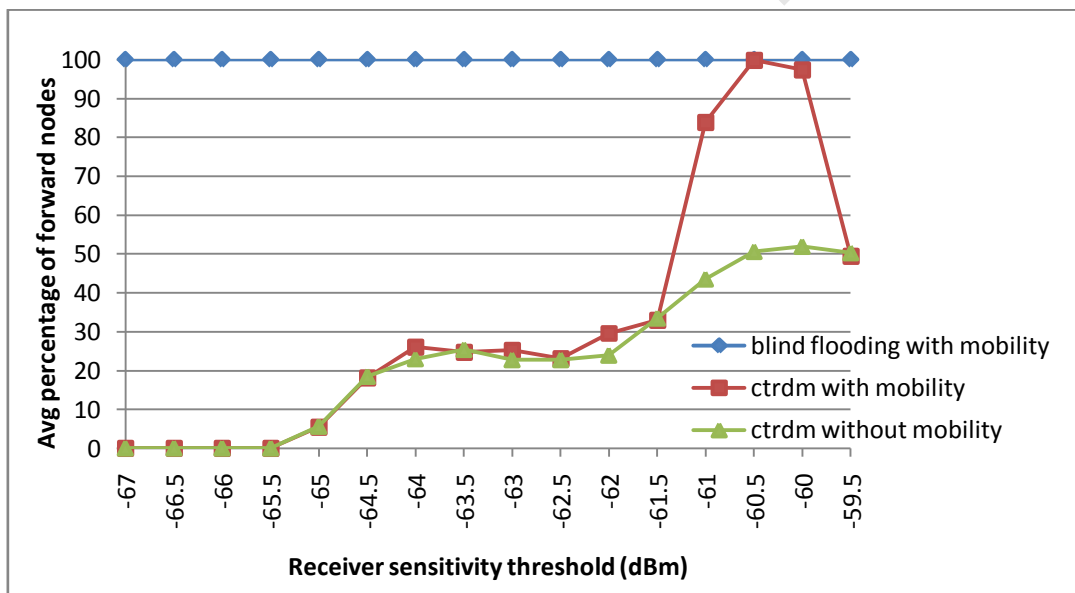


Figure 7.19 Forward nodes in CTRDM with and without mobility and in the blind flooding with mobility

Without including the portion of the graph between -61dBm and -60dBm in Figure 7.19, the number of forward nodes required by CTRDM with mobility is slightly higher than the number of forward nodes required by CTRDM without mobility at a given transmission range. The much higher values of the number (or percentage) of forward nodes required by CTRDM with mobility, with respect to the corresponding values of CTRDM without mobility, shown between -61dBm and -60dBm is possibly due to a joint effect of mobility and a resulting network configuration which together favoured the creation of forward nodes

arising from nodes that failed to receive clustering signals. Note that in this research a node that fails to receive a clustering signal acts as a cluster-head which also acts as a forward node.

In spite of the behaviour of the graph line of CTRDM with mobility between -61dBm and -60dBm in Figure 7.19, the number of forward nodes required by CTRDM in the presence of low mobility considered in this research is much less than that required by the blind flooding method of ODMRP. In fact, based on the results of Figure 7.19, CTRDM with mobility requires approximately only 32% of the number of forward nodes required by the blind flooding method on average. This implies that CTRDM is capable of using significantly fewer forward nodes of control packets than the blind flooding method when mobility is relatively low.

Shown in Figure 7.20 is a performance comparison in terms of the packet delivery ratio involving CTRDM with mobility, CTRDM without mobility, and the blind flooding method with mobility. The conclusion from the results of Figure 7.20 is that CTRDM offers delivery of multicast packets which is as reliable as that of the blind flooding method when mobility is relatively low. It is sensible to suggest that the reliable delivery of packets during mobility in CTRDM is partly due to the use of forward nodes that come from nodes that fail to receive clustering signals.

The overall conclusion on CTRDM is that it is significantly more efficient than the blind flooding method of ODMRP in terms of using energy for broadcasting control packets. CTRDM is intended for use in the ELC-cluster-based network environment characterised by relatively low mobility. Using CTRDM in an environment of very high connectivity like that the one associated with the ELC-matrix topology as defined in this thesis may lead to wastage of network resources due to unnecessary clustering. CTRDM is not suitable for environments of high mobility like the one associated with the ELC-random topology defined in this research because it may lead to generating excessive control overhead in an attempt to achieve reliable active clustering.

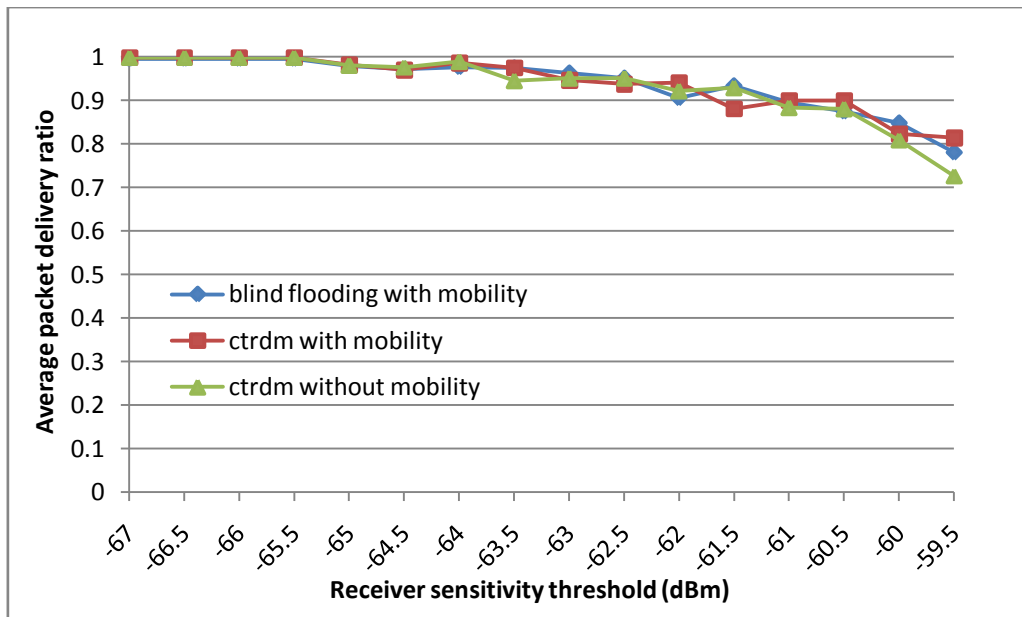


Figure 7.20 Performance of CTRDM with and without mobility and the blind flooding with mobility

7.6 Random Topology Route Discovery Method

Results on the simulation of RTRDM were plotted as shown in Figure 7.21 and 7.22. The graph of Figure 7.21 compares the number of forward nodes required for a complete broadcast of a control packet in RTRDM and in the blind flooding method at a given transmission range. The vertical axis of the graph in Figure 7.21 is explained by Equation (7.5). The graph of Figure 7.21 shows that RTRDM uses fewer forward nodes than the blind flooding method and that the number of forward nodes used by RTRDM is almost independent of the wireless connectivity level of a network.

The results of Figure 7.21 verify that passive clustering is capable of using fewer forward nodes than the blind flooding method for the propagation of Join-Query packets. The hope in this thesis is that the number of forward nodes used by RTRDM shown in Figure 7.21 could have been reduced further if a full implementation of the passive clustering method as proposed in [100] was used. Nevertheless, the results fulfil the first aim of the simulation on RTRDM.

Shown in the graph of Figure 7.22 is a performance comparison in terms of the packet delivery ratio between RTRDM and the blind flooding method. Between -63.5dBm and -59.5dBm in Figure 7.22 RTRDM has a slightly less packet delivery ratio than the blind

flooding method at a given transmission range (or the receiver sensitivity threshold) possibly due to a small deficiency in the partial implementation of the passive clustering used in the simulation which made the implementation fail to guarantee complete broadcasts of Join-Query packets. The implementation was able to yield the packet delivery ratio which was the same as that of the blind flooding method between -67dBm and -63.5dBm.

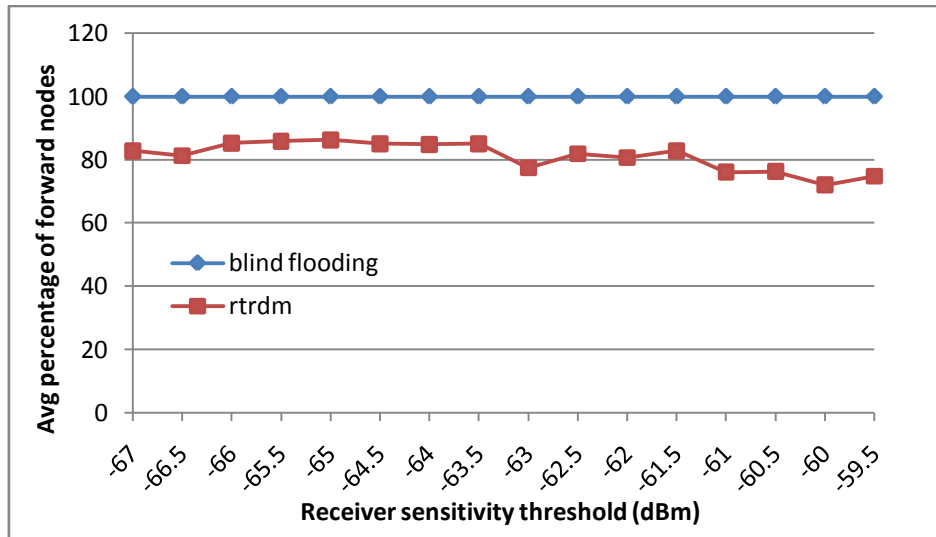


Figure 7.21 Forward nodes in RTRDM and the blind flooding method

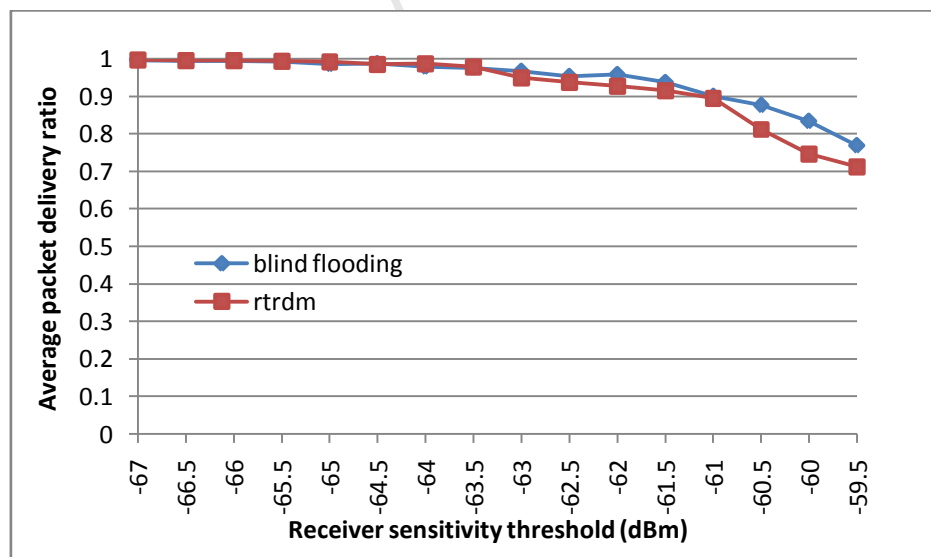


Figure 7.22 Performance in RTRDM and the blind flooding method

The conclusion from the results of Figure 7.22 is that passive clustering is capable of using fewer forward nodes than the blind flooding method with very little compromise on the

packet delivery ratio. Again here, the hope is that a full implementation could have possibly improved the packet delivery ratio. Nevertheless, the second aim of the simulation on RTRDM is fulfilled by the results of Figure 7.22.

The observation in this thesis partly based on the results of RTRDM in Figure 7.21 and 7.22 is that passive clustering needs further research in order for it to provide efficiency and reliability of using forward nodes comparable to that of CTRDM and MTRDM. This observation is made in view of the fact that even the passive clustering method in [100] cannot guarantee a complete broadcast of a packet in a network. Nevertheless, the conclusion in this thesis is that a route discovery method based on passive clustering is capable of providing a more efficient use of forward nodes in a mobile environment than the blind flooding method used by ODMRP.

7.7 Topology-Aware Polymorphic Multicast Routing Protocol

7.7.1 Convergence of TAPMRP after a controller node failure

The first set of data collected from the simulation on the convergence of the election process of a new controller node were plotted as shown in Figure 7.23. The graph of Figure 7.23 shows a convergence curve of new controller announcements following the detection of the absence of a controller node in a network of 55 nodes. The horizontal axis of the graph of Figure 7.23 indicates the times when the new controller nodes almost collectively announced that they were controllers (refer to the methodology chapter for details). Each of the average number of controller nodes that can be obtained from the vertical axis of the graph of Figure 7.23 was determined based on the ten runs of the simulation with each run using a different seed value for the generation of random numbers used by GloMoSim.

Soon after detecting the disappearance of a controller node all the other nodes (i.e. 54 nodes) in the network announced almost at the same time that they were new controller nodes as shown at 750 seconds of the simulation time (the horizontal axis of the graph of Figure 7.23). As the simulation time progressed, the number of new controller announcements was decreasing such that at the simulation time of 1470 seconds there were only 1.4 new controller nodes on average. Ideally, after triggering the 54 new controller announcements at 750 seconds, TAPMRP was supposed to have eliminated 53 new controller announcements by the beginning of the next controller hello interval, which is at 870 seconds in Figure 7.23. Thus, the election process of a new controller node was supposed to converge within a single

controller hello interval. This was not the case in the simulation possibly due to packet collisions which prevented each node from receiving announcements from all the other nodes in the network before sending its second announcement at 870 seconds.

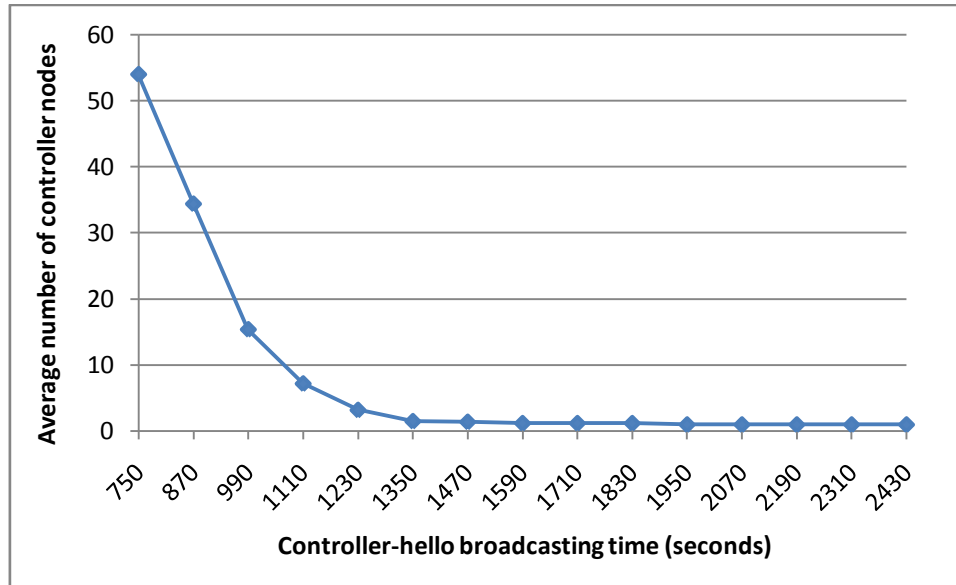


Figure 7.23 Convergence curve of new controller node announcements in TAPMRP

Even though the curve of Figure 7.23 appears to imply that between 750 seconds and 1350 seconds the number of new controller announcements appears to be reduced by half at the next controller-hello broadcasting time, the actual simulation instances were highly unpredictable in terms of the expected number of new controller announcements at the beginning of the next controller hello interval. For example, in one simulation run the number of the new controller announcements dropped from 54 at 750 seconds to 13 at 870 seconds while in another simulation run the number decreased slightly from 54 at 750 seconds to 49 at 870 seconds. This observed unpredictable behaviour is likely due to the probabilistic nature of the wireless medium. Note that the exact cause of the behaviour is not known, and therefore it is one of the limitations of this research.

The results of Figure 7.23 were obtained from a receiver sensitivity threshold setting of -64dBm. The number of controller hello intervals in Figure 7.23 which had multiple new controller announcements before having a single new controller announcement is 5, counting from 870 seconds up to 1350 seconds. In fact, the actual average number of such controller hello intervals was 5.4. The graph of Figure 7.24 shows the average number of similar

controller hello intervals over a range of receiver sensitivity threshold values. Note that the number of controller hello intervals that can be obtained from the vertical axis of the graph of Figure 7.24 represents the speed of convergence of the election process of a single new controller node. For instance, at -65dBm the average number of controller hello intervals with multiple announcements was 7.8 which implies that the election process took a longer time to terminate (or converge) than at -64dBm where the average number of such controller hello intervals was 5.4. The results of Figure 7.24 show that generally convergence of the new controller election process decreases slightly (i.e. taking a longer time) with the decrease in the connectivity level of the network, which according to Figure 7.24 connectivity decreases in the direction of increasing the receiver sensitivity values (i.e. from -67dBm towards -59.5dBm).

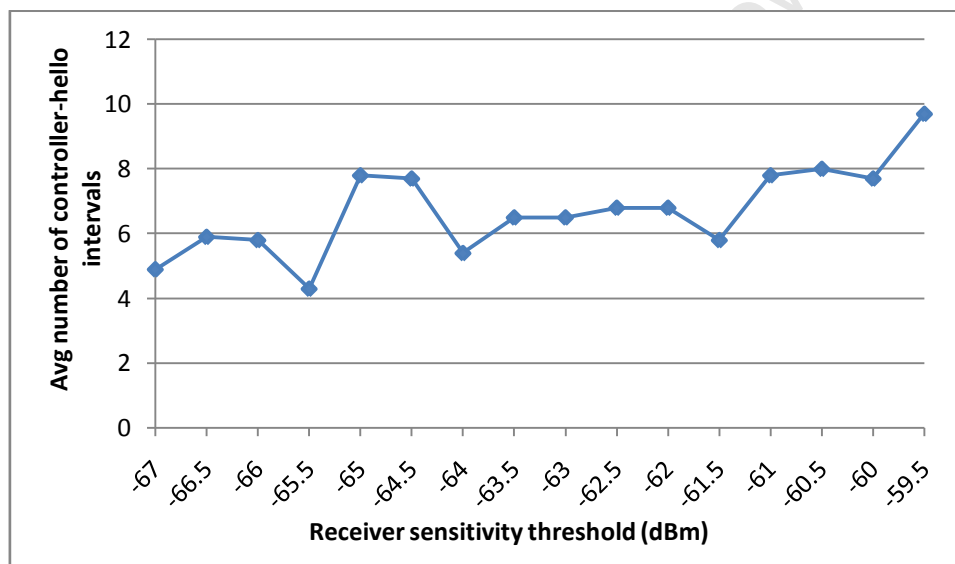


Figure 7.24 Variation of new controller election convergence with the transmission range

Figure 7.25 shows the variation of the convergence of the new controller election process with the network size. The results of the graph of Figure 7.25 are almost the same as those of Figure 7.24 in the sense that the new controller election convergence also has a slight decrease (i.e. taking more time) generally as the network size increases. The conclusion from the results of Figure 7.24 and 7.25 is that the convergence of the election process of a single new controller node has little dependence on the connectivity level and the size of a wireless ad hoc network. Nevertheless, the results (in Figure 7.23, 7.24, and 7.25) show that convergence of the election process takes place within 10 controller hello intervals, which

according to the observation in this research is a reasonably short period of time for practical use.

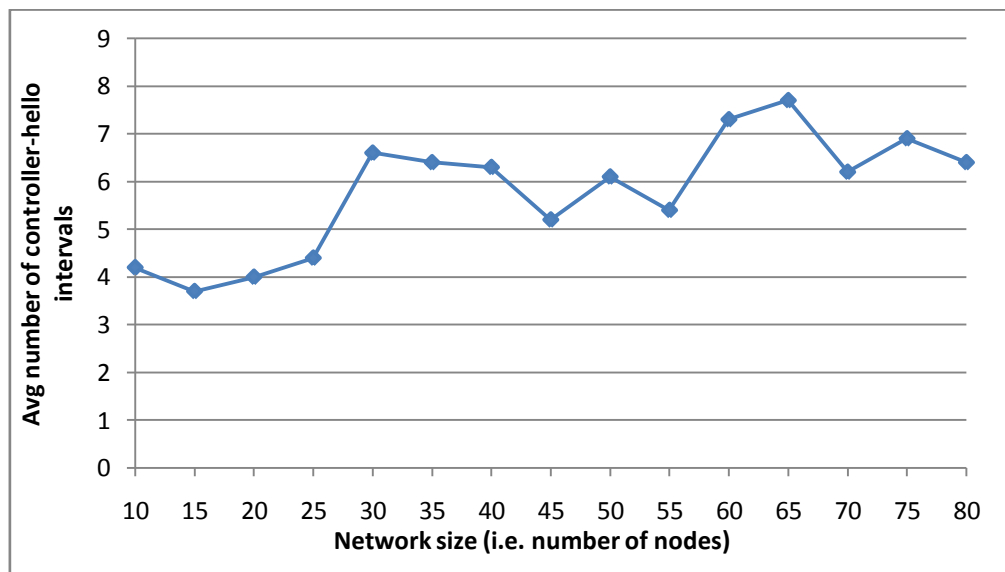


Figure 7.25 Variation of the new controller election convergence with the network size

Performance in terms of the packet delivery ratio of TAPMRP during the time of electing a new controller node is shown in the graph of Figure 7.26. The results of Figure 7.26 show that there is generally a small decrease in the performance of TAPMRP during a new controller election period compared to the period of having one controller node at a given receiver sensitivity setting. The observed decrease in performance is possibly due to rapid changes of route discovery methods during packet transmission.

During an election process competing controllers usually announce incorrect route-discovery switch decisions and different decisions each time they announce because the decisions are based on partial topology information gathered within a short period of time. Normally, nodes in a network were programmed to change their route discovery methods, upon receiving a different route discovery method from a controller, at least after two consecutive controller hello intervals. During the time of electing a new controller it was most likely that nodes changed their route discovery methods after every two consecutive controller hello intervals since the probability of receiving a different route discovery method after two consecutive controller hello intervals from one of the competing controllers was very high. Nevertheless, this research considers that the performance of TAPMRP as shown in Figure 7.26 during a new controller election process is good enough for practical use.

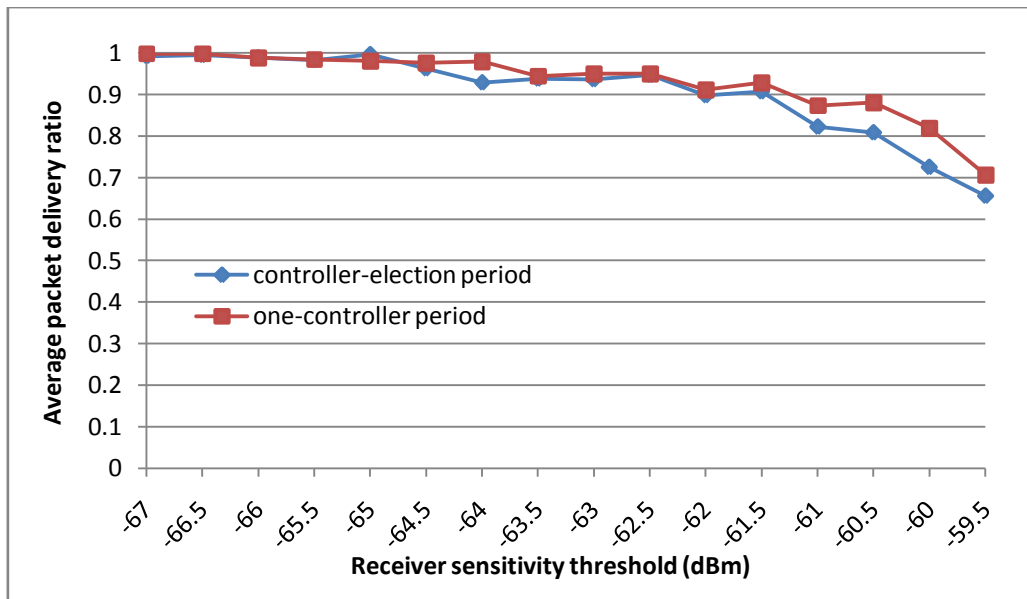


Figure 7.26 Performance of TAPMRP during election of a new controller node

The simulation results (from Figure 7.23 up to 7.26) on the convergence of the election process of a new controller node have shown that the process terminates within a reasonably short period time for practical use, and that there is a very negative effect on the delivery of multicast packets during the election process. Therefore, the proposed criterion of electing a new controller node based on the “First Declare First Wins” rule used in TAPMRP is capable of producing the intended results in practice.

7.7.2 Seamless route discovery method switch

The simulation test on the performance of TAPMRP during the time of switching from one route discovery method to another yielded the results presented in Table 7.2. The first column of Table 7.2 presents the average packet delivery ratios for the route discovery method switches that were considered in the simulation. Shown in column 2 of Table 7.2 are the average packet delivery ratios obtained when using only one route discovery method. The results of column 2 are presented for comparison purposes. The third column of Table 7.2 shows the differences between corresponding entries of column 1 and 2.

The results of Table 7.2 show that generally the packet delivery ratios obtained during switching from one route discovery method to another (column 1) are slightly less in value than the results obtained when using one route discovery method (column 2). This is possibly due to some nodes failing to receive the announcement of a new route discovery method on

time from the controller node. Failure to receive a new route discovery method announcement on time implies that the affected node is likely to use a route discovery method which is different from that used by other nodes for propagating a Join-Query packet. Hence, the resulting multicast tree rooted at the source for delivery of multicast data packets may not reach all the nodes in the network during the switch-over period, resulting in some nodes missing the multicast packets.

Table 7.2 Performance of TAPMRP during changing route discovery methods

Avg packet delivery ratio (two rdms)	Avg packet delivery ratio (one rdm)	Difference
mtrdm-to-ctrdm: 0.885859	ctrdm only: 0.882090	-0.003769
rtrdm-to-ctrdm: 0.874451	ctrdm only: 0.887524	0.013073
mtrdm-to-rtrdm: 0.897566	rtrdm only: 0.916611	0.019045
ctrdm-to-rtrdm: 0.918477	rtrdm only: 0.909836	-0.008641
ctrdm-to-mtrdm: 0.868140	mtrdm only: 0.887055	0.018915
rtrdm-to-mtrdm: 0.886083	mtrdm only: 0.897430	0.011347

The fact that the results obtained during the switch-over period are very close to the results obtained when using one route discovery method and sometimes better than those of one route discovery method, for example mtrdm-to-ctrdm in Table 7.2, it means that the effect of the switch-over on the construction of a multicast tree is generally very small. The conclusion in this thesis is that the switching from one route discovery method to another has almost no negative effect on the performance of TAPMRP, and therefore TAPMRP has put in place the right mechanisms for coping with switching from one route discovery method to another (refer to chapter 5).

7.7.3 Probability of making correct switch decisions

The first set of results on the simulations that were aimed at determining the probability of making correct switch decisions on route discovery methods were plotted as shown in the graph of Figure 7.27. The graph of Figure 7.27 shows the probability with which TAPMRP can switch to MTRDM when the ELC-matrix topology is present. Note that the probability shown in Figure 7.27 is also regarded as the probability of detecting the presence of the ELC-matrix topology. The results of Figure 7.27 indicate that probability of detecting the presence of the ELC-matrix topology is very high regardless of the connectivity level of the network that was considered in the simulation. Between -47dBm and -45dBm in Figure 7.27, the

probability is less than 1.0 possibly due to topology information from some nodes failing to reach the controller node on time.

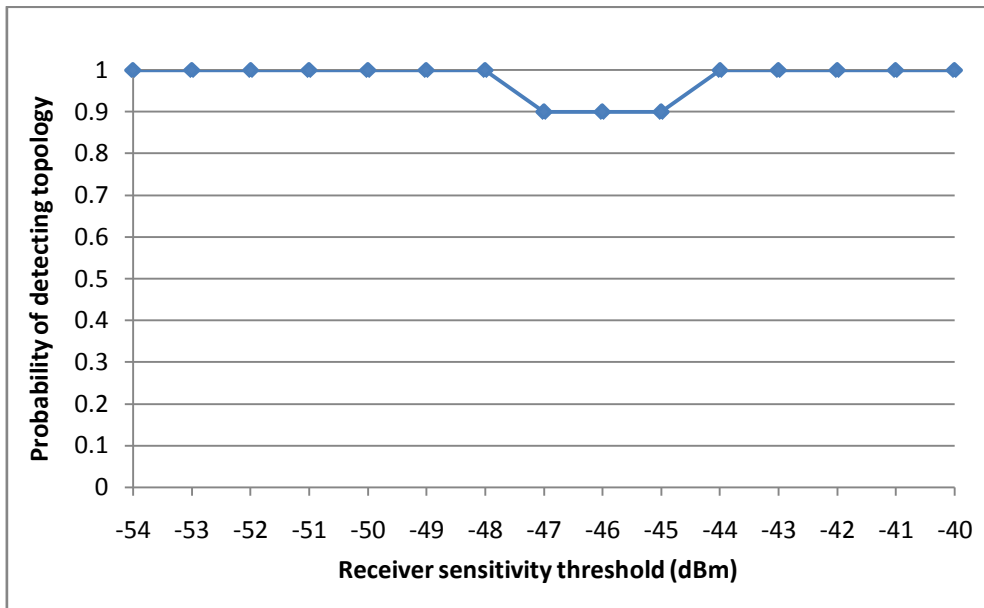


Figure 7.27 Probability of detecting the ELC-matrix topology by TAPMRP

Generally the probability of detecting an ELC topology varies with the level of connectivity (refer to Figure 7.28 and 7.29) because the standard deviation of neighbours of a node that is used to detect the topologies generally depends on the level of connectivity of a network. By having the probability of detecting the ELC-matrix topology as shown in Figure 7.27 that is independent of the connectivity level that was considered in the simulation means that the derivation of Equation (7.2) was able to take care of the standard deviation values that can arise from the ELC-matrix network with almost any level of connectivity including low connectivity.

Note that while in Figure 7.27 low connectivity can be considered as starting perhaps from -44dBm (with 24.4 as the average number of neighbours per node) up to -40 (with 10.7 as the average number of neighbours per node) and beyond, the fact is that low connectivity of the ELC-matrix topology and that of any other topology of a MANET can occur at any receiver sensitivity threshold value depending on the average distance between adjacent nodes in a network. The simulated network that yielded the results of Figure 7.27 had 1.5m as the average distance between adjacent nodes and the network area was 20m × 20m. Increasing the network area with a corresponding increase in the distance between adjacent nodes can

yield low connectivity at a receiver sensitivity value which can be less than any of the values appearing in Figure 7.27. For example, low connectivity can occur at -60dBm or -80dBm. Hence, the probability of detecting the ELC-matrix topology is expected to remain high wherever there is low connectivity regardless of the receiver sensitivity threshold values of the devices. Note that where network connectivity is high detection of the ELC-matrix topology will always have a high probability.

The graph of Figure 7.28 shows the probability of switching to CTRDM when the ELC-cluster-based topology is present. The results of Figure 7.28 show that the probability of detecting the presence of the ELC-cluster-based topology depends on the connectivity level of the network. In addition, the results indicate that detection of the ELC-cluster-based topology has a reasonably high probability within a small range of connectivity level which is from a receiver sensitivity threshold value of -63dBm to -61dBm according to the simulation that yielded the results of Figure 7.28. This is not a surprise because according to Figure 7.4, which shows the demarcation lines between ELC-MANET topologies, the ELC-cluster-based topology is represented by a relatively small range of standard deviation values of neighbours of nodes at any given network size compared to the range of standard deviation values that represent the other two ELC-MANET topologies.

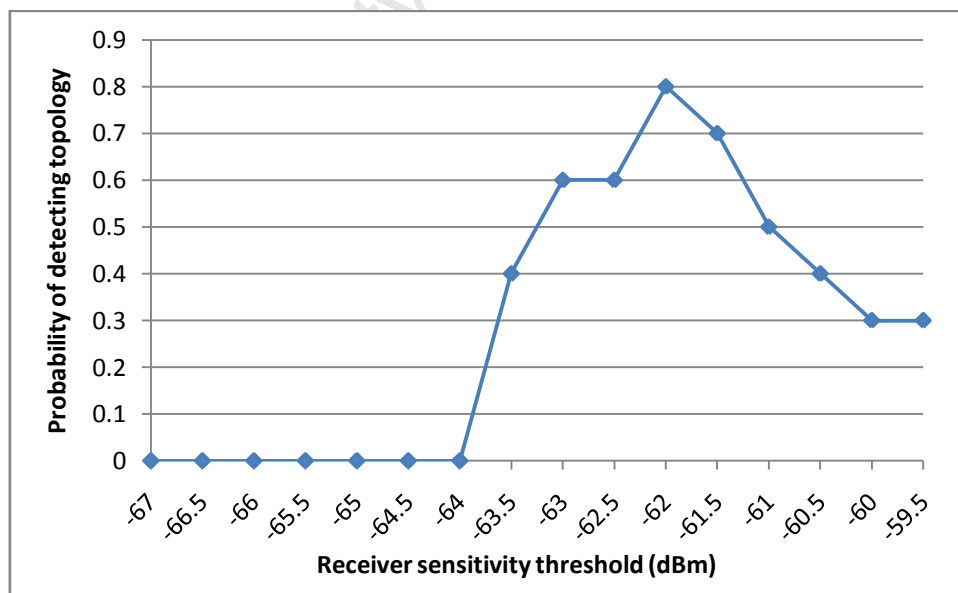


Figure 7.28 Probability of detecting the ELC-cluster-based topology by TAPMRP

Between -67dBm and -64dBm in Figure 7.28, the connectivity level of the simulated network was too high for the ELC-cluster-based topology to be detected in all the ten simulation runs at each receiver sensitivity setting. Failure by TAPMRP to detect the presence of the ELC-cluster-based topology when the connectivity level is very high works to the advantage of TAPMRP because it switches to MTRDM which makes more efficient use of network resources than CTRDM which require clustering of nodes. Failure by TAPMRP to have a high probability of detecting the ELC-cluster-based topology when connectivity level is low, from -60.5dBm up to -59.5 and beyond according the results of Figure 7.28, is a real shortcoming that has no advantage to TAPMRP. When connectivity level is low, the ELC-cluster-based topology is easily misinterpreted as the ELC-random topology. This problem arises from the fact that the use of the standard deviation of neighbours of nodes fails to detect the two ELC topologies with a high probability.

Shown in the graph of Figure 7.29 is the probability of switching to RTRDM when the prevailing topology is the ELC-random topology. Between -67dBm and -63dBm in Figure 7.29, the network connectivity was too high for TAPMRP to detect the presence of the ELC-random topology in all the simulation runs. Hence, TAPMRP continued to use MTRDM between -67dBm and -63dBm. In fact, using MTRDM when network connectivity is high works to the advantage of TAPMRP because in such a network situation MTRDM uses fewer forward nodes than RTRDM.

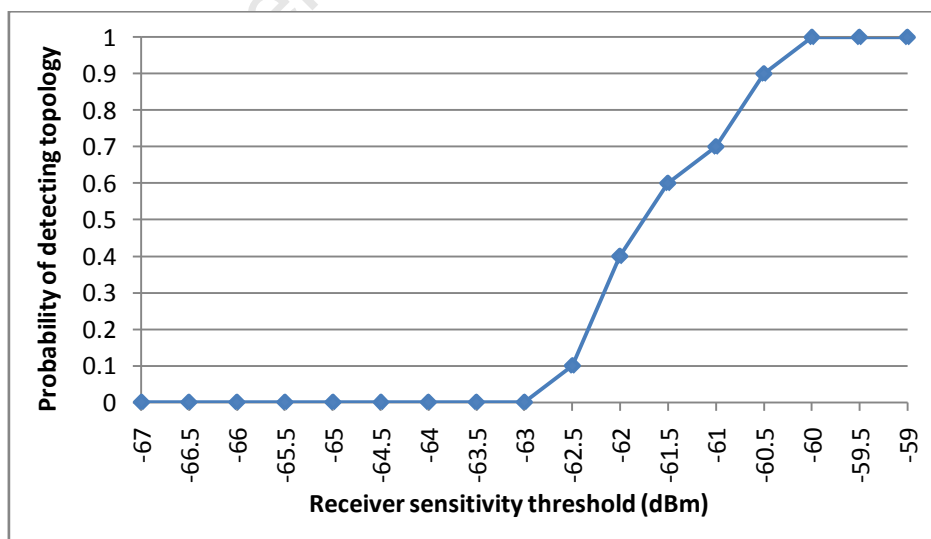


Figure 7.29 Probability of detecting the ELC-random topology by TAPMRP

From -60dBm to -59dBm and beyond in Figure 7.29, the connectivity of the network was relatively low and the probability of detecting the presence of the ELC-random topology was very high. Even though the results of Figure 7.29 appear to suggest that the probability of detecting the ELC-random topology is always 1.0 at low network connectivity, it may not be necessarily the case in practice. However, the probability is still expected to be high when connectivity is low. The extremely high values probability (i.e. 1.0) between -60dBm and -59dBm in Figure 7.29 were mostly due to two factors. First, the probabilities were computed based on only ten simulation runs which were perhaps not enough for sound statistical conclusions. The second factor is that the nature of the random topology used in the simulation may have contributed significantly to the results.

A look at behaviour of the graphs from -60.5dBm to -59.5dBm and beyond in both Figure 7.28 and 7.29 shows that the ELC-random topology has a higher probability of being detected than the ELC-cluster-based topology when network connectivity is relatively low. This observation is in line with demarcation lines of the ELC-MANET topologies shown in Figure 7.4 which assign an infinite range of standard deviation values of neighbours of nodes to the ELC-random topology and only a small finite range of the standard deviation values to the ELC-cluster-based topology.

The overall conclusion from the simulation results on the probability of making correct switch decisions is that out of the three ELC-MANET topologies, the ELC-matrix topology has the highest probability of being detected. The detection of the ELC-matrix topology has a high probability almost at any network connectivity level. The ELC-random topology has a higher probability of being detected than the ELC-cluster-based topology. Hence, detection of the ELC-cluster-based topology has the least probability. When network connectivity is very high TAPMRP almost always interprets any prevailing ELC-topology as the ELC-matrix topology and therefore it switches to MTRDM. This works to the advantage of TAPMRP because MTRDM makes more efficient use of network resources than the other two route discovery methods when connectivity is very high.

7.7.4 Benefit analysis of using TAPMRP

The analysis begins by observing that TAPMRP is composed of three route discovery methods which are designed to operate one at a time. Hence, the demand for network resources by TAPMRP is at its peak when it is using a route discovery method that uses more

network resources than the other two route discovery methods. This research identifies CTRDM as the route discovery method that demands more network resources than the other two methods. Therefore, the benefit analysis of using TAPMRP is based on CTRDM. The usage of network resources that is under consideration is the number of transmissions of control packets performed by all the nodes in a network of 55 nodes. The network size of 55 nodes is chosen because most of the simulations of this research made use of that network size. The analysis considers the transmissions performed in one controller hello interval. For purposes of comparison, the analysis later considers similar transmissions performed by ODMRP in a period of time equivalent to one controller hello interval.

Assuming that 1) we have one source node that continuously sends CBR traffic to the rest of the nodes in a network of 55 nodes for a maximum duration of one controller hello interval that has a period of 120 seconds of GloMoSim clock, 2) there are no acknowledgment transmissions, and that 3) there are no retransmissions resulting from unsuccessful delivery of control packets, the following transmissions of control packets are performed by TAPMRP when using CTRDM:

- (1) Controller hello message: 55 transmissions since blind flooding is used.
- (2) Controller hello reply: 54×54 ; a worst case scenario in which every node in a network participates in propagating the reply to the controller node.
- (3) Clustering signal: 55 transmissions since each node transmits once; a node with the highest weight announces once that it is a cluster-head and each node in its neighbourhood sends only one reply.
- (4) Neighbourhood hello message: $4 \times 55 = 220$ transmissions; a hello message is sent after every 30 seconds and has only one-hop transmission.
- (5) Join-Query packet: $40 \times 28 = 1120$ transmissions; a Join-Query packet is sent after every 3 seconds and therefore 40 Join-Queries are sent in 120 seconds; 28 comes from the assumption that CTRDM uses only half of the nodes in a network as forward nodes.

The total number of transmissions of control packets from one source node in a period of 120 seconds is 4366. Now, if we consider a situation of 10 source nodes, for example, we realise that the number of 4366 transmissions that we associate with one source node is composed of a portion that depends on the number of sources in a network and another portion that is independent of the number of sources in the 120 second period that we have considered. In terms of the two mentioned portions, the 4366 transmissions are obtained as follows: 3246

transmissions (fixed component) + 1120 transmissions (variable component) = 4366 transmissions. The fixed component which is independent of the number of sources is made up of the control transmissions from (1) up to (4) outlined in the above paragraph, while the variable component which depends of the number of sources comes from (5) only which is on the Join-Query packets. Hence, a situation of 10 source nodes, all sending CBR traffic during the 120 second period, we have 10×1120 transmissions + 3246 transmissions (fixed) = 14446 transmissions.

We now consider ODMRP. The Join-Query packet, which also contains data, sent by ODMRP is a control packet just like that of TAPMRP because, among other reasons, it reaches every node in a network regardless of whether the node is a multicast receiver or not. In a scenario of one source node that continuously sends CBR traffic for a maximum duration of 120 seconds there are 2200 (i.e. 40×55) transmissions of the Join-Query packet resulting from the blind flooding method. In a situation of having 10 source nodes such that all of them are sending CBR traffic within the 120 second period under consideration, we have a total of 22000 transmissions of control packets.

A comparison between TAPMRP and ODMRP on the number of transmissions of control packets shows that if we consider only one source node, TAPMRP has more transmissions than ODMRP. This is due to the fixed component of 3246 transmissions resulting from the proactive part of TAPMRP. As the number of sources in a network that simultaneously send traffic increases, the effect of the fixed component of 3246 transmissions becomes smaller and smaller. Hence, in a situation of 10 sources considered in this analysis, TAPMRP (with 14446 transmissions) has less control transmissions than ODMRP (with 22000 transmissions). The results of this analysis are summarised graphically in Figure 7.30. According to Figure 7.30, TAPMRP begins to have fewer transmissions than ODMRP when the number of simultaneous sources is more than three. Considering the ELC-MANET environment for which TAPMRP is developed is highly interactive, it is likely to have more than three simultaneous sources in such an environment. Hence, TAPMRP is likely to be useful almost all the time in the ELC-MANET environment.

An algebraic summary of the analysis is given by the following expressions. For a network of size n with only one source, the number of transmissions of control packets in TAPMRP is given by $(n^2 + 4n + 1) + 20n$, where n is 1, 2, 3, etc. In the expression, $(n^2 + 4n + 1)$ is the fixed component (with respect to the number of sources) which yields 3246 transmissions in

the case of 55 nodes, while $20n$ is the variable component which yields 1100 transmissions (note that in the example above, 1120 transmissions come from the rounding up of the half of 55). The number of transmissions of control packets in ODMRP is given by $40n$. If we let m be the number of simultaneous sources, then the expression for the transmissions of control packets of TAPMRP in the period of 120 seconds becomes $(n^2 + 4n + 1) + 20mn$, while the corresponding expression for ODMRP becomes $40mn$.

The conclusion from the benefit analysis of TAPMRP is that TAPMRP is more efficient than ODMRP in network resource utilisation when a network has a number of simultaneous multicast sources rather than just one source. For a network of 55 nodes, a minimum of only four simultaneous sources are required for TAPMRP to begin to be more efficient. For networks of less than 55 nodes, it is possible to have a minimum of less than four simultaneous sources for TAPMRP to begin to be more efficient. Note that TAPMRP can be much more efficient than ODMRP when MTRDM is used instead of CTRDM which is used in this analysis. When MTRDM is used, the variable component of the transmissions of control packets of TAPMRP considered in this analysis is almost zero especially in situations of high network connectivity.

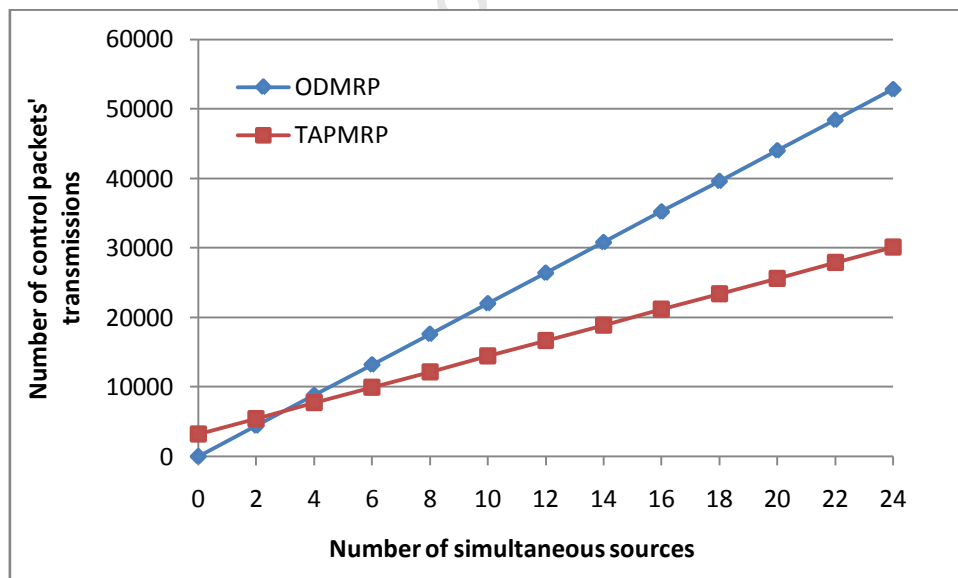


Figure 7.30 Number of transmissions of control packets in ODMRP and TAPMRP in a network of 55 nodes

7.8 Discussion on the Overall Results

The results of the simulations conducted in this research which are presented in this chapter confirm that the proposed design of having three route discovery methods in one MANET multicast routing protocol enhances the efficiency and reliability of the routing protocol when operating in the ELC-MANET. If only MTRDM is used in all the three phases of the ELC-MANET, it is likely to be very efficient in terms of minimising the number of forward nodes when operating in the ELC-matrix topology as evidenced by the results of Figure 7.7. From the results of Figure 7.9, it can be deduced that MTRDM is also likely to operate reliably in terms of minimising packet losses in the ELC-matrix topology environment due the presence of high connectivity in that environment. The results of Figure 7.9 indicate that MTRDM performs poorly in terms of the packet delivery ratio in the presence of mobility and therefore it is likely that MTRDM can perform poorly in the ELC-random topology where mobility is relatively high. MTRDM is also likely to perform poorly in the ELC-cluster-based topology due to the combined effect of mobility and low connectivity. The poor performance of MTRDM in the presence of mobility is due to lack of an advanced mechanism for adapting to mobility.

Using solely CTRDM in all the three phases of the ELC-MANET is likely to optimise both efficiency and reliability when operating in the ELC-cluster-based topology only as confirmed by the results of Figure 7.17 up to Figure 7.20. Operating CTRDM in the ELC-matrix topology is likely to be highly inefficient due to the unnecessary creation of logical clusters. The ELC-matrix topology environment is characterised by high connectivity in which most transmissions are accomplished through one or two hops. Hence, in such an environment, CTRDM is likely to use more resources through the creation of logical clusters than MTRDM to achieve the same performance. CTRDM's mechanism for adapting to mobility provided by Equation (4.4) is not advanced enough to deal with the mobility level expected in the ELC-random topology, and therefore CTRDM is likely to perform less reliably (i.e. having more packet losses) than RTRDM in the ELC-random topology.

If RTRDM is used for all the three phases of the ELC-MANET, it is likely to optimise both efficiency and reliability in the ELC-random topology only. A comparison of the results of Figure 7.7 and 7.21 shows that RTRDM is likely to be less efficient than MTRDM in the ELC-matrix topology as evidenced by the fact that the number of forward nodes required by RTRDM for a complete broadcast of a control packet (shown in Figure 7.21) is constantly

high at any connectivity level. RTRDM is also likely to be less efficient by CTRDM in the ELC-cluster-based topology as confirmed by a comparison of the results of Figure 7.19 and 7.21. The inefficiency of RTRDM is due to the fact that the passive clustering technique used by RTRDM is less effective in reducing forward nodes than the pruning techniques used by MTRDM and CTRDM.

Note that the performance of MTRDM, CTRDM and RTRDM are on a probabilistic basis. MTRDM has over 90% probability of being deployed at the right time as evidenced by the results of Figure 7.27. The deployment probability of MTRDM is higher than that of CTRDM and RTRDM. Figure 7.28 and 7.29 present the probabilities of the right time deployment of CTRDM and RTRDM respectively.

By having MTRDM, CTRDM and RTRDM in one routing protocol (i.e. TAPMRP) it means that when the routing protocol operates in the ELC-MANET, it is likely to optimise both efficiency and reliability at any instant. Hence, TAPMRP is capable of optimising performance in the ELC-MANET due to the design of having three route discovery methods in one multicast routing protocol.

The simulation results are based on three important factors, namely the geographical network areas of the simulated networks, the distances between the nodes in the simulated networks and the receiver sensitivity threshold settings used in the simulations. As mentioned in the introduction section of the methodology chapter, the simulated network areas of $20\text{m} \times 20\text{m}$ for the ELC-matrix topology, $80\text{m} \times 80\text{m}$ for the ELC-cluster-based topology, and $100\text{m} \times 100\text{m}$ for the ELC-random topology are considered in this thesis to be network areas that can arise on average in the ELC networking environment. This means that the presented simulation results represent the envisaged average performance of the ELC-MANET phases in practice with respect to the simulated coverage areas of the three ELC-MANET phases.

The factor of the network area upon which the simulation results depend is determined by the distances between nodes. It was observed that it is difficult to have a well-defined basis that can be used to determine the average distance between two adjacent nodes in the ELC-MANET. The simulations made use of the average distance of 1.5m between adjacent devices used by students who are seated. This average distance of 1.5m was used in ELC-matrix topology and in the ELC-cluster-based topology within clusters. Distances between clusters were guided by the need to have a uniform distribution of clusters within a simulated

area. Simulations on the ELC-random topology mainly used an in-built function of GloMoSim which determined random topologies during running time of a simulation. The simulations of all the three ELC-MANET topologies are considered to have used distances between adjacent nodes that occur on average in the ELC environment. Hence, the simulation results represent the envisaged average performance of the ELC-MANET phases in practice with respect to the distances between adjacent nodes.

While the simulation results represent the envisaged average performance with respect to the envisaged average network areas and envisaged average distances between adjacent nodes, they may not represent the envisaged average performance with respect to the average receiver sensitivity threshold value of the devices in the ELC-MANET. The ranges of the receiver sensitivity threshold settings in the simulations are from -54dBm to -36dBm for the ELC-matrix topology and from -67dBm to -59.5dBm for both the ELC-cluster-based topology and the ELC-random topology. These receiver sensitivity ranges were imposed on the simulations by a combination of the sizes of the simulated network areas and the distances between the nodes without regard to whether they represent typical values in practice or not.

The special and the most useful characteristic of the ranges of the receiver sensitivity threshold values used in the simulations is that they cover a whole of network connectivity from the highest to the lowest. The highest connectivity level is herein understood as a connectivity level in which every node has any other node in a network as its one-hop neighbour while the lowest connectivity is understood as the level of connectivity just before a network is disconnected. Hence, the simulation results apply to any connectivity level that may result from any combination of a receiver sensitivity threshold value and distances between nodes (refer to 2nd paragraph of page 127 for details on how the receiver sensitivity threshold values used in the research were selected).

The ELC-MANET is envisaged to be served by devices of various types and it is likely that such devices can have a wide range of receiver sensitivity threshold values. Assuming that most of the devices in the ELC-MANET are laptop computers then it is likely that, on average, the devices in the ELC-MANET can have receiver sensitivity threshold values that are lower than -67dBm. The interpretation of the simulation results with respect to the average network areas, the average distances between nodes, and the average receiver

sensitivity values of the devices reveals that the envisaged average performance of TAPMRP in practice in the ELC-MANET is similar to the one obtained at the lowest endpoints of the receiver sensitivity settings used in the simulations (i.e. -67dBm for the ELC-cluster-based topology and ELC-random topology, and -54dBm for the ELC-matrix topology).

By having the envisaged performance of the ELC-MANET in practice which is the same as the simulation results obtained at the lowest endpoints of the receiver sensitivity settings used in the simulations, it means that TAPMRP is expected to use MTRDM on average. Since network connectivity is expected to be high on average in the ELC-MANET, TAPMRP is likely to interpret all the three phases of the ELC-MANET as being the ELC-matrix topology. There is no problem in using MTRDM in most instances in the ELC-MANET because MTRDM is the most efficient of three route discovery methods of TAPMRP in scenarios of high connectivity. In the less frequent occasions when connectivity is not very high possibly due to an increase in the network area beyond the average values considered in this simulation, TAPMRP may correctly detect the prevailing topology and use a corresponding route discovery method. Hence, CTRDM and RTRDM are expected to be used less frequently than MTRDM. There is no problem in using CTRDM and RTRDM less frequently than MTRDM; what is important is that at any instant TAPMRP should use the most efficient route discovery method.

7.9 Summary

Presented in this chapter are the results of the simulations that were conducted in this research with the intention of confirming the existence of the research problem and testing the ideas which were generated in the course of conducting this research. The tested ideas constitute the proposed solution to the research problem. The results on the validation of the research problem indicate that the blind flooding method of propagating control packets generates a lot of redundant retransmissions. Given a network of any size, every node in the network, except for the source of a broadcast being propagated, retransmits the broadcast packet at least once regardless of the connectivity level of the network when the blind flooding method is used. Hence, the blind flooding method leads to wastage of energy and bandwidth.

The computer simulation that was aimed at formulating a mechanism for detecting the three phases of the ELC-MANET based on the standard deviation of neighbours of nodes in a

network yielded two equations: Equation (7.1) and (7.2). Equation (7.1) creates a demarcation line between the ELC-matrix topology and the ELC-cluster-based topology for a MANET of any size, while Equation (7.2) separates the ELC-cluster-based topology from the ELC-random topology. Overlaps along the demarcation lines do exist. Nevertheless, the ELC-matrix topology has a very high probability of being detected, and its detection probability is the highest of the detection probabilities of the three ELC-MANET phases. The ELC-cluster-based topology has the least probability of being detected.

The results on the simulation of MTRDM show that, in general, MTRDM uses many fewer forward nodes for the propagation of control packets than the blind flooding method. In fact, when network connectivity is very high MTRDM requires almost no forward nodes for the Join-Query packets. When connectivity is high, MTRDM performs well regardless of the presence of mobility of nodes. Hence, MTRDM is suitable for the ELC-matrix topology because it is characterised by relatively high network connectivity.

The simulations confirm that CTRDM uses fewer forward nodes for propagation of the Join-Query packet than the blind flooding method. When connectivity is very high, the number of forward nodes used by CTRDM can get as low as zero. The simulation results show that the use of secondary cluster-heads and the node redistribution parameter helps in providing load balancing, which is intended to reduce overloading of few nodes in a network.

The results on RTRDM show that RTRDM is capable of using fewer forward nodes for broadcasting control packets than the blind flooding method. The passive clustering technique used by RTRDM is able to yield good performance, in terms of the packet delivery ratio, in the presence of node mobility. Therefore, RTRDM is suitable for the ELC-random topology which is characterised by relatively high node mobility.

The first simulation on the operations which affect TAPMRP as a whole was on the speed of termination of the process of electing a new controller node. The results show that the “First Declare First Wins” principle employed by TAPMRP for electing a new controller node after detecting the absence of the current controller node is capable of terminating the election process within a period of ten controller hello intervals. The research considers the termination period to be reasonably short for practical use.

The simulation on the design effort of TAPMRP to offer a seamless switch from one route discovery method to another has yielded results which indicate that the switch-over period has an insignificantly small negative effect on the performance of TAPMRP. Hence, TAPMRP has the right mechanisms for coping with switching from one route discovery method to another.

Results on the probability of making correct switch decisions from one route discovery method to another show high correlation with the demarcation lines of the ELC-MANET phases given by Equation (7.1) and (7.2). The results confirm that the ELC-matrix topology has a very high probability of being detected which seems to be independent of the connectivity level of a network. The ELC-matrix topology has the highest probability of being detected followed by the ELC-random topology, while the ELC-cluster-based topology has the least detection probability.

The final topic on TAPMRP is a benefit analysis of using TAPMRP over ODMRP. The analysis reveals that TAPMRP becomes more efficient, in terms of network resource usage, than ODMRP when a network has a number of simultaneous sources of traffic, and not when it has only one source. For a network of 55 nodes, a minimum of only four simultaneous sources is needed for TAPMRP to begin to be more efficient than ODMRP.

The chapter ends with a discussion on the overall results of the simulations. The discussion points out that the simulation results represent the envisaged average performance of TAPMRP in practice mainly with respect to the envisaged average network areas and envisaged average distances between nodes in the three ELC-MANET phases. Inclusion of the envisaged average receiver sensitivity threshold values in the interpretation of the simulation results indicates that the envisaged average performance of TAPMRP in practice is the one given by the lowest endpoints of the ranges of the receiver sensitivity threshold settings used in the simulations. Hence, on average, MTRDM is expected to be used more often than CTRDM and RTRDM.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Introduction

This is the concluding chapter of the thesis. The chapter begins with a summary and conclusions on the entire work conducted in this research which begins from chapter one, which covers the introduction of the thesis, up to chapter 7, which is on the simulation results and how they were analysed.

A section on the major contributions outlines novel ideas and techniques which were developed and tested in the course of conducting the research reported in this thesis. The section on the major contributions is followed by a section on minor contributions. In this thesis, minor contributions are considered to be applied contributions which are based on applying techniques that are already available in the literature. Both the major and minor contributions justify the existence of this thesis.

The chapter includes a section on recommendations for future work. The inclusion of the section is based on the fact that there were some practical limitations which prevented the solution to the research problem to be improved further. Time was a major practical limitation because the scheduled time of this research was not long enough to enable some time-demanding components of the research problem to be investigated comprehensively.

The organisation of this chapter is as follows. Section 8.2 covers a summary and conclusions. Major contributions of the thesis are presented in section 8.3 followed by minor contributions in section 8.4. Section 8.5 presents the limitations of this research. The chapter ends with recommendations for future work presented in section 8.6.

8.2 Summary and Conclusions

The thesis begins with some background information on the application of MANETs to E-Learning covered in chapter one. Chapter one points out that the focus of this research is on how MANET multicasting can efficiently and reliably serve learners in the E-Learning

environment. Some of the MANET multicast services which can be offered to learners include file sharing, video streaming and offline websites. After covering background information on E-Learning, multicasting, and MANETs, the research problem is introduced based on the observation that the existing multicast routing protocols have excessive control overhead which does not make efficient use of the limited battery energy and the limited bandwidth capacity of a MANET. The need to use an adaptive multicast routing protocol that exploits the connectivity patterns of the underlying network for improving efficiency and reliability is also introduced in chapter one as the proposed solution to the research problem.

Chapter two starts with a literature review of some of the existing the work on the application of MANETs to E-Learning. The review shows that most of the current work on the use of MANETs in E-Learning has been on the application layer of the OSI reference model. Chapter two also presents a review of the existing MANET multicast routing protocols, MANET broadcasting protocols, and on MANET clustering techniques. A detailed analysis of ODMRP is also covered in chapter two. Chapter two presents a full definition of the research problem. In summary, the research problem is the broadcast redundancy associated with route discovery operations in the existing MANET multicast routing protocols.

Formulation of the solution to the research problem starts in chapter three. Chapter three begins with the observation that the ELC-MANET has the tendency of transforming itself into three distinct topologies depending on the nature of the students' activities. The three distinct topologies are the ELC-matrix topology, the ELC-cluster-based topology and the ELC-random topology. Each one of these three ELC-MANET phases has its own characteristics which impose unique requirements on a multicast routing protocol for efficient and reliable operation. Next, chapter three presents the requirements of the ideal multicast routing protocol for the ELC-MANET environment. In general, the ideal solution is expected to show efficiency, robustness, scalability, simplicity and adaptability. Chapter three also presents a proposed mechanism for detecting the three phases of ELC-MANET. The mechanism is based on the standard deviation of the number of neighbours of nodes in a network.

Chapter four covers the three ELC-MANET phases in detail. The first to be covered is the ELC-matrix topology. The observation in this research is that the ELC-matrix topology occupies a relatively small geographical area compared to the other two ELC topologies because it is confined to classroom environments. Hence, the ELC-matrix topology is

characterised by high network connectivity. It is also observed that the ELC-matrix topology is characterised by occasional and low mobility of nodes. Matrix Topology Route Discovery Method (MTRDM) is proposed to serve the ELC-Matrix topology. MTRDM is one of the three route discovery methods which constitute the proposed multicast routing protocol for the ELC-MANET called a Topology-Aware Polymorphic Multicast Routing Protocol (TAPMRP). The design of MTRDM is based on a pruning technique of forward nodes of control packets that make use of the greedy set cover algorithm [7] and the partial dominant pruning algorithm [39].

After presenting the ELC-matrix topology, chapter four covers the ELC-cluster-based topology. The ELC-cluster-based topology is characterised by having high connectivity within clusters and low connectivity outside the clusters. Mobility of nodes the ELC-cluster-based topology is occasional and low just like in the environment of the ELC-matrix topology. The proposed route discovery method of TAPMRP to serve the ELC-cluster-based topology is called a Cluster-based Topology Route Discovery Method (CTRDM). CTRDM has an active clustering component which periodically reconstructs logical clusters in the underlying network. In CTRDM, only cluster-heads and a subset of bridge nodes are allowed to forward control packets. Two novel techniques, namely secondary cluster-heads and the node redistribution parameter, are used in CTRDM to provide load balancing to the transmission of both control packets and multicast data packets.

Chapter four also covers the ELC-random topology in detail. The ELC-random topology is characterised by having no definite pattern of connectivity. Another characteristic of the ELC-random topology is that it has a relatively high mobility of nodes compared to the other two ELC-MANET phases. The route discovery method of TAPMRP intended for the ELC-random topology is called a Random Topology Route Discovery Method (RTRDM). RTRDM makes use of passive clustering which allows only cluster-heads and bridge nodes to forward Join-Query packets.

Chapter five covers a functional specification of TAPMRP. TAPMRP makes use of a node called a controller node. The controller node coordinates all network-wide operations affecting a particular multicast group. Periodically, the controller node sends out a controller hello message to the rest of multicast group members. The controller node runs a very important function of TAPMRP called a route-discovery-method switch. The route-discovery-method switch gathers topology information from multicast member nodes and

uses the information to determine the most appropriate route discovery method to be used by the multicast group. The decision to switch from one route discovery method to another is announced by the controller node.

The packet header fields and routing databases used by MTRDM, CTRDM and RTRDM are specified in chapter five. TAPMRP is designed in such a way that its three route discovery methods, namely MTRDM, CTRDM and RTRDM use the same routing databases, namely routing tables, message caches and forward group tables. The operations of MTRDM, CTRDM and RTRDM are also described in chapter five. In general, all the three route discovery methods are characterised by a route request phase and a route reply phase. A route request phase is initiated when a source wants to find a route to multicast receivers. A Join-Query packet is broadcast to all the nodes in a network during the route request phase. During a route reply phase, multicast receivers unicast reply messages back to the source node.

The research methodology is covered in chapter six. Experimental testing of the ideas developed in this research was done by using computer simulations which were conducted in GloMoSim-2.03. Since TAPMRP is developed as an improvement on ODMRP, the ideas were tested by incorporating them in ODMRP which is already implemented in GloMoSim-2.03. The simulations made use of the assumptions provided by GloMoSim-2.03 such as the use of one transmission medium, the use of one transmission range by all nodes, and bidirectional transmission of packets between nodes.

The simulations used geographical network areas of $20\text{m} \times 20\text{m}$ for the ELC-matrix topology, $80\text{m} \times 80\text{m}$ for the ELC-cluster-based topology and $100\text{m} \times 100\text{m}$ for the ELC-random topology. These network areas are considered to be the average areas of network coverage of the three phases of the ELC-MANET. It is possible that in some larger universities or school campuses the ELC-cluster-based topology and the ELC-random topology may extend much further than the average values used in the simulations. For instance, the ELC-random topology can cover an area of more than 500m in diameter.

Chapter seven presents the results of the simulations conducted in this research. The results of the validation tests on the existence of the research problem are the first to be presented in chapter seven. The results indicate that the blind flooding method of broadcasting control packets uses every node in a network, except for the source node of a control packet being propagated, as a forward node regardless of the connectivity characteristics of the network.

This makes the blind flooding method generate broadcast redundancy which wastes bandwidth and energy resources. Next, the result chapter presents two equations which represent demarcation lines that are used to detect the three phases of the ELC-MANET.

The results show that MTRDM and CTRDM require many fewer forward nodes than the blind flooding method for propagation of control packets. RTRDM requires fewer forward nodes than the blind flooding method but not comparable to the level of MTRDM and CTRDM which use many fewer forward nodes. Performance of MTRDM, CTRDM and RTRDM in terms of the packet delivery ratio is comparable to that of the blind flooding method.

The benefit analysis of using TAPMRP over ODMRP shows that TAPMRP is more efficient than ODMRP in terms of network resource usage when a network has a number of simultaneous sources and not when the network has only one source of traffic. For a network of 55 nodes, only a minimum of four simultaneous sources is required for TAPMRP to begin to be more efficient than ODMRP. The ELC-MANET environment is envisaged to be a highly interactive environment which is likely to have a number of simultaneous sources at any instant. Hence, TAPMRP is likely to be very useful in most instances in such an environment.

8.3 Major Contributions

This thesis has seven contributions which are considered to be major contributions. The first of such contributions is the observation that when a MANET is used in the environment of E-Learning, it has the tendency of transforming itself from one distinct topology to another depending on the activities of the learners. The resulting topologies have distinct connectivity characteristics which affect the performance of a multicast routing protocol operating in them. The distinct topologies are also known as phases of the ELC-MANET in this thesis. The thesis has identified three distinct topologies into which the ELC-MANET can transform itself, and these are the ELC-matrix topology, the ELC-cluster-based topology and the ELC-random topology. The understanding in this thesis is that no experimentation is needed to verify the transformation of the ELC-MANET. However, some aspects of the transformation such as length of time in each phase and the time of transition need experimentation but that was not done in this research.

This observation on the transformation of the ELC-MANET is not confined to E-Learning environments only. There are many situations in practice in which transformation of a MANET can be observed. For instance, in conferences participants may at a certain time sit in a regular arrangement in a big room and thereafter they may decide to rearrange themselves into small discussion groups thereby forming a cluster-based topology. This observation on MANET transformation opens a new direction in which MANET researchers will begin to think about other types of topologies which can result from the transformation of a MANET.

The second major contribution is the novel design of using three different route discovery methods, namely MTRDM, CTRDM and RTRDM, in one MANET routing protocol. This design has the capability of optimising both efficiency and reliability of a MANET multicast routing protocol. MANET researchers have combined different aspects of different routing protocols, such as tree-based designs and mesh-based designs, into one protocol in order to optimise quality of service. The comprehensive literature review of this thesis has shown that all such efforts of improving designs of routing protocols have been confined to the use of only one route discovery method in a MANET routing protocol. The use of more than one route discovery method in one MANET routing protocol is therefore a novel contribution of this thesis. This contribution opens a new way of designing MANET routing protocols that takes into consideration the possibility of using more than one route discovery method as a way of enhancing performance.

The novel idea of using the standard deviation of the number of neighbours of nodes in a network as a mechanism for detecting the three identified phases of the ELC-MANET is the third major contribution of this thesis. The standard deviation mechanism is more computationally efficient than a mechanism based on signal strengths and more reliable for indoor operations than a mechanism based on the Global Positioning System (GPS). The simulation results have shown that the standard deviation mechanism has a reasonably high probability of detecting the three topologies of the ELC-MANET.

The fourth major contribution is the introduction of secondary cluster-heads to support a primary cluster-head in a one-hop clustering method of MANETs. When the degree of connectivity is used as the criterion for selecting nodes for the role of cluster-head in a one-hop clustering scheme, there is a possibility of creating one extremely large cluster and possibly few small ones when network connectivity is very high. This can easily lead to

overloading the cluster-head with an extremely large number of member nodes. Secondary cluster-heads provide load balancing to the transmission of both control packets and multicast data packets. Load balancing is intended to prevent rapid depletion of battery energy of cluster-heads. The need to introduce secondary cluster-heads is based on the observations expressed in *theorem 4.1* through *theorem 4.3*.

The formulation of the node redistribution parameter as one of the factors to consider when selecting nodes for the role of cluster-head in a one-hop clustering scheme is the fifth major contribution of this thesis. The node redistribution parameter is computed at every node using neighbourhood information. The effect of the parameter is to create logical clusters that almost match with the underlying physical clusters when network connectivity is not too high. The resulting clusters are more in number than in the case of cluster-head selection based on the degree of connectivity, and therefore there are more cluster-heads available in the network for forwarding of both control packets and multicast data packets. Hence, the node redistribution parameter provides load balancing by providing more forward nodes than those provided by the cluster-head selection criterion based on the degree of connectivity. The node redistribution parameter is given by Equation (4.2).

The sixth major contribution is Equation (4.4) which is used to predict mobility level of a node with respect to that of its neighbours in the next interval of time. This mobility formula (i.e. Equation (4.4)) is used as one of the factors for selecting nodes for the role of cluster-head in this thesis. The use of the mobility formula is based on the reasoning that less mobile nodes are more likely to maintain stable clusters than more mobile nodes. Equation (4.4) is based on the differences in the number of neighbours of a node obtained at successive intervals of time. The formula uses mathematical operations based on the set theory, and is likely to be computationally more efficient than a method based on signal strengths and more reliable than a method based on the Global Positioning System for indoor operations. The validity of the mobility formula has been tested by computer simulations, and its simulation results are presented in subsection 7.5.3.

The use of nodes that have failed to take part in a clustering process as cluster-heads during a route discovery process that makes use of the underlying clustering platform is considered as the seventh major contribution of this thesis. Some reported work in the literature assumes that every node in a network receives every clustering signal from other nodes. This is not true in practice in a wireless medium because control packets can be lost due to collisions.

Other reported work in the literature reply on retransmissions of control signals after a time-out period as a way of ensuring a successful clustering process, but this can easily lead to uncontrollable triggering of control packets. Furthermore, retransmissions of control signals still do not guarantee participation of every node in a clustering process. The approach taken by this thesis of using every node that fails to take part in a clustering process as a full cluster-head ensures that the resulting clustering platform is highly reliable for a successful route discovery process, but of course with some compromise on efficiency of network resource usage. Note that this contribution arises from the implementation of TAPMRP and not from the theoretical component of TAPMRP covered in the thesis.

8.4 Minor or Applied Contributions

There are three contributions which are considered in this thesis to be minor or applied contributions. The first of such contributions is the use of the greedy set cover algorithm [7] and the partial dominant pruning technique [39] in the reduction of forward nodes of control packets during a route discovery process of a MANET multicast routing protocol. The partial dominant pruning technique was primarily developed for a generic MANET broadcasting method and therefore its use in MTRDM is an applied contribution of this thesis. Simulation results have shown that the partial dominant pruning method together with the greedy set cover algorithm are capable of tremendously reducing the forward node set of Join-Query packets during a route discovery process. In fact, when network connectivity is very high, as the case is in the ELC-matrix network environment, the two pruning techniques require almost no forward nodes of broadcast packets; and the resulting packet delivery ratio is excellent regardless of the presence of node mobility.

The second minor contribution comes from the partial application of the cluster-based broadcasting method developed by J. Wu and W. Lou [46] and the application of the theoretical framework for weight-based clustering of MANET nodes proposed by S. Basagni [106] in a route discovery process of a MANET multicast routing protocol. The two cluster-related ideas [46, 106] are used in CTRDM where they have proved to be quite effective. The development of CTRDM by combining the ideas in [46] and [106] can arguably be considered as a novel design which may fall under major contributions of this thesis. CTRDM is capable of using many fewer forward nodes of control packets than the blind flooding method.

The use of the passive clustering technique proposed by Y. Yi et al. [100] in RTRDM is the third minor contribution of this thesis. The main advantage of passive clustering is that it constructs a clustering platform which effectively adapts to the mobility patterns of nodes. Hence, passive clustering is considered to be a suitable broadcasting method of control packets in a route discovery process of a MANET multicast routing protocol operating in the ELC-random topology environment, which is characterised by high mobility of nodes.

8.5 Limitations of the Research

The experimental part of this research which tests the validity of the theoretical concepts of TAPMRP developed in chapter 4 and 5 is based on a number of assumptions. These assumptions are in the form of numerical values of parameters that were used in the simulations. Some of these numerical values were chosen arbitrarily without performing analyses of their optimal performance. Examples of the numerical values that were chosen almost arbitrarily are the maximum number of nodes per cluster (i.e. 15), the neighbour hello interval (i.e. 30 seconds), and the controller-hello interval (i.e. 120 seconds).

Not all the numerical values used in the research were chosen purely arbitrarily. For example, the ranges of the receiver sensitivity threshold settings were imposed on the simulations by the geographical network areas and the distances between the nodes (refer to section 6.1 for details). Another example is the choice of 1.5m/s as the maximum walking speed of students in the ELC-environment (covered in chapter 6). This speed (i.e. 1.5m/s) was not chosen purely arbitrary since intuitively any speed unreasonably higher than that, say 2 times 1.5m/s, is obviously not walking but running.

Note that even though the numerical settings were mainly chosen arbitrarily, they were able to fulfil the objectives of this research. Nevertheless, the absence of the optimal performance analyses of these numerical values implies that the values were used as assumptions. This thesis considers such assumptions as the limitations of this research which require future investigation.

8.6 Recommendations for Future Work

The thesis presents four areas that require future investigations. Where possible, attempts have been made to provide suggestions of how a particular area can be investigated further.

One of such areas is the probability of detecting the presence of the ELC-cluster-based topology. While the ELC-matrix topology and the ELC-random topology can be detected with reasonably high probabilities, the detection probability of the ELC-cluster-based topology is not as high as the probabilities of the other two ELC-MANET topologies. Future investigations on improving the detection probability of the ELC-cluster-based topology may consider finding a mechanism which can reinforce (or replace) the standard deviation mechanism proposed in this thesis. For instance, a component based on the Global Positioning System can be added to detect the presence of at least one physical network cluster located where such a positioning system can work effectively. The accurate detection of at least one physical network cluster can be used alongside the standard deviation mechanism to improve the accuracy of detecting the presence of the ELC-cluster-based topology.

The second area that requires future research is the possibility of including a mechanism for coping with node mobility in the operations of the greedy set cover algorithm [7] and the partial dominant pruning technique [39] used by MTRDM. The simulation results have shown that MTRDM is the most efficient of the three route discovery methods of TAPMRP in terms of network resource usage, but its excellent performance is only confined to environments of high network connectivity partly due to the fact that node mobility has almost no negative effect on the performance of a MANET multicast routing protocol operating in such an environment. It has been shown further by the simulation results than MTRDM begins to perform poorly in an environment with a combination of low network connectivity and presence of node mobility. It is possible to use MTRDM in the ELC-cluster-based topology and the ELC-random topology if the pruning mechanisms of forward nodes in [7] and [39] are equipped with a mechanism for coping with mobility.

The passive clustering method recommended for RTRDM is the third area which requires further research. The simulation results show that generally RTRDM requires more forward nodes of control packets than MTRDM and CTRDM. Obviously, there is a need to reduce the number of forward nodes required by RTRDM so as to make RTRDM as efficient as MTRDM and CTRDM. The passive clustering technique [100] adopted in this thesis fails to guarantee a complete broadcast of a control packet that can cover every node in a network. This observation is also made in [100] by the owners of the technique. The failure to guarantee a complete network broadcast operation can be addressed by making use of

neighbourhood information exchanged through hello messages. The benefit analysis of TAPMRP carried out in this research shows that the contribution of hello messages to the control overhead of TAPMRP is negligible. Hence, exchanging of hello messages can be included in RTRDM without compromising significantly on resource usage efficiency of TAPMRP.

The last area that requires further investigation is the testing of TAPMRP. In this research, the ideas that constitute TAPMRP were tested using computer simulations which were run in GloMoSim-2.03. All the aims and objectives of the simulations were fulfilled using GloMoSim-2.03. However, while all the other tests were adequately carried out in GloMoSim-2.03, the tests on the seamless route discovery method (subsection 6.7.2) and the probability of making correct switch decisions (subsection 6.7.3) were not tested in an ideal way due to lack of an explicit provision for dynamic transitions between topologies. Hence, there is a need in future to test TAPMRP in network simulators that are more advanced than GloMoSim-2.03. Additionally, testing TAPMRP in a real MANET is likely to give more concrete evidence about its performance. It is therefore important to test TAPMRP in a real MANET in future.

APPENDIX

PSEUDO CODE OF TAPMRP'S FUNCTIONS

The Appendix presents the pseudo code of ten out of twenty two functions of TAPMRP which were developed in this research. Note that the full list of the functions of TAPMRP and their actual code are contained in a folder named “Developed Software” which is on the compact disc that accompanies this thesis.

The style of comments used in the pseudo code is that of C programming language. Any statement in the pseudo code which ends with a semicolon is either a piece of code or represents a piece of code. Each of such statements specifies some action performed by the function in which the statement exists. Note that in some cases the pseudo code is almost the same as the actual code while in other cases, which were chosen arbitrarily, the pseudo code is expressed in the form of an ordinary statement or sentence that ends with a semicolon.

The pseudo code of the ten functions is as follows:

1) **TapmrpSendHello()**

```
{  
    /* This function is executed at every node in a network and after every 30 seconds of  
    * the GloMoSim-2.03 clock.  
    * The purpose of this function is to send a hello message to 1-hop neighbours only.  
    */
```

First, the function uses information of neighbours located within a radius of 2 hops to determine whether the node is a bridge or not using the marking process by J. Wu and H. Li [72];

```
/* Next, the function computes the value of its node redistribution parameter based  
* on the following lines of pseudo code.  
* MAXNBRS stands for the possible maximum number of 1-hop neighbours.  
* 999 stands for a null value.
```

```

*/

/* Starts by preparing the list of 1-hop neighbours, odmrp->Nv[ ], and the table of
* neighbours within 2-hops, odmrp->NNv[ ][ ], for use in the computation by
* excluding weights and cluster status entries.
*/

tempNv[0] = odmrp->Nv[0]; // copies the first entry of odmrp->Nv[ ], which is
// node ID, to a temporary list tempNv[ ].

for (i = 3; i < MAXNBRS; i++)
    tempNv[i - 2] = odmrp->Nv[i]; // copies the rest of the entries of odmrp->Nv[ ]
// to the temporary list tempNv[ ].

for (i = 0; i < MAXNBRS; i++)
    tempNNv[i][0] = odmrp->NNv[i][0]; // copies node ID to a temporary table
// tempNNv[ ][ ].

for (i = 0; i < MAXNBRS; i++)
    for (j = 3; j < MAXNBRS; j++)
        tempNNv[i][j - 2] = odmrp->NNv[i][j]; // copies the rest of the entries of
// odmrp->NNv[ ][ ] to tempNN[ ][ ].

/* Determines the size of tempNv[ ] list */
for (i = 1; i < MAXNBRS; i++)
{
    if (tempNv[i] != 999)
        sizeNv++;
}

/* Determines the intersection of the list of 1-hop neighbours of this node, tempNv[ ],
* with all the records in tempNNv[ ][ ].
* Note that the records in tempNNv[ ][ ] are lists of 1-hop neighbours of 1-hop
* neighbours of this node.
*/

for (i = 0; i < MAXNBRS; i++)
{
    for (j = 0; j < MAXNBRS; j++)

```

```

{
  for (k = 0; k < MAXNBRS; k++)
  {
    if ((tempNv[i] == tempNNv[j][k] && (tempNNv[j][k] != 999))
        counter_1++; // counter_1 records the number of matches of an entry in
                    // tempNv[ ] with entries in the table tempNNv[ ][ ].
    }
  }

  /* Counter_1 is equal to sizeNv when an entry or an element in tempNv[ ] is
  * found in all the records (or lists of 1-hop neighbours) of tempNNv[ ][ ].
  */
  if ((counter_1 == sizeNv) && (sizeNv != 0))
    counter_2++; // counter_2 records the number of elements or entries of
                // tempNv[ ] which are found in all the records of tempNNv[ ][ ].
  counter_1 = 0; // reinitialises counter_1 for reuse.
}

NRParameter = 100 × counter_2 ÷ (sizeNv + 1); // expresses the node redistribution
                                             // parameter as a percentage.
Node_Redistribution_Parameter = (int)NRParameter;

Node_Score = (0.6 × Node_Redistribution_Parameter) + (0.4 ×
Node_Stability_Value);

Finally, the following data are sent to 1-hop neighbours:
1) A list of 1-hop neighbours
2) The node score (or weight)
3) The cluster status (i.e. bridge node, cluster-head, ordinary node, or undecided
node) (Note that an undecided node is a node which has not yet joined a clustering
system; cluster-head and ordinary node status are supplied by another function);

} /* End of TapmrpSendHello( ) */

```

2) TapmrpHandleHello()

```
{
/* TapmrpHandleHello() is invoked whenever a hello message is received from a
* neighbour node.
*/

/* First, the function updates the table of 1-hop neighbours.
* Note that MAXNBRS stands for the possible maximum number of 1-hop
* neighbours, 999 represents a null value, and NBR_EXPIRATION_TIME represents
* a timeout period of a neighbour node.
*/
for (i = 0; i < MAXNBRS; i++)
{
    if (odmrp->neighbourTable[i][0] == source address of the received message)
    {
        odmrp->neighbourTable[i][1] = now; // updates the time when the neighbour was
        // last heard.

        neighbourExists = TRUE;
        break;
    }
}

if (neighbourExists == FALSE)
{
    for (i = 0; i < MAXNBRS; i++)
    {
        if (odmrp->neighbourTable[i][0] == 999)
        {
            odmrp->neighbourTable[i][0] = source address of the received message;
            odmrp->neighbourTable[i][1] = now;
            break;
        }
    }
}
}
```

```

/* Finds and records expired neighbours in the 1-hop neighbour table */
for (i = 0; i < MAXNBRS; i++)
{
    if ((odmrp->neighbourTable[i][0] != 999) && (now - odmrp->neighbourTable[i][1]
        > NBR_EXPIRATION_TIME))
    {
        for (j = 0; j < MAXNBRS; j++)
        {
            if (tempList[j] == 999)
            {
                tempList[j] = (int)odmrp->neighbourTable[i][0]; // copies an expired entry
                                                                    // to a temporary list, tempList[ ]
                break;
            }
        }
    }
}

```

Uses tempList[] to delete expired records in the table of neighbours located within 2 hops, NNv[][];

Removes expired 1-hop neighbours from neighbourTable[][] using data contained in the tempList[];

Uses the received message to either insert a new record or update an existing one in NNv[][], which is the table of neighbours located within 2 hops. Note that each record in NNv[][] consists of a node ID and its list of 1-hop neighbours;

Finally, the received message is destroyed;

```

} /* End of TapmrpHandleHello( ) */

```

3) **TapmrpSendControllerHello()**

```
{  
    /* TapmrpSendControllerHello ( ) sends a controller hello message which contains  
    * the latest route discovery method announced by the controller node.  
    * Only a controller node executes this function after every 120 seconds of the  
    * GloMoSim clock  
    */
```

Prepares a new message;

Gets the route discovery method (MTRDM, CTRDM or RTRDM) determined by the route-discovery-method switch for use in the current controller hello interval and beyond if possible, and puts the method in the message;

Finally, the function sends out the message;

```
} /* End of TapmrpsendControllerHello( ) */
```

4) **TapmrpHandleControllerHello()**

```
{  
    /* This function is executed at a multicast group member whenever a controller hello  
    * packet is received.  
    * Note that in this research the implementation of TAPMRP considered all the nodes  
    * of a network as members of one multicast group.  
    */
```

```
/* Processes the packet only if it is not a duplicate. */
```

```
if {the received packet is not a duplicate )
```

```
{
```

```
    Insert the packet in a message cache;
```

```
    Update the route table;
```

```
    Relay the packet if the number of hops does not exceed the time-to-live;
```

```

/* The next pseudo code determines if it is possible to switch to another route
* discovery method.
* Note that in this implementation of TAPMRP the decision to switch to another
* route discovery method after two controller hello intervals is made at every
* member node of a multicast group and not by the controller node as indicated in
* the thesis.
*/
if (the number of elapsed controller hello intervals since the last decision to switch
to another route discovery method is two or more)
    Replace the route discovery method currently in use by the latest one
    announced by the controller node;

/* Determines the level of mobility of the node */
Calls TapmrpDetermineMobilityLevel( );

/* Sends a reply to the controller node which contains the number of 1-hop
* neighbours of the node.
*/
Calls TapmrpSendControllerReply( );

/* Runs the initialisation procedure of clustering nodes */
if (If the updated route discovery method is CTRDM)
    Call TapmrpInitClusterCTRDM( );

} /* If the packet is not a duplicate */
else
    Destroy the received packet;

} /* End of TapmrpHandleControllerHello( ) */

```

5) TapmrpSendControllerReply()

```
{  
  /* TapmrpSendControllerReply() is invoked at every multicast member node upon  
  * receiving a controller hello message.  
  * The controller reply message contains data pertaining to the number of 1-hop  
  * neighbours of the node which is sending the reply message.  
  */  
  
  Prepares a new message;  
  
  Gets the value of the size of the table of 1-hop neighbours, and puts the value in the  
  reply message;  
  
  Finally, the reply message is sent out by unicasting;  
  
} /* End of TapmrpSendControllerReply() */
```

6) TapmrpHandleControllerReply ()

```
{  
  /* TapmrpHandleControllerHello () is invoked at an intermediate node and finally at  
  * the controller node whenever a controller reply message is received.  
  * Note that the route-discovery-method switch of TAPMRP is implemented by this  
  * function.  
  */  
  
  /* Process the received packet only if it is not a duplicate. */  
  if (the packet is received for the first time)  
  {  
    Insert the packet in the message cache;  
  
    /* The next piece of code is executed if the reply packet has reached the destination  
    * which is the controller node.  
    */  
  }
```

```

if (this is the controller node)
{
    /* Updates the sender information table*/
    for (i = 0; i < possible maximum number of records; i++)
    {
        if (sender of the received packet already exists in the senders' table)
        {
            Update the record corresponding to the sender of the received packet;
            senderExists = TRUE;
            Break;
        }
    } /* End of for loop */

    /* Enters new sender information */
    if (senderExists == FALSE)
        Enter a new entry in the senders' table;

    /* Deletes expired senders */
    for (i = 0; i < possible maximum number of records; i++)
    {
        if (the record has stayed beyond the timeout period for senders)
            Delete the record;
    }

    /* The following lines which include determining the topology of a network
    * topology implement the route-discovery-method switch.
    */
    Calls TapmrpComputeStdev( );

    /* Determines the topology of a network */
    MatrixTopologyVariable = (0.1416 * odmrp->cinfoTableSize) + 0.2259;
    ClusterbasedTopologyVariable = (0.1713 * odmrp->cinfoTableSize) + 0.0946;

```

```

if (output of TapmrpComputeStdev( ) < MatrixTopologyVariable)
    The prevailing topology is an ELC-matrix topology;
else
    if ((output of TapmrpComputeStdev( ) > MatrixTopologyVariable) &&
        (output of TapmrpComputeStdev( ) < ClusterbasedTopologyVariable))
        The prevailing topology is an ELC-cluster-based topology;
    else
        The prevailing topology is an ELC-random topology;

    Destroys the received message;

} /* End of if (this is a controller node) */

else
    /* Packet is relayed */
    if (routing information is available)
    {
        Relay the received packet;
    } else
        Destroy the received packet;

} /* End of if (the packet is received for the first time) */

else
    Destroy the received packet;

} /* End of TapmrpHandleControllerReply( ) */

```

7) TapmrpInitClusterCTRDM()

```
{
/* TapmrpInitClusterCTRDM( ) initialises a clustering process of nodes.
* The resulting logical clusters are used by CTRDM.
*/

/* Records all neighbours with weights greater than or equal to that of this node.
* MAXNBRS stands for possible maximum number of 1-hop neighbours.
*/
for (i = 0; i < MAXNBRS; i++)
    Find nodes with weights greater than or equal to that of this node and put them
    in ChJoinCheck[i][j];

/* Records all neighbours with weights less than that of this node */
for (i = 0; i < MAXNBRS; i++)
    Find nodes with weights less than that of this node and put them in
    ChJoinLessCheck[i][j];

/* Finds a node with the highest weight in the neighbourhood of this node */
for (i = 0; i < MAXNBRS; i++)
    Find a node with the highest weight in the neighbourhood of this node; in the case
    of a tie, the node with a lower ID is selected;

/* sends a CH(v) message to neighbours */
if (this is the node with the highest weight)
{
    Set cluster status of this node to cluster-head;
    Call TapmrpChMessageCTRDM( ); // sends a CH(v) message
}

} /* End of TapmrpInitClusterCTRDM ( ) */
```

8) TapmrpChMessageCTRDM()

```
{
/* This function is used to send a cluster-head announcement to neighbours.
* The node that sends the announcement is called a primary cluster-head.
* The function is also used by the primary cluster-head to appoint secondary
* cluster-heads if possible.
*/

/* The following lines of pseudo code implement the secondary cluster-head idea.
* A primary cluster-head appoints secondary cluster-heads and assigns members to
* them
*/

/* Determines the number of 1-hop neighbours.
* MAXNBRS is the possible maximum number of 1-hop neighbours.
*/
for (i = 0; i < MAXNBRS; i++)
    Count each neighbour and store the result in clusterSize;

/* Starts the process of creating a second cluster and appoints one secondary
* cluster-head to be in charge of the second cluster.
* This is the case in which the clusterSize can only allow the appointment of only
* one secondary cluster-head.
* CLUSTER_MAXSIZE is the upper bound on the cluster size and was set at 15
* nodes.
*/
if ((clusterSize > CLUSTER_MAXSIZE) && (clusterSize <= (2 *
    CLUSTER_MAXSIZE)))
{
    Copy entries of the table of neighbours within 2-hops and store them in a temporary
    table called tempNNv[ ][ ];

    Determines the intersection of each record in tempNNv[ ][ ] with the list of 1-hop
    neighbours of this node, and stores the results in the last column of tempNNv[ ][ ];
```

Uses the data stored in last column of tempNNv[][] to determine the record or node ID with the highest intersection;

```
/* Assigns members to a second cluster starting with the newly selected
 * secondary cluster-head
 */
chPkt->secondClusterList[0] = Largest intersection node; // assigns the
                                                         // secondary cluster-head
```

Copies the record of the node with the largest intersection in tempNNv[][] to a temporary list called tempList[];

Also copies the list of 1-hop neighbours of this node (i.e. the primary cluster-head) to a temporary table called tempList2[];

```
/* The primary cluster-head finds neighbours that it shares with the newly
 * appointed secondary cluster-head and assigns them to the secondary cluster-
 * head.
```

```
* Note that 999 stands for a null value.
```

```
*/
```

```
for (i = 3; i < MAXNBRS - 1; i++)
  for (j = 3; j < MAXNBRS - 1; j++)
  {
    if ((tempList[i] == tempList2[j]) && (tempList[i] != 999))
    {
      for (k = 1; k < MAXNBRS - 1; k++)
      {
        if (chPkt->secondClusterList[k] == 999)
        {
          chPkt->secondClusterList[k] = tempList[i];
          counter_2++;
          break;
        }
      }
    }
  }
```

```

        }
    }

    if(counter_2 >= CLUSTER_MAXSIZE)
        break;
}

} /* End of first if */

/* The next lines of pseudo code are for the case when the clusterSize is large enough to
* allow the appointment of two secondary cluster-heads each with its own cluster.
* The process first repeats the one above for one secondary cluster-head and then
* proceeds after that to appoint the second secondary cluster-head.
*/
if (clusterSize > (2 * CLUSTER_MAXSIZE))
{
    Repeats the procedure outlined above for appointing the first secondary cluster-head;

    /* Next the primary cluster-head appoints the second secondary cluster-head, and the
    * process starts by making some deletions.
    */
    /* First to be deleted are members of the first secondary cluster-head from the 1-hop
    * neighbour list of the primary cluster-head.
    */
    for (i = 0; i < MAXNBRS; i++)
        for (j = 3; j < MAXNBRS; j++)
        {
            if ((chPkt->secondClusterList[i] == tempList2[j]) &&
                (chPkt->secondClusterList[i] != 999))
                tempList2[j] = 999; // deletion is done here
        }

    /* Deletes members of the first secondary cluster-head from tempNNv[ ][ ] */
    for (i = 0; i < MAXNBRS; i++)

```

```

{
  for (j = 0; j < MAXNBRS; j++)
  {
    for (k = 3; k < MAXNBRS; k++)
    {
      if ((chPkt->secondClusterList[i] == tempNNv[j][k]) && (tempList[i] != 999))
        tempNNv[j][k] = 999; // deletion takes place here
    }
  }
}

```

/* Deletes entries of column 3 of tempNNv[][] which have already been considered
* in appointing the first secondary cluster-head.
*/

```

for (i = 0; i < MAXNBRS; i++)
  for (j = 0; j < MAXNBRS; j++)
  {
    if ((chPkt->secondClusterList[i] == tempNNv[j][0]) &&
        (chPkt->secondClusterList[i] != 999))
      tempNNv[j][2] = 100; // 100 means a null value
  }

```

/* Deletes the last column entries in tempNNv[][] */

```

for (i = 0; i < MAXNBRS; i++)
  tempNNv[i][MAXNBRS - 1] = 999;

```

```

for (i = 0; i < MAXNBRS; i++)
  tempList[i] = 999; // prepares tempList[ ] for reuse

```

Determines the intersection of each remaining record in tempNNv[][] with the remaining entries in tempList2[], and stores the results in the last column of tempNNv[][];

Determines the record or node ID with the highest intersection in tempNNv[][] using

the data stored in the last column of tempNNv[][];

```
/* Assigns members to a third cluster starting with the newly selected second
 * secondary cluster-head.
```

```
*/
```

```
chPkt->thirdClusterList[0] = Largest intersection Node; // assigns the second
// secondary cluster-head
```

The primary cluster-head finds neighbours that it shares with the newly appointed second secondary cluster-head and assigns them to the second secondary cluster-head, just like in the case of the first secondary cluster-head;

Finally, the primary cluster-head, which appoints itself as the cluster-head by virtue of having the largest node weight in its neighbourhood, prepares a cluster-head announcement message and puts in the message chPkt->secondClusterList[] or chPkt->secondClusterList[] and chPkt->thirdClusterList[] if available, and then sends out the message;

```
} /* End of TapmrpChMessageCTRDM( ) */
```

9) TapmrpSelectFwdNodesMTRDM()

```
{
```

```
/* TapmrpSelectFwdNodesMTRDM( ) is used by MTRDM to select forward nodes
 * among 1-hop neighbours and puts the resulting list in a Join-Query packet.
```

```
 * This function is executed at a source node and a forward node.
```

```
 * /
```

```
/* Execution of this function starts by copying all the contents of odmrp->NNv[ ][ ],
 * which stores neighbours within 2-hops, to a another table, odmrp-> K[ ][ ].
```

```
 * MAXNBRS stands for the possible maximum number of 1-hop neighbours.
```

```
*/
```

```
for (i = 0; i < MAXNBRS; i++)
```

```
    for (j = 0; j < MAXNBRS; j++)
```

```
odmrp->K[i][j] = odmrp->NNv[i][j];
```

```
/* The next lines of pseudo code implement the partial dominant algorithm [39].  
* The implementation starts by deleting the list of 1-hop neighbours of the sender  
* node u of the just received Join-Query packet from the table K[ ][ ].  
*/  
if (address of this node != address of the source node) // applies to non-source  
//nodes only.  
{  
  for (i = 0; i < MAXNBRS; i++)  
  {  
    if (odmrp->K[i][0] == last node address of the just received Join-Query)  
    {  
      for (j = 3; j < MAXNBRS; j++)  
        tempList[j] = odmrp->K[i][j]; //copies record of node u to a temp list  
      break;  
    }  
  }  
  
  for (i = 3; i < MAXNBRS; i++)  
  {  
    for (j = 0; j < MAXNBRS; j++)  
    {  
      for (k = 3; k < MAXNBRS; k++)  
      {  
        if (tempList[i] == odmrp->K[j][k])  
          odmrp->K[j][k] = 999; // where deletion takes place  
      }  
    }  
  }  
}  
  
/* Deletes entries of the set P defined in [39] from the table K[ ][ ].  
* This applies only to a non-source node.  
*/
```

```

for (i = 3; i < MAXNBRS; i++)
    for (j = 3; j < MAXNBRS; j++)
        if (tempList[i] == odmrp->Nv[j])
            tempList2[i] = tempList[i]; // creates the intersection set and stores it in
                                        // tempList2[ ]

```

```

for (i = 3; i < MAXNBRS; i++)
{
    for (j = 0; j < MAXNBRS; j++)
    {
        if ((tempList2[i] == odmrp->K[j][0]) && (tempList2[i] != 999))
        {
            for (k = 3; k < MAXNBRS; k++)
                tempList3[k] = odmrp->K[j][k]; //stores a record in tempList3[ ]
                                                // corresponding to an element or node-ID in tempList2[ ]
        }
    }
}

```

```

for (j = 3; j < MAXNBRS; j++)
    for (k = 0; k < MAXNBRS; k++)
        for (l = 3; l < MAXNBRS; l++)
        {
            if ((tempList3[j] == odmrp->K[k][l]) && (tempList3[j] != 999))
                odmrp->K[k][l] = 999; //deletes entries in K[ ][ ] that appear in
                                        // templist3[ ]
        }

```

```

} /* End of for loop */

```

Clears tempList[] for reuse;

Clears tempList2[] for reuse;

```

} /* End of first if statement */

```

Removes the list of 1-hop neighbours of this node from the table K[][];

Removes entries of the source node of the just received Join-Query packet from the table $K[][]$;

/* The next lines of the pseudo code implements the greedy set cover algorithm [7].

* The implementation is done using the **do ... while** loop.

*/

do {

Determines the size of each record in $K[][]$ and stores the results in the last column of $K[][]$;

/* Next, determines the record or node ID with the highest size (or the node with
* the highest number of 1-hop neighbours) using the data stored in the last
* column of the table $K[][]$.

*/

largestSize = odmrp->K[0][MAXNBRS - 1];

NodeWithLargestSize = odmrp->K[0][0];

for (i = 1; i < MAXNBRS; i++)

{

if ((odmrp->K[i][1] != 1) && (odmrp->K[i][MAXNBRS - 1] != 0))

//excludes marked and null records

{

if (largestSize < odmrp->K[i][MAXNBRS - 1])

{

largestSize = odmrp->K[i][MAXNBRS - 1];

NodeWithLargestSize = odmrp->K[i][0];

}

}

}

/* Puts the node with the highest number of 1-hop neighbours on the list

* of forward nodes, odmrp->F[].

* Note that 999 stands for a null value.

*/

```

if ((largestSize > 0) && (largestSize < 999))
{
    for (i = 0; i < MAXNBRS; i++)
    {
        if (odmrp->F[i] == 999)
        {
            odmrp->F[i] = NodeWithLargestSize;
            break;
        }
    }
}

```

Copies entries of NodeWithLargestSize from K[][] to a temporary list;

Uses the temporary list to remove entries of NodeWithLargestSize from the table K[][];

Deletes entries in the last column of K[][];

```

if (largestSize == 999)
    largestSize = 0; // a situation where K[ ][ ] has no element to be processed.

```

```

} while (largestSize > 0);

```

```

} /* End of TapmrpSelectFwdNodesMTRDM( ) */

```

10) TapmrpSelectFwdNodesCTRDM()

```

{
    /* This function is used by CTRDM to select forward nodes of a Join-Query packet.
    * Only a cluster-head node executes this function by selecting forward nodes
    * among bridge nodes.
    */

    /* Execution of this function begins by copying contents of the table of neighbours
    * within 2-hops, odmrp->NNv[ ][ ], to another table, odmrp->K[ ][ ].
    * MAXNBRS stands for possible maximum number of 1-hop neighbours.
    */

    for (i = 0; i < MAXNBRS; i++)
        for (j = 0; j < MAXNBRS; j++)
            odmrp->K[i][j] = odmrp->NNv[i][j];

    /* Selection of the forward node set begins with a pruning operation which in turn
    * starts with the deletion of a cluster-head set  $C(u)$  from the sender node  $u$  defined
    * in [46] from the table K[ ][ ].
    */

    if (address of this node != address of the source node) // applies to non-source
                                                            // nodes only.

    {
        for (i = 0; i < MAXNBRS; i++)
        {
            for (j = 0; j < MAXNBRS; j++)
            {
                for (k = 3; k < MAXNBRS; k++)
                {
                    if (queryPkt->clusterheadList[i] == odmrp->K[j][k])
                        odmrp->K[j][k] = 999; // deletion takes place here
                }
            }
        }
    }
}

```

Removes the list of 1-hop neighbours of this node from the table $K[i][j]$;

Removes occurrences of this node from the table $K[i][j]$;

Deletes occurrences of the source node in the table $K[i][j]$;

/* The next lines of pseudo code delete ordinary nodes from $K[i][j]$, and start by

* copying all ordinary nodes from $K[i][j]$ to a temporary list.

* Note that 999 stands for a null value.

*/

```
for (i = 0; i < MAXNBRS; i++)
```

```
{
```

```
    if (odmrp->K[i][2] == 0)
```

```
    {
```

```
        for (j = 0; j < MAXNBRS; j++)
```

```
        {
```

```
            if (tempList[j] == 999)
```

```
            {
```

```
                tempList[j] = odmrp->K[i][0]; // copies all ordinary nodes in  $K[i][j]$ 
```

```
                // to tempList[j].
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

/* Where actual deletion of ordinary nodes takes place */

```
for (i = 0; i < MAXNBRS; i++)
```

```
{
```

```
    for (j = 0; j < MAXNBRS; j++)
```

```
    {
```

```
        for (k = 3; k < MAXNBRS; k++)
```

```
        {
```

```
            if (tempList[i] == odmrp->K[j][k])
```

```

        odmrp->K[j][k] = 999; // deletion takes place here
    }
}
}

```

Copies the remaining contents of $K[][]$ to a list of cluster-heads $C(u)$ which is put in a forwarded Join-Query packet;

Deletes entries of column 1 of $K[][]$ so as to use the column for purposes other than storing weights of nodes;

Clears `tempList[]` for reuse;

```

/* The following lines of pseudo code implement the greedy set cover algorithm
* [7] using the do ... while loop.
*/

```

```

*/

```

```

do {

```

```

    Determines the size of each record in  $K[ ][ ]$  and stores the results in the
    last column of  $K[ ][ ]$ ;

```

```

    Determines the record (or node ID) with the highest size or the highest
    number of 1-hop neighbours using the data stored in the last column of
     $K[ ][ ]$ ;

```

```

    Puts the node with the highest size on the list of forward nodes;

```

```

/* The next piece of pseudo code deletes the list of 1-hop neighbours of the
* node with the largest record size.

```

```

* The deletion starts by copying the record of the node to a temporary list.

```

```

*/

```

```

for (i = 0; i < MAXNBRS; i++)

```

```

{

```

```

    if (odmrp->K[i][0] == NodeWithLargestSize)

```

```

    {

```

```

        for (j = 3; j < MAXNBRS; j++)
            tempList2[j] = odmrp->K[i][j];
    }
}

/* Actual deletion entries of NodeWithLargestSize in K[ ][ ] */
for (i = 3; i < MAXNBRS; i++)
{
    for (j = 0; j < MAXNBRS; j++)
    {
        for (k = 3; k < MAXNBRS; k++)
        {
            if (tempList2[i] == odmrp->K[j][k])
                odmrp->K[j][k] = 999; // deletion takes place here
        }
    }
}

Deletes entries of the last column in K[ ][ ] for reuse;

if (largestSize == 999)
    largestSize = 0; // a situation where K[ ][ ] has no element to be processed.

} while (largestSize > 0);

} /* End of TapmrpSelectFwdNodesCTRDM() */

```

REFERENCES

- [1] J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester, “An overview of mobile ad hoc networks: applications and challenges”, *Journal of Communications and Network*, FITCE, vol. 3, pp. 60 – 66, 2004.
- [2] M. Ilyas, “The handbook of ad hoc wireless networks”, *chapter 2*, CRC Press LLC, Boca Raton, Florida, USA, 2003.
- [3] S. K. Soni and T. C. Aseri, “A review of current multicast routing protocols of mobile ad hoc network”, *2nd IEEE International Conference on Computer Modelling and Simulation ICCMS’10*, vol. 3, pp. 207 – 211, 2010.
- [4] S. Corson and J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, *Network Working Group, Request for Comments (RFC): 2501*, The Internet Society, 1999.
- [5] UCLA, “GloMoSim-2.0: A scalable simulation environment for wireless and wired network systems”, *Parallel Computing Laboratory, Computer Science Department, University of California, Los Angeles*. <http://pcl.cs.ucla.edu/projects/glomosim/>, Last accessed April 2011.
- [6] P. Mohapatra, C. Gui and J. Li, “Group communications in mobile ad hoc networks”, *Computer (IEEE Journal)*, vol. 37, issue 2, pp. 52 – 59, 2004.
- [7] H. Lim and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks”, *3rd ACM International Workshop on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, pp. 61 – 68, 2000.
- [8] S. Ni, Y. Tseng, Y. Chen and J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking MOBICOM’99*, pp. 151 – 162, 1999.
- [9] Y. Yi, S. Lee, W. Su and M. Gerla, “On-Demand Multicast Routing Protocol (ODMRP) for ad hoc networks”, *IETF Internet draft: draft-ietf-manet-odmrp-04.txt*, February 2003.
- [10] E. M. Royer and C. E. Perkins, “Multicast Ad hoc On-demand Distance Vector (MAODV) routing”, *IETF Internet draft: draft-ietf-manet-maodv-00.txt*, July 2000.
- [11] W. Lou and J. Wu, “Efficient broadcast with forward node set in clustered mobile ad hoc networks,” *11th IEEE International Conference on Computer Communications and Networks*, pp. 398 – 403, 2002.

- [12] S. Funke, A. Kesselman, U. Meyer and M. Segal, "A simple improved distributed algorithm for minimum CDS in unit disk graphs", *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WiMob'2005*, Vol. 2, pp. 220 – 223, 2005.
- [13] C. Chang and J. Sheu, "Design and implementation of ad hoc classroom and eSchoolbag systems for ubiquitous learning," *IEEE international workshop on Wireless and Mobile Technologies in Education (WMTE'02)*, 2002.
- [14] C. Chang, J. Sheu and T. Chan, "Concept and design of ad hoc and mobile classrooms," *Journal of Computer Assisted Learning*, vol. 19, pp.336-346, September 2003.
- [15] S. Vijayaragavan, K. Duraiswamy, B. Kalaavathi and S. Madhavi, "A performance study of reactive multicast routing protocols in virtual classroom using mobile ad hoc network," *Journal of Computer Science* 5 (11), pp. 788-793, 2009.
- [16] E. M. Royer and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks", *IEEE Personal Communications (Journal)*, vol. 6, issue 2, pp. 46 – 55, 1999.
- [17] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) for mobile computers", *ACM SIGCOMM Computer Communication Review*, vol. 24, issue 4, pp. 234 – 244, 1994.
- [18] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing", *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90 – 100, 1999.
- [19] C. M. Cordeiro, H. Gossain and D. P. Agrawal, "Multicast over wireless mobile ad hoc networks: present and future directions", *IEEE Network (Journal)*, vol. 17, issue 1, pp. 52 – 59, 2003.
- [20] O. Badarneh and M. Kadoch, "Multicast routing protocols in mobile ad hoc networks: a comparative survey and taxonomy", *EURASIP Journal of Wireless Communications and Networking*, volume 2009, ISSN: 1687-1472, 2009.
- [21] A. A. Hashim, M. M. Qabajeh, O. Khalifa and L. Qabajeh, "Review of multicast QoS routing protocols for mobile ad hoc networks", *International Journal of Computer Science and Network Security IJCSNS*, vol. 8 no. 12, 2008.
- [22] S. K. Sarkar, T. G. Basavaraju and C. Puttamadappa, "Ad hoc mobile wireless networks: principles, protocols and applications", *Auerbach Publications – Taylor & Francis Group*, Chapter 4, ISBN 13:978-1-4200-6221-2, 2008.

- [23] C. W. Wu, Y. C. Tay and C. Toh, “Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) functional specification”, *IETF Internet draft: draft-ietf-manet-amris-spec-00.txt*, 1998.
- [24] C. Toh, G. Guichal and S. Bunchua, “ABAM: On-demand associativity-based multicast routing for ad hoc mobile networks”, *52nd IEEE Conference on Vehicular Technology VTS-Fall VTC 2000*, vol. 3, pp. 987 – 993, 2000.
- [25] R.S. Sisodia, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, “A preferred link based multicast protocol for wireless mobile ad hoc networks”, *IEEE International Conference on Communications (ICC)*, vol. 3, pp. 2213 – 2217, 2003.
- [26] L. Canourgues, J. Lephay, L. Soyer and A. Beylot, “STAMP: Shared-tree ad hoc multicast protocol”, *IEEE Conference on Military Communications MILCOM 2006*, pp. 1 – 7, 2006.
- [27] C. C. Huang, R. S. Chang and M. H. Guo, “Weight-based clustering multicast routing protocol for mobile ad hoc networks”, *IEEE Conference on Wireless Communications and Networking WCNC 2003*, vol. 2, pp. 1112 – 1117, 2003.
- [28] C. Oh, J. Park, J. Ahn, M. Lee, T. Lee, W. Cha and P. S. Mah, “Tree-based multicast protocol using multi-point relays for mobile ad hoc networks”, *3rd IEEE International Conference on Ubiquitous and Future Networks ICUFN 2011*, pp. 174 – 178, 2011.
- [29] T. Ozaki, J. B. Kim and T. Suda, “Bandwidth-efficient multicast routing protocol for ad hoc networks”, *8th IEEE International Conference on Computer Communications and Networks*, pp. 10 – 17, 1999.
- [30] Y. S. Chen, T. S. Chen and C. J. Huang, “SOM: Spiral-fat-tree-based on-demand multicast protocol in a wireless ad hoc network”, *15th IEEE International Conference on Information Networking*, pp. 17 – 24, 2001.
- [31] F. Sato, “An efficient shared-tree-based multi-source multicast routing protocol in mobile ad hoc networks”, *IEEE International Conference on Advanced Information Networking and Applications Workshops WAINA '09*, pp. 377 – 382, 2009.
- [32] S. Sajama and Z. J. Haas, “Independent-Tree Ad hoc Multicast Routing (ITAMAR)”, *54th IEEE Conference on Vehicular Technology VTC 2001 Fall VTS*, vol. 2, pp. 600 – 604, 2001.
- [33] J. J. Garcia-Luna-Aceves and E. L. Madruga, “The core-assisted mesh protocol”, *IEEE Journal on Selected Areas in Communications*, vol. 17, issue 8, pp. 1380 – 1394, 1999.

- [34] C. Chiang, M. Gerla and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for multihop mobile wireless networks", *Journal of Cluster Computing (Springer)*, vol. 1, no. 2, pp. 187 – 196, 1998.
- [35] M. Chatterjee, S. K. Das and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks", *Cluster Computing 5*, Kluwer Academic Publishers, Netherlands, pp. 193 – 204, 2002.
- [36] Y. Qin, "Analysis of cluster-based hierarchical routing in ad hoc wireless networks", *Electronics Letters (IET Journal)*, vol. 42, issue 8, pp. 474 – 476.
- [37] H. Moustafa and H. Labiod, "A multicast on-demand mesh-based routing protocol in multihop mobile wireless networks", *58th IEEE Conference on Vehicular Technology VTC 2003-Fall*, vol. 4, pp. 2192 – 2196, 2003.
- [38] S. Lee and C. Kim, "Neighbor supporting ad hoc multicast routing protocol", *1st Annual IEEE Workshop on Mobile and Ad Hoc Networking and Computing MobiHOC 2000*, pp. 37 – 44, 2000.
- [39] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks", *IEEE Transaction on Mobile Computing*, Vol. 1, No. 2, pp. 111-123, 2002.
- [40] H. Lim and C. Kim, "Flooding in wireless ad hoc networks", *Elsevier-Computer Communications 24(2001)*, pp. 353 – 363, 2001.
- [41] S. K. Das, B. S. Manoj and C. S. R. Murthy, "A dynamic core based multicast routing protocol for ad hoc wireless networks", *3rd ACM International Symposium on Mobile Ad hoc Networking and Computing MobiHoc '02*, 2002.
- [42] R. Vaishampayan and J. J. Garcia-Luna-Aceves, "Efficient and robust multicast routing in mobile ad hoc networks", *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 304 – 313, 2004.
- [43] C. C. Shen and C. Jaikao, "Ad hoc multicast routing algorithm with swarm intelligence", *Journal of Mobile Networks and Applications*, vol. 10, issue 1-2, Kluwer Academic Publishers, 2005.
- [44] J. Xie, R. R. Talpade, A. McAuley and M. Liu, "AMRoute: Ad hoc Multicast Routing protocol", *Journal of Mobile Networks and Applications (Springer)*, vol. 7, no. 6, pp. 429 – 439, 2002.
- [45] P. Sinha, R. Sivakumar and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing", *IEEE Conference on Wireless Communications and Networking*, vol. 3, pp. 1313 – 1317, 1999.

- [46] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks", *Journal of Wireless Communication and Mobile Computing*, vol. 3, pp. 155 – 173, 2003.
- [47] W. Lou and J. Wu, "A cluster-based backbone infrastructure for broadcasting in MANETs", *IEEE International Symposium on Parallel and Distributed Processing*, 2003.
- [48] X. Zhang and L. Jacob, "Multicast zone routing protocol in mobile ad hoc wireless networks", *28th Annual IEEE International Conference on Local Computer Networks LCN'03*, pp. 150 – 159, 2003.
- [49] J. Biswas, M. Barai and S. K. Nandy, "Efficient hybrid multicast routing protocol for ad hoc wireless networks", *29th Annual IEEE International Conference on Local Computer Networks*, pp. 180 – 187, 2004.
- [50] L. Ji and M. S. Corson, "Differential destination multicast: a MANET multicast routing protocol for small groups", *20th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001*, vol. 2, pp. 1192 – 1201, 2001.
- [51] W-H. Cheng, C-Y. Wen and K-T. Feng, "Power-controlled hybrid multicast routing protocol for mobile ad hoc networks", *63rd IEEE Conference on Vehicular Technology*, vol. 3, pp. 1087 – 1091, 2006.
- [52] C. K. Chen, K. Wang and L. S. Lee, "A hybrid overlay multicast routing protocol for mobile ad hoc networks", *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, 784 – 789, 2005.
- [53] J. Lu and L. Bin, "Hybrid zone based multicast routing scheme for mobile ad hoc network", *4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing WiCOM '08*, pp. 1 – 4, 2008.
- [54] W. K. Lai, M. L. Weng, C-H. Lin and C-S. Shieh, "Adaptive multicast routing protocol for wireless mobile ad hoc networks", *5th IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing IIH-MSP '09*, pp. 844 – 847, 2009.
- [55] C. C. Chiang, M. Gerla and L. Zhang, "Adaptive shared tree multicast in mobile wireless networks", *IEEE Conference on Global Telecommunications GLOBECOM 98, the Bridge to Global Integration*, vol. 3, pp. 1817 – 1822, 1998.
- [56] S. Park and D. Park, "Adaptive core multicast routing protocol", *Journal of Wireless Networks, Kluwer Academic Publishers*, vol. 10, issue 1, pp. 53 – 60, 2004.

- [57] A. B. Mnaouer, L. Chen, C. H. Foh and J. W. Tantra, "A new polymorphic multicast routing protocol for MANET", *IEEE International Conference on Communications*, vol. 1, pp. 233 – 238, 2005.
- [58] I. Yamamoto, T. Ohta and Y. Kakuda, "An adaptive inter-cluster data forwarding configuration mechanism for hierarchical multicast routing protocol in mobile ad hoc networks", *29th IEEE International Conference on Distributed Computing Systems Workshops ICDCS Workshops '09*, pp. 330 – 337, 2009.
- [59] A. B. Mnaouer, L. Chen, C. H. Foh, and J. W. Tantra, "OPHMR: An optimized polymorphic hybrid multicast routing protocol for MANET", *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 551 – 562, 2007.
- [60] S. Y. Oh, J. S. Park and M. Gerla, "E-ODMRP: Enhanced ODMRP with motion adaptive refresh", *2nd IEEE International Symposium on Wireless Communication Systems*, pp. 130 – 134, 2005.
- [61] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks", *2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc '01*, 2001.
- [62] D. Kim, C. Toh, J. Cano and P. Manzoni, "A bounding algorithm for the broadcast storm problem in mobile ad hoc networks", *IEEE Conference on Wireless Communications and Networking WCNC*, Vol. 2, pp. 1131 – 1136, 2003.
- [63] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks", *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 194 – 205, 2002.
- [64] D. Kouvatsos and I. Mkwawa, "Broadcasting methods in mobile ad hoc networks: an overview", *Proceedings of the HetNet*, UK, 2005.
- [65] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, D. J. Rosenkrantz, "Simple heuristics for unit disk graphs", *Networks (Journal)*, Vol. 25, pp. 59 – 68, 1995.
- [66] F. Kuhn, T. Moscibroda and R. Wattenhofer, "Unit disk graph approximation," *Joint Workshop on Foundations of Mobile Computing DIALM-POMC'04*, ACM press, pp. 17 – 23, 2004.
- [67] B. Lee, C. Yu and S. Moh, "Issues in scalable clustered network architecture for mobile ad hoc networks". *Mobile Computing Handbook, Chapter 26, CRC Press*, 2005.
- [68] R. Purtoosi, H. Taheri, A. Mohammadi and F. Foroozan, "A high performance cluster-based flooding algorithm for wireless ad hoc networks," *5th IEEE*

- International Conference on Computer and Information Technology CIT 2005*, pp. 417 – 423, 2005.
- [69] R. Purtoosi, H. Taheri, A. Mohammadi and F. Foroozan, “A light-weight contention-based clustering algorithm for wireless ad hoc networks”, *4th International Conference on Computer and Information Technology (CIT2004)*, Wuhan, China, pp. 627 – 632, 2004.
- [70] A. Pandey and J. S. Lim, “CTB-MAC: Cluster-based TDMA broadcast MAC protocol for mobile ad hoc network”, *5th IEEE International Conference on Future Information Technology (FutureTech)*, pp. 1 – 6, 2010.
- [71] I. Stojmenovic, M. Seddigh and J. Zunic, “Internal nodes based broadcasting in wireless networks”, *34th IEEE Annual Hawaii International Conference on System Sciences*, 2001.
- [72] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks”, *3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIALM’99*, pp. 7 – 14, 1999.
- [73] M. Hoque, F. Rahman, S. K. Kundu and A. Rahman, “Enhanced partial dominant pruning (EPDP) based broadcasting in ad hoc wireless networks,” *IEEE International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS*, pp. 143 – 150, 2008.
- [74] N. Mitton and E. Fleury, “Efficient broadcasting in self-organizing multi-hop wireless networks”, *Ad Hoc Networks & Wireless (Ad Hoc-Now’05)*, Cancun, Mexico, pp. 192 – 206, 2005.
- [75] K. Drira and H. Kheddouci, “A new clustering algorithm for MANETs”, *14th IEEE International Symposium (NETWORKS) on Telecommunications Network Strategy and Planning*, pp. 1 – 6, 2010.
- [76] Y. Sasson, D. Cavin and A. Schiper, “Probabilistic broadcast for flooding in wireless mobile ad hoc networks”, *IEEE Conference on Wireless Communications and Networking WCNC 2003*, vol. 2, pp. 1124 – 1130, 2003.
- [77] J. Cartigny and D. Simplot, “Border node retransmission based probabilistic broadcast protocols in ad hoc networks”, *Telecommunication Systems*, pp. 189 – 204, 2003.
- [78] Q. Zhang and D. P. Agrawal, “Dynamic probabilistic broadcasting in mobile ad hoc networks”, *58th IEEE Conference on Vehicular Technology VTC 2003-Fall*, vol. 5, pp. 2860 – 2864, 2003.

- [79] D. J. Scott and A. Yasinsac, "Dynamic probabilistic retransmission in ad hoc networks", *International Conference on Wireless Networks ICWN'04*, CSREA press, pp. 158 – 164, 2004.
- [80] T. Bano and J. Singhai, "Probabilistic broadcasting protocol in ad hoc network and its advancement: a review", *International Journal of Computer Science and Engineering Survey (IJCES)*, vol. 1, no. 2, 2010.
- [81] M. Ould-Khaoua, "Performance analysis of adjusted probabilistic broadcasting in mobile ad hoc networks", *International Journal of Wireless Information Networks* 13(2), pp. 127 – 140, 2006.
- [82] M. B. Yassein and M. B. Khalaf, "A performance comparison of smart probabilistic broadcasting of Ad hoc Distance Vector (AODV)" *Journal of Supercomputing*, 2010.
- [83] A. Mohammed, "An efficient counter-based broadcast scheme for mobile ad hoc networks", *4th European Performance Engineering Conference on Formal Methods and Stochastic Models for Performance Evaluation*, Springer-Verlag, 2007.
- [84] A. Mohammed, M. Ould-Khaoua, L. M. Mackenzie, and J. Abdulai, "An adjusted counter-based broadcast scheme for mobile ad hoc networks", *10th IEEE International Conference on Computer Modelling and Simulation*, pp. 441 – 446, 2008.
- [85] A. Mohammed, M. Ould-Khaoua, L. M. Mackenzie and J. Abdulai, "Dynamic probabilistic counter-based broadcasting in mobile ad hoc networks", *2nd IEEE International Conference on Adaptive Science and Technology ICASST 2009*, pp. 120 – 127, 2009.
- [86] M. B. Yassein, S. F. Nimer and A. Y. Al-Dubai, "A new dynamic counter-based broadcasting scheme for mobile ad hoc networks", *Journal of Simulation Modelling Practice and Theory* 19 (2011), Elsevier, pp. 553 – 563, 2011.
- [87] A. Keshavarz-Haddad, V. Ribeiro and R. Riedi, "Color-based broadcasting for ad hoc networks", *4th IEEE International Symposium on Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks WIOPT'06*, Boston MA, pp. 1 – 10, 2006.
- [88] M. T. Sun, W. C. Feng and T. H. Lai, "Location aided broadcast in wireless ad hoc networks", *IEEE Conference on Global Telecommunications GLOBECOM'01*, vol. 5, pp. 2842 – 2846, 2001.
- [89] M. T. Sun, and T. H. Lai, "Location aided broadcast in wireless ad hoc network systems", *IEEE Conference on Wireless Communications and Networking WCNC 2002*, vol. 2, pp. 597 – 602, 2002.

- [90] X. Liu, X. Jia, H. Liu and L. Feng, "A location aided flooding protocol for wireless ad hoc networks", *3rd International Conference on Mobile Ad-Hoc and Sensor Networks MSN'07*, Springer-Verlag Berlin, 2007.
- [91] S. Jun, Y. Kun, C. Yuanda, and S. Dengfu, "RLABP: A receiver-deciding location-aided broadcast protocol for mobile ad hoc networks", *IMACS Multiconference on Computational Engineering in Systems Applications*, Beijing, China, pp. 1631 – 1638, 2006.
- [92] Y. Cai, K. A. Hua and A. Phillips, "Leveraging 1-hop neighbourhood knowledge for efficient flooding in wireless ad hoc networks", *24th IEEE International Conference on Performance, Computing, and Communications, IPCCC 2005*, pp. 347 – 354, 2005.
- [93] K. Kim, Y. Cai and W. Tavanapong, "A priority forwarding technique for efficient and fast flooding in wireless ad hoc networks", *14th IEEE International Conference on Computer Communications and Networks, ICCCN 2005*, pp. 223 – 228, 2005.
- [94] A. Rahman and P. Gburzynski, "MAC-assisted broadcast speedup in ad hoc wireless networks", *International Conference on Wireless Communications and Mobile Computing IWCMC'06*, ACM Press, 2006.
- [95] M. Chawla, J. Singhai, S. Jain, and A. Shrivastava, "Node stability based clustering algorithm for mobile ad hoc networks", *4th IEEE International Conference on Wireless Communication and Sensor Networks WCSN*, pp. 31 – 35, 2008.
- [96] D. Gavalas, G. Pantziou, C. Konstantopoulos, and B. Mamalis, "Clustering of mobile ad hoc networks: an adaptive broadcast period approach", *IEEE International Conference on Communications ICC'06*, pp. 4034 – 4039, 2006.
- [97] J. Guo-xing and Y. Zhi-ya, "A distributed clustering algorithm based on δ -cluster stability for mobile ad hoc networks", *4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing WiCOM '08*, pp. 1 – 6, 2008.
- [98] J. An, C. Li and B. Li, "An improved weight based clustering algorithm in mobile ad hoc networks", *IEEE Youth Conference on Information, Computing and Telecommunication YC-ICT '09*, pp. 220 – 223, 2009.
- [99] F. D. Tolba, D. Magoni and P. Lorenz, "A stable clustering algorithm for highly mobile ad hoc networks", *2nd IEEE International Conference on Systems and Networks Communications ICSNC 2007*, 2007.
- [100] Y. Yi, T. J. Kwon and M. Gerla, "Passive clustering (PC) in ad hoc networks", *draft-yi-manet-pc-00.txt*, November 2001.

- [101] T. J. Kwon, M. Gerla, V. K. Varma, M. Barton and T. R. Hsing, "Efficient flooding with passive clustering—an overhead-free selective forward mechanism for ad hoc/sensor networks", *Proceedings of the IEEE*, vol. 91, issue 8, pp. 1210 – 1220, 2003.
- [102] Y. Yi, M. Gerla and T. J. Kwon, "Efficient flooding in ad hoc networks using on-demand (passive) cluster formation", *2nd Annual Mediterranean Ad Hoc Networking Workshop (Med-hoc-net 2003)*, 2003.
- [103] A. Ephremides, J. E. Wieselthier and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signalling", *Proceedings of the IEEE*, vol. 75, issue 1, pp. 56 – 73, 1987.
- [104] D. Gavalas, G. Pantziou, C. Konstantopoulos, and B. Mamalis, "Lowest-ID with adaptive ID reassignment: a novel mobile ad-hoc networks clustering algorithm", *1st IEEE International Symposium on Wireless Pervasive Computing*, 2006.
- [105] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network", *Journal of Wireless Networks*, vol. 1, pp. 255 – 265, 1995.
- [106] S. Basagni, "Distributed clustering for ad hoc networks", *4th IEEE International Symposium on Parallel Architectures, Algorithms, and Networks ISPAN'99*, pp. 310 – 315, 1999.
- [107] S. Adabi, S. Jabbehdari, A. Rahmani and S. Adabi, "A novel distributed clustering algorithm for mobile ad-hoc networks", *Journal of Computer Science 4(2)*, Science Publications, pp. 161 – 166, 2008.
- [108] A. B. McDonald and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1466 – 1487, 1999.
- [109] S. Adabi and S. Jabbehdari, "Distributed fuzzy score-based clustering algorithm for mobile ad hoc networks", *IEEE Asia-Pacific Services Computing Conference APSCC '08*, pp. 193 – 198, 2008.
- [110] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms", *IEEE Communications Magazine*, vol. 43, issue 7, pp. 78 – 85, 2005.
- [111] W. Choi and M. Woo, "A distributed weighted clustering algorithm for mobile ad hoc networks", *IEEE International Conference on Internet and Web Applications ICIW'06*, 2006.

- [112] L. Gao, D. Mu, Y. Wang, G. Zhang and L. Zhang, "A novel clustering algorithm for ad hoc network", *4th IEEE Conference on Industrial Electronics and Applications ICIEA*, pp. 445 – 448, 2009.
- [113] T. C. Huang, L. C. Shiu and H. C. Ke, "A double manager k-hop clustering algorithm in mobile ad hoc networks", *4th IEEE International Conference on Computer and Information Technology CIT'04*, pp. 640 – 645, 2004.
- [114] V. S. Anitha and M. P. Sebastian, "SCAM: Scenario-based clustering algorithm for mobile ad hoc networks", *1st IEEE International Conference on Communication Systems and Networks and Workshops COMSNETS 2009*, Bangalore, pp. 1 – 8, 2009.
- [115] A. D. Amis, R. Prakash, T. H. P. Vnong and D. T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks", *19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2000*, vol. 1, pp. 32 – 41, 2000.
- [116] J. Zhou, J. Chen, W. Xie and J. Li, "Improved weight clustering algorithm for IDS in mobile ad hoc network", *IEEE International Conference on Wireless Communications, Networking and Mobile Computing WiCOM 2007*, pp. 1565 – 1568, 2007.
- [117] C. Tselikis, C. Douligeris, S. Mitropoulos and N. Komninos, "Consistent re-clustering in mobile ad hoc networks", *IEEE International Conference on Wireless Communications and Mobile Computing IWCMC '08*, pp. 825 – 830, 2008.
- [118] Y. Zeng, J. Cao, S. Guo, K. Yang and L. Xie, "SWCA: a secure weighted clustering algorithm in wireless ad hoc networks", *IEEE Conference on Wireless Communications and Networking WCNC 2009*, pp. 1 – 6, 2009.
- [119] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265 – 1275, 1997.
- [120] S. Sivavakeesar, G. Pavlou and A. Liotta, "Stable clustering through mobility prediction for large-scale multihop intelligent ad hoc networks", *IEEE Conference on Wireless Communications and Networking WCNC 2004*, vol. 3, pp. 1488 – 1493, 2004.
- [121] Y. Zhang and J. M. Ng, "A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks", *IEEE International Conference on Communications ICC '08*, pp. 3161 – 3165, 2008.

- [122] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks", *12th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001*, vol. 2, pp. 1028 – 1037, 2001.
- [123] C. Tselikis, S. Mitropoulos, C. Douligeris, E. Ladis, K. Georgouleas, C. Vangelatos and N. Komninos, "Empirical study of clustering algorithms for wireless ad hoc networks", *16th IEEE International Conference on Systems, Signals and Image Processing IWSSIP 2009*, pp. 1 – 6, 2009.
- [124] I. Inn-Er and W. K. G. Seah, "Clustering overhead and convergence time analysis of the mobility-based multi-hop clustering for mobile ad hoc networks", *11th IEEE International Conference on Parallel and Distributed Systems*, vol. 2, pp. 130 – 134, 2005.
- [125] A. Akbari, A. Khosrozadeh and N. Lasemi, "Clustering algorithms in mobile ad hoc networks", *4th IEEE International Conference on Computer Sciences and Convergence Information Technology ICCIT '09*, pp. 1509 – 1513, 2009.
- [126] H. Shigeno, C. Chen, K. Arai and Y. Ohnishi, "New MAC schemes: CTMA and CRMA – One approach for hidden terminal problem", *12th Annual IEEE International Phoenix Conference on Computers and Communications*, pp. 326 – 331, 1993.
- [127] M. Gerla, G. Pei and S. Lee, "On-Demand Multicast Routing Protocol (ODMRP) for ad hoc networks", *IETF Internet draft: draft-ietf-manet-odmrp-00.txt*, November 1998.
- [128] S. Lee, W. Su and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for ad hoc networks", *IETF Internet draft: draft-ietf-manet-odmrp-02.txt*, January 2000.
- [129] H. Dhillon and H. Q. Ngo, "CQMP: a mesh-based multicast routing protocol with consolidated query packets", *IEEE Conference on Wireless Communications and Networking*, vol. 4, pp. 2168 – 2174, 2005.
- [130] Z. Yao, X. Leiming and S. Meilin, "On-demand multicast routing protocol with multipoint relay (ODMRP-MPR) in mobile ad hoc network", *IEEE International Conference on Communication Technology ICCT 2003*, vol. 2, pp. 1295 – 1300, 2003.
- [131] H. Peng, L. Yuanan, M. Xiaolei, and C. Zhe, "Enhanced ODMRP using conditioned broadcasting algorithm", *ISECS International Colloquium on Computing, Communication, Control and Management CCCM 2009*, vol. 3, pp. 140 – 143, 2009.

- [132] A. Ganguli, S. Nandi, and P. K. Meher, "An improvement of ODMRP for reliable delivery of bursty traffic", *IEEE Conference on Convergent Technologies for Asian-Pacific Region TENCON 2003*, vol. 3, pp. 1067 – 1071, 2003.
- [133] M. Naderan-Tahan, A. Darehshoorzadeh and M. Dehghan, "ODMRP-LR: ODMRP with link failure detection and local recovery mechanism", *8th IEEE International Conference on Computer and Information Science ICIS 2009*, pp. 818 – 823, 2009.
- [134] H. Ying-xin and J. Yu-feng, "Improvement of wireless multicast routing with energy-efficiency based ODMRP", *IEEE International Symposium on Computer Network and Multimedia Technology CNMT2009*, pp. 1 – 4, 2009.
- [135] S. J. Begdillo, H. Mohamamdzadeh, S. Jamali, and A. Norouzi, "Stable route selection in ODMRP with energy based strategy", *21st IEEE International Symposium on Personal Indoor and Mobile Radio Communications PIMRC 2010*, pp. 1741 – 1745, 2010.
- [136] Y. Yao, Y. Zhou and D. Sidan, "Providing interference-aware quality of service support for ODMRP", *IEEE Conference on Information Processing APCIP 2009, Asia-Pacific*, vol. 2, pp. 618 – 622, 2009.
- [137] A. Darehshoorzadeh, M. R. J. Motlagh, and M. Dehghan, "Quality of service support for ODMRP unicast routing in ad hoc networks", *IEEE International Conference on Digital Telecommunications ICDT '06*, 2006.
- [138] H. Tebbe and A. J. Kassler, "QAMNet: Providing quality of service to ad-hoc multicast enabled networks", *1st IEEE International Symposium on Wireless Pervasive Computing*, pp. 1 – 5, 2006.
- [139] J. Biswas and S. K. Nandy, "Quality of support for multicasting over mobile ad hoc networks", *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 220 – 225, 2005.
- [140] K. Zhang, H. Wang and F. Liu, "Multicast routing for delay and delay variation bounded Steiner tree using simulated annealing", *IEEE Conference on Networking, Sensing and Control*, pp. 682 – 687, 2005.
- [141] S. Aggarwal, M. Limaye, A. Netravali and K. Sabnani, "Constrained diameter Steiner trees for multicast conferences in overlay networks", *1st IEEE International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks QSHINE '04*, pp. 262 – 271, 2004.

- [142] P. Agrawal, R. K. Ghosh and S. K. Das, "Localization of wireless sensor nodes using proximity information", *16th IEEE International Conference on Computer Communications and Networks ICCCN 2007*, pp. 485 – 490, 2007.
- [143] N. Swangmuang and P. Krishnamurthy, "Location fingerprint analyses toward efficient indoor positioning", *6th Annual IEEE International Conference on Pervasive Computing and Communications PerCom 2008*, pp. 100 – 109, 2008.
- [144] M. Kj_rgaard, "Indoor positioning with radio location fingerprinting," *PhD dissertation*, Department of Computer Science, University of Aarhus, Denmark, April 2010.
- [145] Wikipedia (online), http://en.wikipedia.org/wiki/Algorithms_for_calculating_variance, Last accessed September 2011.
- [146] J. Nuevo, "A comprehensible Glomosim tutorial", *INRS-Université du Québec*, 2003.
- [147] P. R. J. Ostergard, "A fast algorithm for the maximum clique problem", *Discrete Applied Mathematics 120*, Elsevier Science B.V., pp. 197 – 207, 2002.
- [148] A. Penttinen and J. Virtamo, "Improving multicast tree construction in static ad hoc networks", *28th Annual IEEE International Conference on Local Computer Networks LCN'03*, pp. 762 – 765, 2003.
- [149] A. S. Akyurek and E. Uysal-Biyikoglu, "A low complexity distributed algorithm for computing minimum-depth multicast trees in wireless networks", *IEEE Conference on Military Communications MILCOM 2010*, pp. 1918 – 1923, 2010.
- [150] B. H. Liu, M. J. Tsai, and W. C. Ko, "Dynamical construction of a core-based group-shared multicast tree in mobile ad hoc networks", *19th IEEE International Conference on Advanced Information Networking and Applications AINA 2005*, vol. 1, pp. 90 – 95, 2005.
- [151] A. Penttinen, "Efficient multicast tree algorithm for ad hoc networks", *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 519 – 521, 2004.
- [152] S. K. S. Gupta and P. K. Srimani, "Adaptive core selection and migration method for multicast routing in mobile ad hoc networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, issue 1, pp. 27 – 38, 2003.
- [153] N. Skorin-Kapov and M. Kos, "The application of Steiner trees to delay constrained multicast routing: a tabu search approach", *7th IEEE International Conference on Telecommunications ConTEL 2003*, vol. 2, pp. 443 – 448, 2003.

- [154] J. Konc and D. Janezic, "An improved branch and bound algorithm for the maximum clique problem", *MATCH Communications in Mathematical and in Computer Chemistry (Journal)*, vol. 58, ISSN 0340-6253, pp. 569 – 590, 2007.
- [155] Y. Shinano, T. Fujie, Y. Ikebe, and R. Hirabayashi, "Solving the maximum clique problem using PUBB", *1st IEEE Merged International ... and Symposium on Parallel and Distributed Processing IPSP/SPDP 19998*, pp. 326 – 332, 1998.
- [156] D. Kumlander, "A simple and efficient algorithm for the maximum clique finding reusing a heuristic vertex colouring", *IADIS International Journal on Computer Science and Information Systems*, vol. 1, no. 2, ISSN 1646-3692, pp. 32 – 49, 2006.
- [157] T. Ohta, S. Inoue, Y. Kakuda, K. Ishida and K. Maeda, "An adaptive maintenance of hierarchical structure in ad hoc networks and its evaluation", *22nd IEEE International Conference on Distributed Computing Systems Workshops ICDCSW'02*, pp. 7 – 13, 2002.
- [158] A. R. Hussein, S. Yousef, S. Al-Khayatt and O. S. Arabeyyat, "An efficient weighted distributed clustering algorithm for mobile ad hoc networks", *IEEE International Conference on Computer Engineering and Systems (ICCES)*, pp. 221 – 228, 2010.
- [159] R. A. Meyer and R. Bagrodia, "PARSEC user manual for PARSEC release 1.1", *Parallel Computing Laboratory, Computer Science Department, University of California, Los Angeles*, September 1999.