Group communications in ad hoc networks

Zhang, Yan

2011

Zhang, Y. (2011). Group communications in ad hoc networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46321>

<https://doi.org/10.32657/10356/46321>

**GROUP COMMUNICATIONS IN AD HOC
NETWORKS**

ZHANG YAN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2011

To My Parents

Acknowledgements

First of all, I would express my gratitude to Nanyang Technological University for providing me the opportunity to carry on my research work.

I would like to give my sincerely and deepest thanks to my supervisors, Dr. Low Chor Ping, and Dr. Ng Jim Mee for their helpful guidance, careful mentoring and great supports during the course of my PhD candidature, especially for his patience of Dr. Low in reviewing my thesis and his constructive advices. Their broad horizon, precise mind and unlimited enthusiasm in research benefit me a lot.

Especially, I would like to thank my parents and all of my friends for their dedication, encouragement, and support on me during the period of study.

Last but not least, I would thank all the staffs and students in the Division of Communications and the Infocomm Research Lab, for their kind helps in my research work.

TABLE OF CONTENTS

Acknowledgements	i
Table of Contents	iii
List of Figures.....	vii
List of Tables	x
Summary.....	xi
Chapter 1: Introduction	1
1.1 Background	1
1.2 Motivation	3
1.3 Objectives and Contributions	7
1.3.1 Group Mobility Modeling	7
1.3.2 Network Performance Analysis	8
1.3.3 Clustering Algorithms for Network Topology Control.....	9
1.3.4 Group Partition Prediction	11
1.4 Thesis Organization	12
Chapter 2: Reference Region Group Mobility Model	13
2.1 Introduction	13
2.2 Literature Review	15
2.2.1 Entity Mobility Models	15
2.2.2 Group Mobility Models	16
2.3 An Overview of the Reference Region Group Mobility Model	18
2.3.1 Mobility of a Group Assigned with a Single Destination	18
2.3.2 Mobility of a Group Assigned with Multiple Destinations.....	22
2.3.3 Merger of Groups	23

2.4 Model Configurations and Implementations.....	25
2.5 Node's Movement and Displacement of a Reference Region	30
2.6 The Merger of a Standby Group with Another Group.....	33
2.7 Group Partitions and Mergers with Predefined Destinations.....	33
2.8 Simulation and Discussion	36
2.8.1 Generic Group Mobility Pattern in MANETs.....	36
2.8.2 Group Mobility with Predefined Destinations	40
2.8.3 Entity Mobility	42
2.9 Conclusion.....	46

Chapter 3: Performance Evaluation Of Network Routing Protocols Using

RRGM.....	47
3.1 Introduction	47
3.2 Review of Network Routing Protocols	48
3.2.1 Table Driven Routing Protocols.....	48
3.2.2 On-demand Routing Protocols	52
3.2.2.1 Ad Hoc On-demand Distance Vector Routing (AODV)....	53
3.2.2.2 Dynamic Source Routing (DSR)	54
3.2.2.3 Comparison between AODV and DSR	56
3.2.3 Comparison between Table Driven Routing Protocols and On-demand Routing Protocols.....	57
3.3 Network Environment Configurations.....	59
3.3.1 Experimental Settings for Investigation on Node Speeds.....	60
3.3.1.1 Scenario I - Group Mobility with Group Partition Disabled	61
3.3.1.2 Scenario II - Group Mobility with Group Partition Enabled	61
3.3.2 Experimental Settings for Investigation on Group Densities.....	62
3.4 Performance Evaluation of Routing Protocols Based on RRGM	62
3.4.1 Performance Metrics	63
3.4.1.1 Packet Delivery Ratio (PDR)	63
3.4.1.2 Average Control Packets per Data Packet Sent (ACP)	63
3.4.1.3 End-to-end Delay (EED)	64
3.4.2 Simulation Results.....	65
3.4.2.1 Varying the Average Speed.....	65
3.4.2.1.1 Packet Delivery Ratio vs. Average Speed	65

3.4.2.1.2	Average Control Packets per Packet Sent vs. Average Speed	69
3.4.2.1.3	End-to-end Delay vs. Average Speed	71
3.4.2.2	Varying the Node Density	74
3.4.2.2.1	Packet Delivery Ratio vs. Node Density	74
3.4.2.2.2	Average Control Packets per Data Packet Sent vs. Node Density	75
3.4.2.2.3	End-to-end Delay vs. Node Density	76
3.4.3	Discussion	77
3.5	Conclusion.....	79
Chapter 4: Distributed Group Mobility Adaptive Clustering Algorithm		80
4.1	Introduction	80
4.2	Literature Review	84
4.2.1	Clustering Algorithms in MANETs	84
4.2.2	Spatial Dependency	87
4.3	Mobility Metric - Linear Distance Based Spatial Dependency	88
4.4	Distributed Group Mobility Adaptive Clustering Algorithm.....	91
4.4.1	The Nomination Process	94
4.4.2	The Maintenance Process.....	96
4.4.3	Analysis of the DGMA Algorithm.....	99
4.5	Simulation and Discussions on DGMA under Group Mobility Pattern ..	102
4.5.1.	Mobility Model	103
4.5.2.	Configurations of DGMA	104
4.5.3.	Performance Evaluations of DGMA	105
4.5.3.1	Varying Transmission Ranges.....	107
4.5.3.2	Varying Speeds.....	111
4.5.3.3	Varying Partition Intervals	115
4.5.3.4	Varying Node Densities	119
4.6	Performance Evaluation of DGMA Using Entity Mobility Model.....	122
4.6.1	Varying Transmission Ranges	123
4.6.2	Varying Speeds	125
4.7	Conclusion.....	127

Chapter 5: A Framework for Predicting Group Partitions in MANETs.....	128
5.1 Introduction	128
5.2 Review of Partition Prediction Algorithms.....	133
5.3 Group Partition Prediction	134
5.3.1. Group Mobility Generator.....	134
5.3.2. Clustering Algorithm.....	135
5.3.3. Gathering Group Mobility Information.....	136
5.3.4. Mobility Prediction --- Quadratic Regression Trend Model	140
5.3.5. Group Safe Distance Analysis.....	144
5.3.6. Group Partition Time Estimation	147
5.3.7. Prediction Result Evaluation	149
5.4 Simulation	149
5.5 Discussion on the Impact of Clustering Schemes on the Prediction Results... ..	157
5.6 Conclusion.....	158
 Chapter 6: Conclusions and Future Work	160
6.1 Conclusions	160
6.2 Recommendation for Future Work	163
 Appendix-A.....	165
 Author's Publications	167
 Bibliography	169

LIST OF FIGURES

Figure 2.1: Mobility of a group assigned with a single destination.....	21
Figure 2.2: Mobility of a group assigned with multiple destinations	22
Figure 2.3: Merger of two groups	24
Figure 2.4: Illustration of a reference region	28
Figure 2.5: Snapshot of displacement of reference regions by adjusting d_x and d_y	32
Figure 2.6: Illustration of a group passing a "hidden" destination.....	35
Figure 2.7: Generic group mobility pattern with group partitions and mergers	39
Figure 2.8: Group mobility inside a building (e.g. an exhibition hall)	42
Figure 2.9: An example of Random Waypoint Mobility model implemented by the RRGGM model	45
Figure 2.10: The movement trajectory of a mobile node in Random Waypoint Mobility generated by RRGGM	45
Figure 3.1: The routing table of the node B in DSDV	49
Figure 3.2: The routing table of node B after node D joins in the network in DSDV	50
Figure 3.3: Propagation of the RREQ in AODV	53
Figure 3.4: Propagation of the RREP in AODV	54
Figure 3.5: Propagation of the RREQ in DSR.....	55
Figure 3.6: Propagation of the RREP in DSR.....	56
Figure 3.7: Packet delivery ratio vs. Average speed (m/s) --- group partition disabled	65
Figure 3.8: Packet delivery ratio vs. Average speed (m/s) --- group partition enabled	66

Figure 3.9: Average control packets per packet sent vs. Average speed --- group partition disabled	69
Figure 3.10: Average control packets per packet sent vs. Average speed --- group partition enabled	70
Figure 3.11: End-to-end delay vs. Average speed --- group partition disabled	72
Figure 3.12: End-to-end delay vs. Average speed --- group partition enabled	72
Figure 3.13: Packet delivery ratio vs. Node density --- group partition enabled	74
Figure 3.14: Average control packets per data packet sent vs. Node density --- group partition enabled	76
Figure 3.15: End-to-end delay vs. Node density --- group partition enabled	77
Figure 4.1: Examples of clusters in different node densities formed by DGMA	92
Figure 4.2: Protocol specification of the <i>nomination process</i>	95
Figure 4.3: Protocol specification of maintenance process	98
Figure 4.4: Performance metrics vs. Transmission ranges	107
Figure 4.5: Performance metrics vs. Speeds	112
Figure 4.6: Performance metrics vs. Group partition intervals.....	115
Figure 4.7: Performance metrics vs. node densities	119
Figure 4.8: Performance metrics vs. Transmission range with RWP mobility	123
Figure 4.9: Performance metrics vs. Speed with RWP mobility	125
Figure 5.1: Architecture of group partition prediction system	129
Figure 5.2: Procedure of gathering group mobility information.....	137
Figure 5.3: Example of group <i>macro movement</i>	138
Figure 5.4: Example of the safe distance between two groups.....	145
Figure 5.5: Illustration of partition time estimation.....	147
Figure 5.6: Example of group partitions generated by RRGM.....	150
Figure 5.7: Prediction Accuracy, $\rho=0.5$ node/(100m) ² , Tr=100m.....	152
Figure 5.8: Prediction Accuracy, $\rho=1$ node/(100m) ² , Tr=100m.....	153

Figure 5.9: Prediction Accuracy, $\rho=1 \text{ node}/(100\text{m})^2$, $\text{Tr}=150\text{m}$	154
Figure 5.10: Prediction Accuracy, $\rho=1 \text{ node}/(100\text{m})^2$, $\text{Tr}=200\text{m}$	154
Figure 5.11: Prediction deviation, $\rho=1 \text{ node}/(100\text{m})^2$	155
Figure 5.12: Prediction deviation, $\text{Tr}=100\text{m}$	156
Figure 5.13: Estimation error with respect to clustering schemes	158

LIST OF TABLES

Table 2.1: Input parameters of RRGM	25
Table 2.2: Input parameters for the simulation of generic group mobility pattern	38
Table 2.3: Input parameters for the simulation of group mobility with predefined destinations	41
Table 2.4: Input parameters for modeling Random Waypoint Mobility	44
Table 3.1: Comparison of table-driven and on-demand routing protocols	58
Table 3.2: Configurations of network environment.....	59
Table 4.1: Mobility parameters	103
Table 4.2: Clustering parameters	105
Table 5.1 Prediction Parameters	151

Summary

A mobile ad hoc network (MANET) is an autonomous wireless mobile network that does not rely on any fixed infrastructure or centralized administration, and its distant nodes communicate through multi-hop paths. With the increasing need of group communications in wireless mobile network applications, efficient support of it is a critical issue. This thesis focuses on the research of four key areas of group communications of MANETs, namely group mobility modeling, network performance evaluations, clustering for topology control, and group partition prediction.

The presence of group partitions is a prevalent mobility behavior in most group communication applications of MANETs, which damages network interconnections and induces communication interruptions. The phenomenon of group partitions can be described with mobility models. Mobility models represent the movement of mobile users, and how their location, velocity and acceleration change over time. However, to the best of our knowledge, existing mobility models for mobile ad hoc networks are not able to describe group partitions and group mergers. To address this issue, we proposed a new mobility model, namely the *Reference Region Group Mobility* model (RRGM), which can simulate both individual's and group's mobility in MANETs. Based on this model, group partitions and mergers can be mimicked.

Another issue is the degradation of network performance in group communications when group partitions take place. Group partitions can cause communication interruptions since established routing paths are broken. It results

in massive packet loss and higher end to end delay for packet delivery as more time is spent for packet senders to reestablish alternative routes to destinations. In this thesis, we conduct a comprehensive network performance evaluation of *Ad-hoc On-Demand Distance Vector Routing* (AODV) protocol and *Dynamic Source Routing* (DSR) protocol and study the impact that group partition has on the performance of these routing protocols.

Topology control in dynamic group mobility environment is another major concern in MANETs since the network structure of an ad hoc network changes dynamically. However, most clustering algorithms proposed for MANETs are not adaptive to mobility or work under the assumption of rare mobility. So they may not have effective schemes to support fast changes in network topology. To address this, a *Distributed Group Mobility Adaptive* (DGMA) clustering algorithm is proposed. DGMA is a mobility-aware clustering algorithm which is suitable for dynamic network environment, e.g. in MANETs. A feature of this algorithm is the mechanism to use a novel mobility metric, namely *Linear Distance Based Spatial Dependency*, to represent the mobility characteristics of each individual mobile node.

Group partition prediction is a very challenging issue in group communications of MANETs. A group partition prediction algorithm is used to predict partitions of a group into subgroups. Given that the mobility prediction methods for cellular communication networks are not applicable to MANETs, we propose a framework for predicting group partitions in MANETs by integrating our works of the mobility model RRGm and the clustering algorithm DGMA. In this framework, mobility of a group is predicted with quadratic regression model based on group's historical

mobility information. Experimental results show that the proposed scheme is effective in group partition prediction.

CHAPTER 1: INTRODUCTION

1.1 Background

The work presented in this thesis is motivated by recent advances in mobile computer networks and wireless communications. During the last two decades, wireless networking technologies emerged with the development and integration of personal computing, cellular communication networks, and the Internet. Due to the increasing interaction between communication and personal computing, wireless access is changing from “sometime, somewhere” to “anytime, anywhere”. Studies of the evolution of mobile technology suggest that mobile devices will continue to be increasingly functional, often termed as ubiquitous or pervasive computing, while the reliance on fixed infrastructures decreases [1].

Mobile ad hoc network (MANET) is one of the emerging areas in mobile communications [2]. A MANET is an autonomous wireless mobile network that does not rely on any fixed infrastructure, and mobile devices are connected via wireless links [3] that enable people and devices to seamlessly communicate with each other even in areas without pre-existing communication infrastructures.

The development of ad hoc networks is an important milestone in the evolution of wireless communications. During the late of 1970s, the deployment of MANETs

was initially proposed by the US military, which allows soldiers, tanks, aircrafts to form a temporary wireless communication network [4]. With the growth of laptops and 802.11/Wi-Fi wireless networking technologies from the middle of 90's, MANETs became a popular research topic for its usages in civil areas. The interest in such networks is motivated from their ability to provide a temporary wireless networking capability where fixed infrastructures are lacking or infeasible to deploy. MANETs have a variety of applications where infrastructures are either unavailable, or untrusted, or destroyed in emergent situations, such as military and battlefields operations; disaster recovery; scientific explorations in wild lands; and infrastructure-less network in conferencing occasions.

In a MANET, every individual device is responsible to dynamically discover other devices that it can directly communicate with. Every device will forward traffic unrelated to its own usage behaving like a router. The communications in MANETs could be directly one-hop between wireless devices which are within short distance of each other, or multi-hop communications that forward messages via intermediate devices when the receivers are located beyond the transmission range of the sender. Every device in a MANET moves arbitrarily and therefore may change its connections to other devices frequently [5]. A distinct characteristic of MANETs which differentiates it from cellular networks is the rapid change in network topology due to node mobility. The natures of MANETs introduce many new research problems, such as network mobility management, device discovery, topology control, and self-organized routing.

In the recent years, MANETs have become an extensive research topic in the domain of wireless communication networks. Many research papers are published

with various specific research emphasis, which includes mobility management, routing protocol evaluation, cross-layering design, security issues and energy efficiency. While wide deployment of MANETs has not been realized yet, many efforts are currently underway for the network operations and management of such networks [6].

In summary, mobile ad hoc networks allow the construction of flexible and adaptive networks without the aid of fixed infrastructures. This kind of networks are expected to play an important role in the future generations of wireless communication systems. Future wireless technologies will require the highly-adaptive mobile networking technology to effectively support multi-hop communications and efficiently manage ad hoc network clusters, which will not only operate autonomously but also will be able to attach to worldwide networks, such as the Internet.

1.2 Motivation

The nature of dynamic network topology differentiates MANETs from conventional cellular mobile networks. The dynamic network topology makes it very challenging in the design of network routing protocols and cross-layering applications since the connections in ad hoc networks may be interrupted anytime and anywhere due to movement of devices. The need to discover mobility patterns of mobile devices in MANETs is essential to the study in many other areas, such as mobility management, network performance and applications. In the studies of MANETs, mobility patterns of mobile nodes can be described by mobility models. In general, mobility models for MANETs are classified into two categories, namely entity mobility models and group mobility models [7]. Entity mobility models are

used to simulate the mobility patterns of each individual node, while group mobility models can mimic the movement of mobile groups and mobile nodes in the same group may move with a similar mobility pattern.

With respect to the group mobility pattern, providing pervasive and ubiquitous group communications is one of the major tasks for the evolution of MANETs. Group communication implies a many-to-many communication style among various mobile groups. It has demonstrated its importance in many MANET applications ranging from battlefield operations, disaster recovery and scientific exploration, to emergency response. It is common for mobile devices to be clustered into groups according to their functionalities. For example, in the search and rescue operation, rescue teams may be composed of doctors, nurses, firemen, technicians, and soldiers. In a battlefield scenario, a troop can be composed of soldiers, vehicles, tanks, and helicopters. Different roles can form a functional group and a network may be composed by a great number of such groups. However, most previous works in MANETs are based on random node mobility (entities' mobility) which is unrealistic and too simple to reflect the actual effectiveness of such applications as introduced before. Lack of comprehensive research in group mobility will delay the development of MANET applications for group communications. Thus, research on group mobility and network connectivity is required as insufficient research is carried out in this area.

In addition, group partition and merger are the common phenomena in group mobility in MANETs, which make the topic of group communications more challenging. In most group related applications of MANETs, network partitions may take place anytime and anywhere that will damage the intergroup

communications and even the communications of entire network. The performance of network routing is greatly affected. In the research areas of MANETs, mobility pattern is the fundamental issue to be considered, which is distinct from the methodology in the studies of conventional networks. Studying group mobility behaviors, it must rely on group mobility models. However, a generic and realistic group mobility model is lacking. Hence, a mobility model will be addressed in the first part of this thesis, which can be used to describe the mobility patterns of both groups' and entities'. More complex operations among groups, such as group partitions and mergers, can be simulated with this model. With this model, other extended research topics, such as investigation on performance of a network in group mobility, network topology control, and group partition predictions, can be carried out.

Network routing is an important and the most actively researched topic in MANETs. A network routing protocol is used to select routes for exchanging data packets among network users. In MANETs, nodes start out without any knowledge of the network structure; hence, they have to discover it themselves using routing algorithms. Every node in the network is required to periodically broadcast a *'hello'* message to announce its presence to all of its neighbors. Based on the neighborhood information of each node, routes to connect sources and destinations can be established. The performance of a network routing protocol change greatly under different mobility patterns [8-13]. In most of the existing works for investigating the performance of network routing protocols in MANETs, entity mobility models are used to generate the mobility environment for simulation studies. It is insufficient to demonstrate all characteristics of a routing protocol. For the second part of this thesis, the performance of two typical routing protocols in MANETs

under group mobility will be examined by using our proposed mobility model, namely the Reference Region Group Mobility model (RRGM), and the impact of group partitions to their performance will be discussed. Extensive simulation are carried out to obtain better insight on the performance of these routing protocols under different network environments.

With respect to the trend of MANET development, large-scale MANETs will be one of the future directions. Network complexity will be increased by the increasing number of mobile devices. One of the solutions to manage a large-scale network is to cluster it into a hierarchical network structure in order to facilitate network operations. A hierarchical network topology is the fundamental of network management. Clustering algorithms are very important for topology control in MANETs. To support group communications in a MANET, it is necessary to develop a clustering algorithm that can identify groups that could be managed by clusterheads. In the third part of this thesis, a novel mobility-aware clustering algorithm for group mobility in MANETs will be developed by utilizing the metric of relative mobility between mobile nodes. This clustering algorithm can form new clusters dynamically when group partitions take place in group mobility environment.

In group mobility of MANETs, group partition is an undesirable behavior as it may destroy the established links among mobile nodes which may cause communication interruptions. In recent years, some preliminary works in link availabilities and network partitions have been studied [14-24]. However, most of them confined their usages within very specific situations and none of them can be used in predicting group partitions. To address this concern, we will propose a

framework to systematically study the methods for group partition predictions in the fourth part of this thesis. This framework is composed of a number of modules, which include mobility models, group mobility measurement, clustering algorithms, methods to identify adjacent groups, prediction algorithms, partition time estimation, and data process for prediction results. In practical applications, such a partition prediction framework can help to prevent group partitions from taking place if remedial actions can be carried out in advance. For example, in a battlefield, a commander can request a helicopter to be deployed as an intermediate node between separated troops over the operational zone, or to deploy micro air vehicles such as a relay robot between the partitioning groups, whose position can be adjusted with respect to the distance between the partitioning groups in order to maintain their connectivity.

1.3 Objectives and Contributions

The main objective of this thesis is to investigate the mobility behaviors of groups in mobile ad hoc networks that addresses the four major issues as introduced in the previous section, i.e. group mobility modeling, network performance analysis, clustering algorithms for network topology control and group partition prediction schemes.

1.3.1 Group Mobility Modeling

As discussed in the earlier sections, group partitions and mergers are inherent phenomena in most group communication applications of MANETs. To study the characteristics and behaviors of group mobility in MANETs, it is necessary to develop a group mobility model that can describe more realistic group mobility

behaviors. Although in some existing group mobility models, group leaders are identified before running a simulation, it is not reasonable to predefine group leaders in group partitioning scenarios because the number of groups, as well as group leaders, can not be fixed. Instead, they change when group partitions and mergers take place. Hence our research objective is to design a group mobility model to mimic group partitions and mergers. Mobility of nodes is managed in a distributed manner by themselves without any central controller or group coordinator.

To address this issue, the Reference Region Group Mobility model (RRGM) is proposed. The contributions of this model are its integration of group mobility and entity mobility within this model and it is able to mimic group operations, such as group partitions and mergers, which are common phenomena in group mobility of MANETs. Another improvement of this model is that unlike existing models, it is not necessary to predefine group leaders.

1.3.2 Network Performance Analysis

In MANETs, group mobility is more complex than entity mobility as it may have various different mobility patterns for nodes and groups. Network performance of routing protocols would be affected in group mobility environment due to the factors of mobility speeds, number of nodes in a group, and their connectivities. Another concern for group mobility is that mobile groups may partition into smaller subgroups which may further merge together at a later stage. A frequent change of group membership could destroy existing routes and even cause a vast communication interruption across the network. Unfortunately, existing mobility models for evaluating routing protocols are either too simple, such as the Random

Waypoint Mobility model and the Random Direction Mobility model [25], or very synthetic by restricting users' mobility within a predefined graph, i.e., movement is restricted on predefined paths [26-28]. They are not suitable to be used for the study of the impact of group partitions to network routing protocols. Therefore, our research objective on this issue is to evaluate routing protocols and investigate the degradation of network performance when group partition takes place by using a generic group mobility model to simulate group partitions.

We set up the mobility environment of the simulation with our proposed RRGM model in order to study the impact of group partitions to the performance of network routing protocols. A comprehensive network performance comparison between two well-known routing protocols, namely Ad-hoc On-Demand Distance Vector Routing protocol (AODV) [29] and Dynamic source routing protocol (DSR) [30] are conducted in respect to node speed and node density. Their performance is evaluated in different experiments where group partition is enabled or disabled. The impact of group partitions to routing protocols is presented in the aspects of packet delivery ratio, control packet overhead and end-to-end delay.

1.3.3 Clustering Algorithms for Network Topology Control

Network complexity is proportional to the size of a mobile ad hoc network. Clustering, as a key method in network management, can transform a large-scale MANET into a hierarchical structure so as to facilitate network operations. From our research, most of the existing clustering algorithms for MANETs and sensor networks work under the assumption of rare mobility or not mobility-aware. Our research objective is to develop a clustering algorithm to support group mobility in MANETs. Instead of the assumption of a static network environment, we consider a

more complex network environment of group mobility with dynamic topology changes that allow group partitions and mergers. This clustering algorithm is expected to ensure a better cluster stability in terms of longer cluster lifetime and mean residence time of cluster members, and less cluster re-affiliation events.

We proposed a clustering algorithm, namely Distributed Group Mobility Adaptive (DGMA) clustering algorithm, for topology control in group mobility of MANETs. The first feature of this algorithm is that a novel mobility metric, namely Linear Distance Based Spatial Dependency (LDSD), is proposed to measure group movement. This mobility metric tackles micro movement or pendulum-like movement which has little contribution in group location changes and should be ignored. The second feature is that it is adaptive to group partitions and mergers and to the best of our knowledge, these issues have never been addressed to date. The third feature is that it is a distributed algorithm which is executed at each individual node with the knowledge of its local information. Every node has an equal opportunity to be elected as a leader and can join or leave a cluster anytime without any restriction. The next feature is that it is a real-time algorithm since it discards a common assumption that nodes are not moving while clustering formation is in progress [49]. Lastly, it uses a simple coloring approach for the formation of clusters. Different colors represent roles of clusterheads and members. This methodology can be widely used in most of clustering schemes. Simulation results show that this clustering algorithm improve cluster stability significantly in terms of various performance metrics which include the following: mean lifetime of clusterhead, mean resident time of cluster members, number of clusterhead changes and number of cluster reaffiliations.

1.3.4 Group Partition Prediction

A major concern of group mobility in MANETs is that group partitions may cause link disruptions which may result in severe communication interruption across the entire network. However, we note that maintaining universal connectivity in a MANET is theoretically possible but not necessary. By the presence of group partition prediction algorithms and other schemes for group mobility in MANETs, it is possible to prevent group partitions from taking place if certain recovering actions can be performed in advance, such as deploying communication sensors to maintain connectivity. The aim here is to propose a framework for predicting group partitions in MANETs. Based on this framework, systems that can support continuous group communications could be developed in the future.

This framework is composed of a number of modules, including mobility model, clustering algorithms, group mobility measurement, groups' connectivity, mobility prediction algorithms, group partitioning time estimation, and data process for prediction results. The idea of *macro movement* of a group is introduced, and it is used to improve the accuracy of prediction results. Furthermore, analysis and discussion on safe distance between groups are conducted. Quadratic regression model is used to predict movement trajectory of groups. In addition, extended research on the impact of clustering algorithms to the prediction results is carried out. From the experimental results, this framework is effective and efficient in predicting group partitions in MANETs.

1.4 Thesis Organization

This chapter gives an introduction to the wireless ad hoc network and identifies four major issues pertaining to group mobility in MANETs to be addressed, setting the direction and objectives for this thesis. In Chapter 2, we propose a novel group mobility model for mobile ad hoc network including its various applications. Performance evaluation of routing protocols based on the proposed group mobility model will be discussed in Chapter 3. Chapter 4 introduces our proposed group adaptive clustering algorithm. A comprehensive comparison with other clustering algorithms will be discussed. A group partition prediction framework which integrates our proposed mobility model and clustering algorithm with a new prediction algorithm is described in Chapter 5. In the final chapter, conclusions are made regarding the research work done and recommendations for future research work will be discussed.

CHAPTER 2: REFERENCE REGION GROUP MOBILITY MODEL

2.1 Introduction

Mobility models are used in simulation studies to describe the dynamic behaviors of mobile devices for analyzing and evaluating the performance of ad hoc network protocols under various scenarios [7]. Mobility models, which mimic the movement of mobile nodes in the real world, play a significant role in the development of MANETs. It has been shown that the mobility model chosen in a simulation study has a significant effect on the performance of the ad hoc network protocols under investigation [8, 11, 13, 31]. So choosing a suitable mobility model is crucial so that it is able to precisely present the scenario being simulated. Most existing mobility models, such as the Random Waypoint Mobility model [25, 32] and the Random Walk Mobility model [33], are designed to simulate the movement of each individual. However, with the emergence of group-oriented applications, several group mobility models have been recently proposed. One of the commonly

used group mobility models is the Reference Point Group Mobility (RPGM) model [34]. Another recent work on group mobility model is the Reference Velocity Group Mobility (RVGM) model [35], which is a variant of RPGM model but uses group velocity as the reference factor.

However, among all the existing group mobility models, none of them can reflect group partitions and group mergers which are very important in many practical scenarios. Moreover, every model has its own restrictions in applications. For example, in some mobility models group membership is fixed, or velocity is not allowed to change, or the paths of group's movement are predefined by users before running the simulation. By considering these restrictions, most of existing models are not able to describe the behaviors of group mobility realistically.

In this chapter, we will describe our proposed mobility model, which is referred to as the Reference Region Group Mobility (RRGM) model. As introduced in Chapter 1, this model is a generic and parameterized mobility model which is able to model both group's and individual's movement. The novelty of this model is its ability to mimic group operations, such as group partitions and mergers. The applications which require group mobility can be found in various scenarios which include military operations, search and rescue in disaster recovery, visiting an exhibition hall, and firefighters operating in a building. The common characteristic of the above applications is that mobile nodes can be organized in the unit of groups, which could be further partitioned into many subgroups or merged with other groups. In addition, we introduce the concept of node density of a group. With a fixed number of group members, node density can be used to control the coverage area of a group(the range of an area that group members move within). Unlike

existing group mobility models, RRGM allows mobile nodes to move independently without relying on the coordination of group leaders. By taking advantage of this, group partitions and mergers become possible.

2.2 Literature Review

Mobility models proposed for ad hoc networks can be classified into two categories in general: entity mobility models and group mobility models [7]. Entity mobility models attempt to mimic the movement of individual nodes and assume all nodes move in an independent manner, e.g. [36-47]. Group mobility models are used to simulate group's movement and mobility of every group member is related to each other, e.g. [48-52].

2.2.1 Entity Mobility Models

The Random Waypoint Mobility model (RWP) [25, 32] is a well-known entity mobility model. In this model, each mobile node randomly chooses a point as the destination and moves towards it with a randomly selected speed which is uniformly distributed in a range of $[V_{min}, V_{max}]$. After reaching the destination, the node may be stationary for a moment before generating a new destination. This process is repeated until the simulation ends.

Random walk mobility was first introduced in 1905 to describe the mobility of entities in nature. In particular, such entities move in extremely unpredictable ways, for example, molecules which move in random directions that do not yield a pattern or sense of direction [33]. To simulate this kind of mobility pattern, the Random Walk Mobility model was developed [53, 54] in an attempt to mimic the erratic movements of certain objects. Specifically, in the Random Walk Mobility model, a

host moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from pre-defined ranges, $[speed_{min}, speed_{max}]$ and $[0, 2\pi]$ respectively. Each movement in the Random Walk Mobility model occurs in either a constant time interval or a constant traveled distance [55], at the end of which a new direction and speed are generated. The difference between the Random Walk Mobility model and the Random Waypoint Mobility model is that a host in the former chooses a random direction whereas a host in the latter generates a new destination for each movement.

The Random Direction Mobility model is also a widely used mobility model [56]. In this model, a mobile node chooses a random direction in which to travel. Different from the Random Walk Mobility model, a mobile node travels to the border of the simulation area in a selected direction and constant speed. As soon as the boundary is reached, the mobile node stops for a period of time before choosing another angular direction (between 0 and 180 degrees) and it continues this process.

2.2.2 Group Mobility Models

There are also a number of group mobility models designed for MANETs, although they may not be as widely used as entity mobility models. Reference Point Group Mobility (RPGM) model [34] is a generic group mobility model. In this model, the movement path of a group is predefined by a series of points which are referred to as “reference points”. Each group has a group leader which serves as the logical center of the group. Every mobile node follows the movement of the logical center with a random deviation in its position to that of the logical center. It is compulsory

to predefine group membership and group leaders before running a simulation, which are not allowed to change during the simulation.

Reference Velocity Group Mobility model [35] is a variant of RPGM model. The feature of this model is to use velocity as a reference factor to form groups. In this model, each group is configured with a group velocity and every group member is given another velocity which is slightly deviated from the group velocity. In the beginning, all mobile nodes are deployed in an area in which each group may be overlapping to other groups. A movement direction is generated for each group, namely the group direction. Next, mobile nodes start moving based on their group direction. After a period of time, each group will be separated from other groups. In this model, group membership, group velocity and movement direction must be predefined. This is not realistic in practice as the group members in this model are not allowed to change their velocity arbitrarily or move away from their groups.

There are some other group mobility models which are designed for special purposes. Column Mobility model, Nomadic Community Mobility model, and Pursue Mobility model are simulated by a Java application in [32]. These models are designed for particular purposes, such as tracking a target, or searching criminals, etc.. Zhou *et al.* proposed the Virtual Tracks (VT) model [48] which uses some "virtual tracks" to model the dynamics of group mobility based on pre-deployed switch stations such that group movement is restricted by tracks.

We note that group partitions and mergers could take place as a result of group mobility in MANETs. However, the existing models assume the membership of each group does not change which in turns does not allow mobile nodes to partition from groups or merge into other groups. We will address this issue in this chapter.

2.3 An Overview of the Reference Region Group Mobility Model

In this section, we will give an overview of our proposed mobility model that can be used to simulate group partitions and group mergers, namely the Reference Region Group Mobility model (RRGM). A generic case of group mobility in MANETs will be discussed where group partitions are triggered by events that generate new destinations. This can model the scenarios when a new task is assigned to a group, such as in a military operation, or in search and rescue operations in disaster recovery. In the following subsections, we will introduce the group mobility in RRGM, which is separated into two cases: a group assigned with a single destination and a group assigned with multiple destinations. In addition, group operations in RRGM, i.e. group partitions and group mergers, will also be described.

2.3.1 Mobility of a Group Assigned with a Single Destination

RRGM uses a novel concept, namely reference region, which is a dynamic area associated with a group of nodes. Location of a reference region changes dynamically during the simulation. The size of a reference region is determined by the number of nodes which are associated with it. Next, we will introduce the idea of reference region and how it is used by a group to move towards its destination in RRGM.

We bring out the idea of reference region and introduce the mechanism of RRGM model with a basic case first where a mobile group is assigned with a single destination. Initially, a group of mobile nodes are deployed in the simulation area and an area is created as the so-called “reference region” (the grey circle in the

figures) for this group as shown in Figure 2.1 (a). Next, a destination is generated as shown in Figure 2.1 (b) and it is assigned to the group which was created previously. Subsequently, a new reference region is generated in an area between the group and the destination (the location of the reference region and how to determine its size will be introduced in the next section). Assuming that every mobile node has the knowledge of the location of the reference region of its group, each of them randomly selects a point within the new reference region as its target and will move towards it (which is denoted by a dot in Figure 2.1 (c)). Later on, after all mobile nodes arrive at their respective targets in the reference region, the reference region will be relocated to a new area which is closer to the destination as illustrated by Figure 2.1 (d) and Figure 2.1 (e). In this way, the reference region is iteratively relocated such that it is closer to the destination after each iteration. This process will stop when the destination falls within the most recent reference region being generated as shown in Figure 2.1 (f). Finally, all mobile nodes arrived at the destination, after which the destination is removed to indicate the arrival of the group as shown in Figure 2.1 (g) and Figure 2.1(h).

When mobile nodes arrive at the destination, they may either pause for some time or continue their movement within the reference region. A node can continue its movement by randomly selecting a new point within the reference region.

Figure 2.1 (a) : A group of mobile nodes

Figure 2.1 (b): A destination is generated and assigned to the group

Figure 2.1 (c): The reference region is placed to a new location

Figure 2.1 (d): Movement of mobile nodes to the reference region

Figure 2.1 (e): New location of the reference region

Figure 2.1 (f): The destination is one of the points in the reference region

Figure 2.1 (g): Mobile nodes moved to the destination

Figure 2.1 (h): The destination is removed

Figure 2.1: Mobility of a group assigned with a single destination

2.3.2 Mobility of a Group Assigned with Multiple Destinations

Figure 2.2 (a): A group of mobile nodes

Figure 2.2 (b): Two destinations are generated and assigned to the group

Figure 2.2 (c): The original reference region is split into two new reference regions

Figure 2.2 (d): The group partition is completed

Figure 2.2: Mobility of a group assigned with multiple destinations

In group mobility of MANETs, it is possible for a group to be assigned with multiple destinations simultaneously. When that happens, a group partition will take place as illustrated by Figure 2.2 (a) to Figure 2.2 (d). Initially in Figure 2.2 (a), a group of mobile nodes are created and deployed into the simulation area. Its reference region is also generated. Next, two destinations are generated simultaneously and assigned to the group as displayed in Figure 2.2 (b). Correspondingly, two reference regions are created and each of them is placed between the group and one of the destinations respectively. Every mobile node in the original group randomly selects a point within either reference region as its target as shown in Figure 2.2 (c). Finally, each mobile node moves to their respective targets in the corresponding reference regions. Similar to the previous case as described in subsection 2.3.1, the reference regions are iteratively relocated such that it is closer to the destination after each iteration and eventually mobile nodes will arrive at their respective destinations.

2.3.3 Merger of Groups

Reference region is also used for mergers between groups in RRGGM. Practically, it is likely for a small group to merge with a larger group in MANETs. As displayed in Figure 2.3 (a), there are two groups in the simulation area of which one group is bigger than the other. In RRGGM, for a group not assigned with any destination, it can merge with another group. To do so, group members from the smaller group change their membership to the bigger group (the details will be discussed in section 2.6). As a result of more mobile nodes joining into the bigger group, the resultant reference region, the size of which is proportional to the number of nodes in the group, is also enlarged (the details will be discussed in section 2.4) as

illustrated in Figure 2.3 (b). Every mobile node from the smaller group randomly selects a point within the new reference region and moves towards it. When all mobile nodes from the smaller group have reached the larger group's reference region, the merger between these two groups is completed as shown in Figure 2.3 (c).

Figure 2.3: Merger of two groups

It is easy to see that RRGGM can be used to model the occurrence of group partitions and mergers in group mobility of MANETs. An example of such an event taking place is in the search and rescue operations. For such an operation, a rescue team may be assigned with several tasks simultaneously. As a result, some team members have to move apart from the original team that leads to a group partition. After a team of rescuers reached its destination and carried out tasks, they may merge with other teams.

2.4 Model Configurations and Implementations

Table 2.1 summarizes the input parameters used in RRGGM and their default values. Their specific usages will be discussed along with the introduction of model implementations.

Table 2.1: Input parameters of RRGGM

Spatial parameters		Default values
$T\text{-length}$, $T\text{-width}$	Dimensions of the simulation area (terrain size) in length and width (in meters).	1000m, 1000m
(x_0, y_0)	Center's coordinates of the initial group.	(500, 500)
Group and node related parameters		
$N_{total-node}$	Total number of mobile nodes in the simulation.	50
$N_{Initial-dest}$	Number of destinations initially generated.	3
$N_{standby-g}$	Standby groups initially deployed.	0
ρ	Density of nodes in a group. (nodes/100m ²)	0.1
v_0	Average node velocity. (m/s)	10
γ	Velocity coefficient.	0.5
Timing parameters		
T	Simulation time (in second).	1000s
T_d	Interval for generating new destinations (in second).	15s
τ_0	Pause time for a mobile node at a location (in	0s

	second).	
τ_1	Pause time for a reference region at a location (in second).	2s
τ_2	Idle time for a group at a destination (in second).	10s
Miscellaneous		
η	Ratio between the length and the width of a reference region (for rectangular reference region). If $\eta=0$, it represents circular reference region is used.	1
d_x, d_y	Distance granularity for defining movement trajectories of groups' in the range of (0, 1).	0.3, 0.3
δ	A distance threshold between the center of a group and a destination (in meters).	10
$\{(M_{xl}, M_{yl}), \dots, (M_{xk}, M_{yk})\}$	Intermediate checkpoints of reference regions (optional, used in scenarios with predefined destinations).	n.a.

Note as stated in Table 2.1, the reference region used could take the shape of either a rectangle or a circle, which is controlled by the parameter η . During implementation of the model, when $\eta=0$, it indicates that a circle is used as the shape of the reference region. Next, we give the definitions of some terms used in RRGGM.

- a. **Center of a group/location of a group:** it refers to the point whose coordinates are obtained by taking the average of the coordinates of all mobile nodes in the group.
- b. **Center of a reference region/location of a reference region:** it refers to the midpoint of a diagonal in the rectangle reference region, or the center of the circular reference region.
- c. **Distance between two groups:** it refers to the distance between the centers of two groups.

Initially, all mobile nodes (i.e. $N_{total-node}$ nodes) are deployed in the simulation area as a single group. A reference region will be created at the center of the initial group, whose coordinates are (x_0, y_0) .

If a rectangle is used as the shape of the reference region, the length l and width w of the reference region are jointly determined by both $N_{total-node}$ and density of nodes in a group, namely ρ . Their relationship can be expressed by the equation:

$$N_{total-node} = \rho \times (l \times w) \quad (2.1).$$

As the ratio between the length l , and the width w , of a reference region, namely η , is specified, where $\eta = l/w$, l and w can be calculated by:

$$l = \sqrt{\frac{\eta \times N_{total-node}}{\rho}} \quad (2.2), \text{ and}$$

$$w = \sqrt{\frac{N_{total-node}}{\rho \times \eta}} \quad (2.3)$$

Note that the center of the reference region, whose coordinate is (x_0, y_0) , is the midpoint of a diagonal in the rectangle reference region as illustrated in Figure 2.4. So the range of the reference region would be from the bottom left $(x_0-l/2, y_0-w/2)$ to the top right $(x_0+l/2, y_0+w/2)$.

Figure 2.4: Illustration of a reference region

Similarly, if circle is considered as the shape of a reference region, the center of the circular reference region is (x_0, y_0) and its radius r can be calculated as:

$$r = \sqrt{\frac{N_{total-node}}{\pi \times \rho}} \quad (2.4)$$

Subsequently, $N_{Initial-dest}$ destinations are randomly generated. The initial group is logically partitioned into $N_{Initial-dest} + N_{standby-g}$ subgroups which are composed of $N_{Initial-dest}$ active subgroups (equivalent to the number of destinations) and $N_{standby-g}$ standby subgroups as specified in the configurations (Table 2.1). Standby groups are used to model situations where groups do not have destinations to move towards.

The default value of $N_{standby-g}$ in our implementation is zero. The destinations which have been generated are assigned to the active subgroups. As a result each active subgroup has one destination. For mobile nodes, each of them randomly selects a subgroup to join in.

Next, a reference region will be generated for each active subgroup. For a standby subgroup, the location of its reference region is placed at the center of the subgroup. The dimensions (the *length* and the *width* of a rectangle reference region, or the radius of a circular reference region) can be calculated using equations (2.2) and (2.3), or (2.4) respectively by replacing $N_{total-node}$ with the number of nodes in the subgroup. For an active subgroup, the location of its reference region can be calculated by the following steps:

Step 1: Given an active group i , let the coordinates of its destination be (x_d, y_d) , and the number of nodes in the subgroup be N_i .

Step 2: As it is already known that the coordinates of the initial group's location is (x_0, y_0) , we can calculate the location of the reference region of group i , namely (x_i, y_i) , by the equations:

$$x_i = (x_d - x_0) d_x * rand() + x_0 \quad (2.5)$$

$$y_i = (y_d - y_0) d_y * rand() + y_0 \quad (2.6)$$

where d_x and d_y , in the range of $(0,1]$, are the distance granularity for defining the trajectory of a reference region to the destination. $rand()$ is a random seed to generate a random number in the range of $(0, 1)$ in order to avoid generating exactly same trajectory of a reference region to a destination by these equations.

Step 3: The dimensions (the *length* and the *width* of a rectangle reference region, or the radius of a circular reference region) of the reference region can be calculated using equations (2.2) and (2.3), or (2.4) respectively by replacing $N_{total-node}$ with N_i .

2.5 Node's Movement and Displacement of a Reference Region

As stated in subsection 2.3.1, once a reference region is generated for a group, the respective group member will randomly select a point within the range of the reference region as its target for movement. The velocity that each group member travels with, namely v , is varied by this equation:

$$v = \gamma * v_0 + rand() * v_0 \quad (2.7)$$

where v_0 is a pre-defined average velocity of mobile nodes, and the random seed $rand()$ returns a random value within the range of (0, 1). γ is a coefficient of v_0 and $\gamma * v_0$ contributes a fixed value to v . Once a mobile node arrives at the point it has previously selected, it may pause for a period of time τ_0 before continuing its movement by choosing a new point within the reference region as its target. The mobility of a mobile node within the range of its reference region is similar to that of the Random Waypoint Mobility in the sense that a mobile node travels with a selected speed towards a selected destination in each iteration. Practically, this can mimic the mobility of a team of rescuers carrying out operations in a small area before moving to a new disaster venue.

After all group members arrive at the reference region, the reference region may remain at its current location for a certain period, namely pause time of a reference region τ_l , which is timed from the arrival of all group members to the reference region till a new location of the reference region is generated. Assuming that the

current location of a reference region is (x_i, y_i) , its new location (x_{i+1}, y_{i+1}) can be calculated by:

$$x_{i+1} = (x_d - x_i) d_x * rand() + x_i \quad (2.8)$$

$$y_{i+1} = (y_d - y_i) d_y * rand() + y_i \quad (2.9)$$

The basic ideas of these two equations are as same as equations (2.5) and (2.6) respectively. By adjusting the values of d_x and d_y , a group's movement trajectories could be different and it is illustrated by Figure 2.5. Circles in Figure 2.5 represent snapshots of a reference region at each step towards the destination, which starts from the bottom right of the figure and ends at the top left of the figure. Figure 2.5 (a) shows a concave movement trajectory of a reference region where the reference region moves in larger steps in the horizontal direction at the beginning stage. In contrast, the movement trajectories of the reference region in Figure 2.5 (c) and Figure 2.5 (d) respectively are convex where the reference regions move in larger steps in the vertical direction. The movement trajectory of the reference region in Figure 2.5 (b) is neither apparently concave nor convex compared with others. The reason behind this phenomenon is that d_x and d_y are assigned with different values. d_x and d_y are used to control the granularity of every movement of the reference region. When d_x or d_y is given a larger value, the reference region will move in bigger steps in the corresponding horizontal or vertical direction respectively. For instance in Figure 2.5 (d) where $d_y=0.75$ and $d_x=0.25$ (i.e. $d_y > d_x$), the reference region moves closer to the destination faster in the vertical direction than that in the horizontal direction. Hence, the resultant trajectory is convex in nature. Notably, the reference region takes larger steps towards the destination at the initial stage of movement. When it gets closer to the destination, the reference region moves in

smaller steps. This can mimic the situation that a rescue team moves faster when it is far from the destination, and when it gets closer, it needs to slow down for establishing instruments or carrying out operations. Consequently, the distance between each adjacent location of the reference region becomes shorter as illustrated in Figure 2.5.

Figure 2.5: Snapshot of displacement of reference regions by adjusting d_x and d_y

When the distance from the location of a reference region to its destination is smaller than a threshold δ , it may be considered that the group has reached the destination, and the group may pause for a duration τ_2 , namely group idle time, at this location. This ensures sufficient time for a group to move around the destination area to complete their tasks. After τ_2 , if there is no new destination assigned to this group, it becomes a standby group to indicate the completion of the task.

2.6 The Merger of a Standby Group with Another Group

In RRGGM, a standby group can merge with another group. We assume that every mobile node has the knowledge of the location of all the groups, the location of the reference region of its group's, and the number of nodes in each group during a simulation. Based on this information, mobile nodes in a standby group can calculate the distance between its group and other groups respectively. The closest group, which has the smallest distance to the standby group, will be selected as the target for the standby group to merge with. To do so, each group member in the standby group changes its group membership to the target group, and then randomly selects a point within the reference region of the target group's and moves towards it as described in subsection 2.3.3. The dimensions of the new reference region, which is proportional to the number of nodes in the group, will also be enlarged due to more group members joining into this group according to the equations 2.2, 2.3 and 2.4. When all the mobile nodes from the standby group arrive to the reference region of the larger group, the process of this group merger is completed.

2.7 Group Partitions and Mergers with Predefined

Destinations

We have discussed the generic usage of RRGGM model when destinations are randomly generated in Section 2.5. However, in some practical applications of MANETs, locations of destinations could be fixed. Such kinds of applications can be found in a group of people visiting in an exhibition hall, firefighters operating in a building, and policemen searching criminals in a floor. The destinations in the

above mentioned applications are referred to as “predefined destinations” in RRGGM. In this section, we will discuss how RRGGM can be used to model these specific scenarios where group’s destinations need to be predefined before the simulation.

To model scenarios with predefined destinations, an intermediate checkpoint (M_{xI}, M_{yI}) is needed. It is used to represent, for example, the end of a corridor in a building. (M_{xI}, M_{yI}) is assigned to the initial group as the first destination. Meanwhile, all rooms or booths along the two sides of the corridor are considered as “hidden” destinations as their initial state. Similar to what we have discussed in Section 2.5, a reference region is generated between (M_{xI}, M_{yI}) and the location of the initial group. The group will gradually move towards (M_{xI}, M_{yI}) using the same procedure as described in Section 2.3.1 (Mobility of a group assigned with a single destination).

When the reference region passes a hidden destination, that destination becomes visible to the group and is assigned as a new destination for the group. This can be illustrated by Figure 2.6. As shown in Figure 2.6, AC is the distance between the group’s current location and its initial location; AB is the distance between the group’s initial location and the “hidden” destination d; AD is the projection of AB onto AC. The criterion for determining if a group has passed a “hidden” destination or not is to compare the length of AD and AC. If $AD < AC$, the initial group has passed the “hidden” destination and the destination is assigned to the initial group. As the initial group has multiple destinations (i.e. the destination (M_{xI}, M_{yI}) and the newly assigned “hidden” destinations), new reference regions will be created on each newly assigned “hidden” destination. The center of each new reference region

uses the coordinates of respective “hidden” destination’s. Mobile nodes in the initial group will either randomly select a newly created reference region to move towards or continue moving with the initial group.

Figure 2.6: Illustration of a group passing a "hidden" destination

Note that a “hidden” destination that has been visited will be marked and will not be visited again. After all the members of a subgroup reached a “hidden” destination and roamed around the area for a duration which is equal to the group idle time τ_2 , they may join back with the initial group by merging their reference region into the reference region of the initial group’s as discussed in subsection 2.6. After the merger, the dimensions of the initial group’ reference region (the *length* and the *width*, or the *radius*) will be increased since the size of a reference region is proportional to the number of nodes in the group as derived according to equations (2.2) and (2.3) for a rectangle reference region, or (2.4) for a circular reference region. In addition, mobile nodes which belonged to the previous subgroup will randomly select new points within the initial group’s reference region as their

targets to move towards. This process is as same as what is described in the subsection 2.3.3 (Merger of groups).

2.8 Simulation and Discussion

As discussed in the above sections, RRGGM model can be used to model generic group mobility and specific scenarios where destinations are predefined. It also can model entity mobility, e.g. Random Waypoint Mobility. Assuming that a reference region is stationary during the simulation (this can be realized by giving a big value to group partition interval), a node associated with this reference region will iteratively select a new target within the reference region for movement and this process will repeat until the simulation ends. The mechanism of this mobility pattern can be considered the same as that of Random Waypoint Mobility, which is discussed in Section 2.2. In this section, simulations are conducted for each of these scenarios. The simulation is developed under *C++* platform and visualized by *NS2-nam* [57].

2.8.1 Generic Group Mobility Pattern in MANETs

In this subsection, we will present the generic group mobility operations of group partitions and mergers simulated by RRGGM model, where destinations are generated periodically during the simulation. The parameters for the simulation setting are configured as in Table 2.2. The size of simulation area is 1000 meters by 1000 meters for the length and the width respectively (i.e. $T\text{-length}=1000$ and $T\text{-width}=1000$). Each simulation is run for 1000 seconds (i.e. $T=1000$). 50 mobile nodes ($N_{total-node}=50$) are deployed at (500, 0) initially (i.e. $(x_0, y_0)=(500, 0)$). 3 destinations (i.e. $N_{Initial-dest}=3$) are generated initially and no standby group will be

generated (i.e. $N_{standby-g} = 0$). Node density of each group is $0.1 \text{ node}/100m^2$ (i.e. $\rho = 0.1 \text{ node}/100m^2$). The average speed of mobile nodes is $10m/s$ (i.e. $v_0 = 10$) which is distributed in the range of $(5m/s, 15m/s)$ derived according to the equation (2.7) where the coefficient γ is equal to 0.5 . A new destination will be generated at every 15 seconds (i.e. $T_d = 15$). A node will not pause during the simulation (i.e. $\tau_0 = 0$), and a reference region pauses for 2 seconds (i.e. $\tau_l = 2$ which is timed from the arrival of all group members to the reference region till a new location is generated for this reference region). The group idle time τ_2 is configured to 10 seconds. We use circle as the shape of reference regions in the simulation (i.e. $\eta = 0$). The distance granularity for group's movement is 0.3 for both d_x and d_y . When the distance between the center of a group and its destination is less than 10 meters (i.e. $\delta = 10$), this group is considered to have already arrived to the destination.

Traces of nodes' movement are recorded and *NS2-Nam* is used for visualization. Screenshots for mobile nodes in the network are captured to demonstrate their mobility. Figure 2.7 presents the occurrence of a group that is partitioned into several subgroups which subsequently merge. It is used to reflect group mobility related applications in MANETs, such as military operations in battlefields, disaster recovery and scientific explorations. Destinations can be used to represent enemy's bases in military operations, or destroyed sites in disaster recovery scenarios.

Table 2.2: Input parameters for the simulation of generic group mobility pattern

$T\text{-length}=1000$	$T=1000$
$T\text{-width}=1000$	$T_d=15$
$(x_0, y_0)=(500, 0)$	$\tau_0 = 0$
$N_{\text{total-node}} = 50$	$\tau_l = 2$
$N_{\text{Initial-dest}} = 3$	$\tau_2 = 10$
$N_{\text{standby-g}} = 0$	$\eta = 0$
$\rho = 0.1$	$d_x = 0.3$
$v_0 = 10$	$d_y = 0.3$
$\gamma = 0.5$	$\delta = 10$

As shown in Figure 2.7 (a), initially all nodes are deployed in a small area as the initial group. When three destinations D1, D2 and D3 are generated, the initial group is partitioned into three subgroups, A, B and C. Meanwhile, reference regions are also generated for each subgroup denoted by circles in the figures. Mobile nodes move towards their corresponding reference regions gradually. As shown in Figure 2.7 (b) at the time $t=15s$, a new destination D4 is generated. The closest subgroup B, which is moving towards D2, is split into two subgroups, B and D, and the subgroup D heads towards the new destination D4. At $t=20s$, the subgroup C arrived at D1 and it becomes a standby group, and D1 is removed as illustrated in Figure 2.7 (c). From Figure 2.7 (d) to Figure 2.7 (f), screenshots for a group merger process are captured. In Figure 2.7 (d), the two smaller subgroups E and F are standby subgroups while the other subgroup G is an active subgroup. In Figure 2.7 (e), the reference region of the subgroup E is merged with subgroup F's reference region. The process of this group merger is completed at $t=85s$ when all group

members in F move into the reference region of subgroup F as shown in Figure 2.7

(f).

2.8.2 Group Mobility with Predefined Destinations

In this subsection, we will discuss group mobility operations of group partitions and mergers where groups' destinations are predefined. The input parameters to generate this scenario are listed in Table 2.3. Different from the parameter settings for the generic case of group mobility in Table 2.2, the simulation time is 150 seconds (i.e. $T=150$) and the interval to generate new destinations is 200 seconds (i.e. $T_d=200$) which is higher than the simulation time such that no new destination will be generated during the simulation in this scenario; the initial number of destination is zero (i.e. $N_{Initial-dest}=0$); node density is 0.3 node/100m² (i.e. $\rho=0.3$) which is higher than the generic case; The average speed of mobile nodes is at 5m/s (i.e. $v_0=5$) which is distributed in the range of (2.5m/s, 7.5m/s) derived according to the equation (2.7) where the coefficient γ is equal to 0.5; a reference region will not pause at any location as given $\tau_1=0$ and the group idle time τ_2 is configured to 25 seconds; $d_x=0$ indicates that the initial group only moves along the y-axis; d_y is set to 0.01 so that the initial group will advance slowly in every small step. (M_{xI}, M_{yI}) is the destination of the initial group as introduced in Section 2.7. Note, during the implementation of this scenario, subgroups can only merge with the initial group instead of finding the closest one as in the generic case (this can be done by giving the initial group an ID number such as "1" and all subgroups are only allowed to merge with the group with ID number "1").

Table 2.3: Input parameters for the simulation of group mobility with predefined destinations

$T\text{-length}=1000$	$T=150$
$T\text{-width}=1000$	$T_d=200$
$(x_0, y_0)=(500, 0)$	$\tau_0=0$
$N_{\text{total-node}}=50$	$\tau_l=0$
$N_{\text{Initial-dest}}=0$	$\tau_2=25$
$N_{\text{standby-g}}=0$	$\eta=0$
$\rho=0.3$	$d_x=0$
$v_0=5$	$d_y=0.01$
$\gamma=0.5$	$\delta=10$
	$(M_x, M_y)=(500, 1000)$

These kinds of applications can be found in various scenarios which include the cases of firefighters operating in a building, tourists visiting an exhibition hall, and police officers searching criminals in a floor. For instance, consider the case of a group of tourists who are visiting an exhibition hall as illustrated in Figure 2.8. Ten destinations, which represent art galleries or booths along the two sides of the corridor, are predefined and denoted by D1 to D10. Initially, all the tourists walk together along the corridor. When the tourist group passes by exhibition booths, e.g. D1 and D2, some people may split from the group and visit the booths. After staying at the booths for a while, they may join back to the initial group, e.g. subgroups B and C which just visited the booths D1 and D2 respectively as displayed in the figure. It is also notable when the initial group A passed D3 and D4, two new subgroups D and E are partitioned from the initial group A and move towards each respective booth as illustrated in the figure. The initial group A will

continue its movement until it arrives at the end of the corridor and all these “predefined destinations”, e.g. booths from D1 to D10, would be eventually visited.

Figure 2.8: Group mobility inside a building (e.g. an exhibition hall)

2.8.3 Entity Mobility

Entity mobility model is used to mimic the movement of an individual node. As it is well-known, Random Waypoint Mobility model is widely used in many research works. In RRGGM model, a mobile node moves by selecting a new target within the range of the reference region for movement in each iteration. The mechanism of this mobility pattern is similar to that of Random Waypoint Mobility, which is discussed in Section 2.2. By taking this advantage, RRGGM can also be used for modeling entities' mobility. By setting the interval to generate destinations to a maximum value (e.g. 10000), no new destination will be generated during the simulation. Mobile nodes will move within the boundaries of the initial reference region till the end of the simulation. According to equations 2.2, 2.3 and 2.4, the dimensions of a reference region are inversely proportional to the node density in

the group. Therefore, assuming that the number of nodes in a group is fixed, reducing the node density of the group will cause the reference region to become larger. In particular, we note that in equation 2.1 we have $N_{total-node} = \rho * (l * w)$; if we replace l and w with $T-length$ and $T-width$ respectively, the reference region of the initial group would be expanded to the entire simulation area with $T-length$ and $T-width$ to be the length l and the width w of the initial group's reference region respectively. In doing so, the node density ρ would be calculated by: $\rho = N_{total-node} / (T-length * T-width)$. Consequently, entity mobility of mobile nodes can be modeled by RRGm.

The configurations of the input parameters for this application are given in Table 2.4. The size of simulation area is 600 meters by 600 meters for the length and the width respectively (i.e. $T-length=600$ and $T-width=600$). 30 mobile nodes ($N_{total-node}=30$) are deployed at (300, 0) initially (i.e. $(x_0, y_0)=(300, 0)$). Each simulation is run for 150 seconds (i.e. $T=150$). The interval to generate new destinations is 10000 seconds which is much larger than the simulation time such that no new destination will be generated during the simulation. The average speed of mobile nodes is $2m/s$ (i.e. $v_0=2$) in the range of $(1.5m/s, 2.5m/s)$ derived according to the equation (2.7) where the coefficient γ is equal to 0.5. A mobile node will not pause during the simulation (i.e. $\tau_0=0$). Rectangle reference regions are used in the simulation (i.e. $\eta=1$). It should be noted that other input parameters are in a negative value (-1) which indicates that they are not needed in this scenario.

Table 2.4: Input parameters for modeling Random Waypoint Mobility

$T\text{-length}=600$	$T=150$
$T\text{-width}=600$	$T_d=10000$
$(x_0, y_0)=(300, 0)$	$\tau_0=0$
$N_{\text{total-node}}=30$	$\tau_1=-1$
$N_{\text{Initial-dest}}=-1$	$\tau_2=-1$
$N_{\text{standby-g}}=-1$	$\eta=1$
$\rho=N_{\text{total-node}}/(T\text{-length}*T\text{-width})$	$d_x=-1$
$v_0=2$	$d_y=-1$
$\gamma=0.5$	$\delta=-1$

Figure 2.9 illustrates the distribution of mobile nodes within the simulation terrain, which is implemented by RRGm model to imitate Random Waypoint Mobility. The mobile node No.5 is tracked during a period of 150 seconds and its movement trajectory is recorded as shown in Figure 2.10. From the trajectory of this mobile node, we find it appears to have the same mobility pattern as Random Waypoint Mobility by comparing with other simulation results of Random Waypoint Mobility model [7, 32].

Figure 2.9: An example of Random Waypoint Mobility model implemented by the RRGM model

Figure 2.10: The movement trajectory of a mobile node in Random Waypoint Mobility generated by RRGM

2.9 Conclusion

It is essential to support group communications in MANETs, which are very useful in scenarios of disaster recovery, battlefield operations, and scientific exploration. Entity mobility models are not ideal representatives for group mobility in MANETs. Furthermore, existing group models are not able to model group partitions and group mergers which are common phenomenon in MANETs. Therefore, we proposed RRGM model for simulation studies in MANETs which can be used to simulate both group mobility and entity mobility. By taking the advantages of RRGM, group operations, such as partitions and mergers, can be realized. Simulations are conducted in modeling a number of different applications of ad hoc networks. With this mobility model, the effectiveness and the efficiency of routing protocols for MANETs could be evaluated under more realistic group mobility environment (this will be discussed in Chapter 3). In addition, based on this work, clustering algorithms for dynamic network topologies and group partition prediction schemes could be studied and developed (this will be discussed in Chapter 4 and Chapter 5 respectively).

CHAPTER 3: PERFORMANCE EVALUATION OF NETWORK ROUTING PROTOCOLS USING RRGM

3.1 Introduction

Network routing protocols are an important research topic in MANETs and has gained a lot of interest among the research community. A network routing protocol is used to exchange data packets between network users. Different from those in traditional computer networks, routing protocols in MANETs do not rely on fixed equipments, e.g. routers and base stations, for routing packets. In MANETs, nodes start out without any knowledge of the network topology; instead they have to discover who their neighbors are personally. The basic method is that a new node may announce its presence and would listen for announcements broadcasted by its neighbors. Each node learns about nodes nearby and how to reach them, and may also announce that it can reach them. Such discovery mechanisms allow routing information to be exchanged among all mobile nodes.

Mobility patterns have significant impact on the performance of network routing protocols and have been discussed in many previous research works [8, 10, 11, 18, 24, 58, 59]. Various mobility models can be applied to different scenarios of MANETs. In general, there are two types of mobility models, namely entity mobility models and group mobility models [7]. Entity mobility models represent mobile nodes whose movements are independent of each other, while group mobility models are used to simulate movements of groups'. Both of them are essential to the simulation studies of MANETs. However, most existing works for studying the performance of routing protocols uses the entity mobility models which describe individual's mobility behaviors. Examples of such models include the Random Waypoint Mobility model [25, 32] and the Random Walk Mobility model [33, 53]. On the other hand, we note that limited studies have been done on the impact of group mobility on network performance and they may usually assume that the network is connected, i.e. there are no group partitions and mergers. With the aid of RRGM model, we will evaluate the performance of some existing network routing protocols under group mobility patterns in this chapter.

3.2 Review of Network Routing Protocols

In ad hoc networks, routing protocols are typically categorized into two classes, table-driven routing protocols and on-demand routing protocols [60, 61]. Next, we will elaborate on these routing protocols in the following subsections.

3.2.1 Table Driven Routing Protocols

Table-driven routing protocols are also called as proactive routing protocols as these protocols have to maintain fresh lists of destinations and their routes by

Chapter 3: Performance Evaluation Of Network Routing Protocols Using RRGm

periodically distributing routing information throughout the network, even if they do not have any packet for delivery. A table is used to store routing information which is maintained by periodically propagating route discovery messages to establish or update routing information. In table-driven routing protocols, mobile nodes maintain routing information for all of their network destinations and the traffic to such destinations. It requires each node to organize their routing information and whenever there is a change in network topology, a complete updating has to be made throughout the entire network.

Destination-Sequenced Distance-Vector Routing protocol (DSDV) [62] is a well-known table-driven routing protocol in MANETs. In DSDV, every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded as illustrated by Figure 3.1 for the routing table of node B.

Figure 3.1: The routing table of the node B in DSDV

Chapter 3: Performance Evaluation Of Network Routing Protocols Using RRGM

Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. Each entry in the routing table is marked with a sequence number (i.e. Destination Seq #) assigned by the destination node, which is incremented by 1 in every update iteration. The format of the update message is $\langle dest_addr, dest_seq\#, hop_count \rangle$ (i.e. destination address, destination sequence number, and hop counts from the destination, respectively). For example, node D, which wants to join in the network presented in Figure 3.1, will send an update message, i.e. $\langle D, 101, 0 \rangle$. “101” is the destination sequence generated by node D and “0” is the hop count from D. When node B receives the update message of node D via node C, node B will insert a new entry in its routing table as illustrated in Figure 3.2.

Figure 3.2: The routing table of node B after node D joins in the network in DSDV

DSDV is suitable for an ad hoc network with small number of nodes. The sequence numbers in DSDV enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Whenever the topology of

the network changes, an update through the entire network is necessary; thus, DSDV is not suitable for highly dynamic networks.

The Wireless Routing Protocol (WRP) [63] is also one of the popular table-driven routing protocols in MANETs. In WRP, each node in the network is responsible for maintaining four tables: *distance table*, *routing table*, *link-cost table*, and *message retransmission list table (MRL)*. The *distance table* of a node x contains the distance from x to each destination node y via each neighbor z of x . The *routing table* of node x contains the distance information of each destination node y from node x , the predecessor and the successor of node x on this path. The *link-cost table* contains cost of link to each neighbor of the node and the number of timeouts since an error-free message was received from that neighbor. *MRL* contains information to let a node know which of its neighbors has not acknowledged its update message and the node has to retransmit the update message to that neighbor. Nodes exchange routing tables with their neighbors using update messages periodically on link changes. On receiving an update message, the node modifies its distance table and looks for better paths using the new information. If there is any new path found, the information is relayed back to the original nodes so that they can update their respective routing tables. The node also updates its routing table if the new path is shorter than the existing path. WRP has the same advantages as that of DSDV. But the complexity of maintenance of multiple tables in WRP demands a larger memory and greater processing power from nodes in the ad hoc wireless network. At high mobility, the control overhead involved in updating table entries would be extremely high which is the same as that of DSDV and hence WRP is not suitable for a highly dynamic ad hoc wireless network.

Besides DSDV and WRP, there are various other table driven routing protocols, such as Optimized Link State Routing (OLSR) [64], Hierarchical State Routing protocol (HSR) [65], Intrazone Routing Protocol (IARP) [66], Direction Forward Routing (DFR) [67]. This type of protocols maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. They are not suitable for large mobile ad hoc networks and dynamic network environment, as they need to maintain routing entries in the routing table of every mobile node. This causes more overhead for maintaining routing tables and slow reaction on restructuring from connection failures [8, 9].

3.2.2 On-demand Routing Protocols

On-demand routing protocols are also called as reactive routing protocols, because such protocols find routes on demand by broadcasting the route request packets. Every reactive protocol is characterized by a path discovery mechanism that is initiated when a source needs to communicate with a destination that it does not have a path to it. In on-demand routing protocols, the routes are created as required. When a source wants to send packets to a destination, it invokes route discovery mechanisms to find the path to the destination. Ad Hoc On-demand Distance Vector Routing (AODV) [29] and Dynamic Source Routing protocol (DSR) [30] are two well-known on-demand routing protocols for ad hoc networks. Next, we will review these two routing protocols in the following subsections.

3.2.2.1 Ad Hoc On-demand Distance Vector Routing (AODV)

AODV minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a pure on-demand route acquisition system, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges [12, 29]. In AODV, when a source node desires to send a packet but does not have a valid path to the destination, it initiates a *route discovery process* to locate the destination by broadcasting a *route request* (RREQ) message to its neighbors, which then forward the request to their neighbors and so on, until either the destination or an intermediate node with a “fresh enough” route to the destination is located as illustrated in Figure 3.3.

Figure 3.3: Propagation of the RREQ in AODV

Each node that forwards the RREQ creates a reverse route for itself back to the source node. The routing table is updated with the address of the neighbor from which the first copy of the broadcast message is received; thereby the reverse routes are established. Other additional copies of the same RREQ arrived later are discarded. The destination or any intermediate node with a “fresh enough” route to

the destination responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ as illustrated in Figure 3.4.

Figure 3.4: Propagation of the RREP in AODV

The RREP is routed back along the reverse path hop-by-hop. The intermediate nodes update their route tables with the node from which the RREP is received as forward route entries. A route timer is associated with each route entry after which the entry will be deleted. If an intermediate node moves, its upstream neighbors notices it and sends a *link failure notification* message to all its upstream neighbors to inform them of deletion of that route. The *link failure notification* message is relayed to the source which will choose to re-initiate a new *route discovery process* or discard.

3.2.2.2 Dynamic Source Routing (DSR)

DSR is a source-routed on-demand routing protocol [30]. In DSR, a node maintains route cache containing the source routes that it is aware of and updates entries in the route cache when it learns about new routes. The protocol consists of two major phases: *route discovery* and *route maintenance*. The *route discovery phase* is

initiated by broadcasting a *route request* (RREQ) when the source node does not find a route to the destination in its route cache or if the route has expired, as illustrated in Figure 3.5.

Figure 3.5: Propagation of the RREQ in DSR

This RREQ contains the address of the destination, along with the source nodes' address and a unique identification number. To limit the number of RREQs propagated, a node processes the RREQ only if it has not already seen it before. Each node receiving the RREQ checks whether it knows of a route to the destination. If it does not, it adds its own address to the *route record* of the packet and then forwards the packet along its outgoing links. A *route reply* (RREP) is generated when either the destination or an intermediate node with current information about the destination receives the RREQ. Figure 3.6 illustrates the transmission of RREP from the destination with its associated route record back to the source node.

Figure 3.6: Propagation of the RREP in DSR

In the route maintenance phase, each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route. Hello message is used to maintain the local connectivity of a node. By periodically broadcasting a hello message, a node may determine whether the next hop is within communication range. If no hello message is received, the node returns a route error (RRER) message to the original sender of the packet which can send the packet using another existing route or perform a new route discovery and remove the expired route information from its routing table.

3.2.2.3 Comparison between AODV and DSR

As discussed in the above two subsections, both AODV and DSR protocols employ a route discovery procedure. However, they have several important distinctions between each other. The most notable of these is that DSR uses more overhead in route constructions and route maintenance since each packet in DSR keeps much more routing information (such as the address of every node along the route) than that of AODV, whereas in AODV packets only contain the destination and source

address. DSR is intended for networks in which the mobile nodes move at a moderate speed and the network is relatively small [30, 68]. Additionally, DSR allows nodes to keep multiple routes to a destination in their route cache [61, 69]. When a link on a route is broken, the source node can check its route cache for another route. However, DSR does not contain any mechanism to validate route entries when it faces with a choice of multiple routes. This leads to stale route entries, particularly in a high speed mobility environment. On the other hand, AODV allows nodes to keep only one route entry to each destination in the cache. The *route discovery* process will be reinitiated if the route in the route table of the source node is invalid. Furthermore, AODV makes use of timers to indicate expiry of cache entries and uses sequence numbers to store the freshest routes.

3.2.3 Comparison between Table Driven Routing Protocols and On-demand Routing Protocols

The two classes of routing protocols, namely table-driven and on-demand routing protocols, are differentiated by the mechanisms which they use to maintain and update routes in ad hoc networks [9, 10, 13]. The key difference and features of table-driven and on-demand routing protocols are listed in Table 3.1. In table-driven routing protocols, when a source has a packet to send, the routing information will be available immediately from its routing table which is updated periodically by advertisements, e.g. *hello* messages. However, in on-demand routing protocols, the source, which wants to send a packet, has to trigger a route discovery process if it can not find any fresh enough route from its routing table or the routes in its routing table are no longer available, and thus the routing information is updated by request. Both table-driven and on-demand routing

Chapter 3: Performance Evaluation Of Network Routing Protocols Using RRG

protocols use more control overhead than routing protocols for traditional static networks. In a static network, less route maintenance is needed as there is inherently no change to the network topology. Routing overhead and updates are kept to a minimal volume. However, in dynamic network environments such as MANETs, fast change of network topology will result in massive routing overhead generated to update routing tables of each mobile node, especially for the nodes using table-driven routing protocols. On the other hand, on-demand routing protocols only need to update their routing information when they have packets to deliver. Hence, on-demand routing protocols generally outperform table-driven routing protocols in dynamic network environment in the aspect of using less control overhead. As table driven routing protocols are not adaptive to fast network topology changes as discussed in the subsection 3.2.1, we consequently choose two well-known on-demand routing protocols, i.e. AODV and DSR, for the study of the impact of group partitions and mergers on the network performance in this chapter.

Table 3.1: Comparison of table-driven and on-demand routing protocols

Parameters	Table-driven routing protocols	On-demand routing protocols
Availability of routing information	Immediately from the routing table	After a route discovery process if there is no route available in the routing table
Route update	Periodically updated by advertisements	Updated as required
Routing overhead	Proportional to size of the network and network mobility	Proportional to the network traffic

3.3 Network Environment Configurations

The simulation studies are carried out using the mobile network simulator QualNet [70]. QualNet is a commercial network simulation tool that was first released in 2000 by Scalable Network Technologies (SNT). It can be used in the simulation of MANETs, WiMAX networks, satellite networks, and sensor networks, etc.

Table 3.2: Configurations of network environment

Parameters	Values
Terrain size	1200 meters $\times$ 600 meters
Radio propagation range	250 meters
Channel capacity	2Mbits/s
CBR traffic	512 bytes/second
Simulation time	900 second
Number of nodes	50

To set up the simulation network environment, we generate 50 mobile nodes placed within a 1200 meters $\times$ 600 meters grid. Radio propagation range of each node is 250 meters with channel capacity of 2Mbits/s. 20 sender-receiver pairs are designated randomly and each source can generate constant bit rate (CBR) traffic of 512 bytes data packet every second. Every simulation is run for 900 seconds. Each scenario is repeated for 20 times and the average values are finally presented in the simulation results.

The simulation studies are carried out by varying the speeds of mobile nodes and node density¹. Increase of speed can lead to fast change of network topology whereas in turn affects the network performance. Node density is used to describe the relative distance between group members. When the node density is low, it indicates that group members have a longer distance between one another, whereas high node density (e.g. a great many mobile nodes in a small area) means group members will be closer to each other. With a fixed number of mobile nodes in a group, reducing the node density will result in an increase in the average distance between group members. This in turn will result in an increase of the coverage area of the group. We note that the changes of group coverage will eventually have an impact on the network performance especially for a network where group partitions would take place frequently and such phenomenon will be studied in this chapter.

3.3.1 Experimental Settings for Investigation on Node Speeds

This experiment is conducted according to two different scenarios, namely:

- i) Scenario I - group mobility with group partition disabled.
- ii) Scenario II - group mobility with group partition enabled.

As described in Chapter 2, RRG model is used to generate group mobility patterns with group operations, such as group partitions and mergers. In Scenario I, group partitions are disabled by assigning a big value (which is greater than the simulation time) to the “partition interval” in RRG. Therefore, group mergers would also not take place in this scenario. It is notable that due to the random

¹ Node density refers to the node density in a group, i.e. the number of nodes in an area where mobile nodes can move for their local movement in the group.

mobility of groups, the transitory overlapping of groups is not considered as group mergers in this work. On the other hand, group partitions and mergers would take place in Scenario II by assigning a suitable value to the “partition interval” (e.g. every 40 seconds) in the RRGM model. Idle subgroups, which do not have any assigned destinations for movement, will merge with other subgroups (as described in Chapter 2).

3.3.1.1 Scenario I - Group Mobility with Group Partition Disabled

In Scenario I, 50 nodes are initially deployed into three different subgroups such that each of the subgroup will consist of 16 or 17 group members. Since group partitions are not allowed to take place in this scenario, the membership of each group is fixed throughout the simulation. We identify 20 source-destination pairs where 50% data packet transmission is made via inter-group communications (the source and destination pairs are placed in different subgroups respectively) and another 50% is via intra-group communications (the source and destination pairs are placed in the same group).

3.3.1.2 Scenario II - Group Mobility with Group Partition Enabled

In Scenario II, 50 nodes are initially deployed all together in one group. Group partitions will take place in every 40 seconds interval with the RRGM model. However, if each subgroup has less than 10 nodes, it will not be further partitioned and newly generated destinations should be discarded. This is due to the fact that when group size² is very small, there will be many small subgroups generated during the simulation and the communications will mostly be in the manner of

² Group size is defined as the number of nodes in a group.

inter-group communications. Hence, the experimental results will not lose the nature of intra-group communications.

The node density is fixed at 500 nodes/ km² as configured in RRGM. We vary average node speeds from 1 m/s to 20 m/s to investigate its impact on the network performance. When the average node speed is high, more impetuous and arbitrary movement of groups will occur. Hence group partitions are expected to take place more frequently.

3.3.2 Experimental Settings for Investigation on Group Densities

In this experiment, node density is varied from 200 *nodes/km²* to 400 *nodes/km²* in steps of 50 *nodes/km²*. Node speed is randomly generated within the range of (15m/s, 25m/s). Group partitions and mergers are enabled in this experiment. As discussed in the earlier part of this section, changing node density will result in the change of group's coverage area. However, we notice that most intra-group communications can be done via one hop. If group partitions and mergers do not take place, changes of node densities will not have a significant impact on the network performance. Hence, the scenario where group partitions and mergers are disabled is not considered for this experiment. The group partition interval is fixed at 40 seconds in RRGM. Other parameters are the same as those used in the previous experiment described in the subsection 3.3.1.

3.4 Performance Evaluation of Routing Protocols Based on RRGM

In this section, a variety of network performance metrics will be presented for a comprehensive evaluation and comparison between AODV and DSR. These

metrics include the *packet delivery ratio (PDR)*, the *average control packets per data packet sent (ACP)*, and the *end-to-end delay*. *PDR* reflects the percentage of data packets that can be successfully delivered, which is an important metric to evaluate the efficiency of a network. *ACP* refers to the amount of routing packets required to set up and maintain routes in order to deliver data packets and it is then normalized by every data packet sent to indicate the average overhead spent in order to deliver a data packet. *End-to-end delay* measures the average time spent in the period when a packet is successfully sent from the source to the destination. Next, we will specifically introduce each of these performance metrics.

3.4.1 Performance Metrics

3.4.1.1 Packet Delivery Ratio (PDR)

PDR is calculated as the ratio of the number of data packets delivered to the destinations to those generated by the sources. Mathematically, it can be expressed as:

$$PDR = \frac{\sum P_{received}}{\sum P_{sent}} \quad (3.1)$$

In the above equation, $\sum P_{received}$ stands for the total number of data packets received at receivers, while $\sum P_{sent}$ is the total number of data packets sent during a simulation.

3.4.1.2 Average Control Packets per Data Packet Sent (ACP)

The control packets include *route request(rreq)*, *route reply* initiated by intermediate nodes(*rrep(1)*) and by destinations(*rrep(2)*) separately, and *route error*

$messages(rerr)$. We calculate the sum of all the control packets incurred during simulation, which is then normalized by the total number of data packet sent as represented by equation (3.2).

$$ACP = \frac{\sum rreq + \sum rrep(1) + \sum rrep(2) + \sum rerr}{\sum P_{sent}} \quad (3.2)$$

3.4.1.3 End-to-end Delay (EED)

End-to-end delay refers to the time taken for a packet to be transmitted across a network from a source to its destination. This metric can reflect the quality of communications between users. This includes all possible delays caused by buffering during route discovery, queuing at the interface queue, retransmission delays at the MAC layer, propagation delay and transmission delay. Node's mobility may cause the breakage of established links which result in the loss of data packets and additional delay incurred as a result of packet retransmissions. The overall *end-to-end* delay can be defined as:

$$EED = \frac{1}{P_{received}} \sum_{i=1}^{P_{received}} (r_i - s_i) \quad (3.3)$$

where $p_{received}$ is the number of successfully received packets at destinations, i is the unique packet identifier, and r_i is the time at which a packet with the unique id i is received, while s_i is the time at which a packet with the unique id i is sent.

3.4.2 Simulation Results

3.4.2.1 Varying the Average Speed

3.4.2.1.1 Packet Delivery Ratio vs. Average Speed

Figure 3.7 and Figure 3.8 presents the results which show how the *packet delivery ratio* varies with mobility speeds. As illustrated in Figure 3.7, the trends of *packet delivery ratio* for both AODV and DSR increase as the average speed increases when group partition is disabled.

Figure 3.7: Packet delivery ratio vs. Average speed (m/s) --- group partition disabled

The growth is 2% to 3% for both of AODV and DSR. Since 50% source-destination pairs are placed in different groups respectively for inter-group communications, they may not be connected initially if the sources can not find routes to their destinations (which may be due to the long distance beyond the transmission range or the lack of intermediate nodes between each other). When the network topology

changes as nodes move faster, those previously disconnected source-destination pairs which are placed in different groups would possibly get connected. As a result, more data packets can be received by the destinations in inter-group communications. Hence, the *packet delivery ratio* is increased as the speed increases.

Figure 3.8: Packet delivery ratio vs. Average speed (m/s) --- group partition enabled

On the contrary, the performance of *packet delivery ratio* for both DSR and AODV falls dramatically under the group partitioning scenario when nodes move faster as shown in Figure 3.8. A larger number of group partitions will occur when the node speed is increased. For example, a pair of source-destination previously in a same group would be easily separated into different groups when more group partitions take place and consequently data packets from the source can not be delivered to the destination. As a result, *packet delivery ratio* will drop, as reflected by the decreasing trend for both AODV and DSR respectively. It is notable that AODV still can retain a relatively high *packet delivery ratio* of 70% when the

average speed is increased to $15m/s$ to $20m/s$. Comparatively, only 50% packets can be received in DSR at the speeds of $15m/s$ to $20m/s$. As discussed in subsection 3.2.2.1, AODV reacts faster than DSR when the network topology changes because AODV only keeps one route entry and when it becomes invalid, it will reestablish a new route. However, DSR has to maintain multiple routes entries before reinitiating a new route discovery process. Hence, AODV is more adaptive to frequent network topology changes than DSR and consequently it yields a higher *packet delivery ratio* than DSR. It is notable the packet delivery ratio for both AODV and DSR is up to 95% to 98% where DSR is slightly higher than AODV for about 1% when the speed is $1m/s$. As described in the earlier subsection 3.3.1.2, initially all mobile nodes are deployed in one big group for this scenario. When the speed is low, nodes barely move during the simulation (compared to the high speed cases). Hence, routes maintained by both AODV and DSR can be valid for longer period of time. Therefore, the *packet delivery ratio* is higher for low average speeds.

Comparing with Figure 3.7 and 3.8, we notice that when group partition is enabled the *packet delivery ratio* drops very fast, which is due to the frequent changes of network topology. On the other hand, when group partition is disabled, the *packet delivery ratio* yielded from the intra-group communications will not change dramatically. As a result, the *packet delivery ratio* only varies in a small range from 60% to 65% as illustrated in Figure 3.7. Comparing with Figure 3.7 and 3.8, we notice that when group partition is enabled the packet delivery ratio drops very fast, which is due to the frequent changes of network topology. On the other hand, when group partition is disabled, the packet delivery ratio yielded from the intra-group communications will not change dramatically. As a result, the packet delivery ratio only varies in a small range from 60% to 65% as illustrated in Figure

3.7. The difference in the results of packet delivery ratio in group partition enabled and disabled scenarios is largely caused by their different mobility patterns. During the simulations in group partition enabled scenario, the number of subgroups in group partition enabled scenario should normally be higher than that in the group partition disabled scenario. In that experiment as described in section 3.3.1, new destinations are generated in an interval of 40 seconds. Mobile groups formed by 50 mobile nodes will partition if no idle group is available. Therefore, a sender and destination pair could have many options in establishing their connections via nodes in different groups, and once one route becomes invalid it is still easier for them to find another alternative route. In particular, when group partition is enabled, it implies group mergers also take place during a simulation. So a sender and destination pair could move into the same group for some duration before they partition again, which also improves the packet delivery ratio. Thus, the results of packet delivery ratio in group partition enabled scenario as shown in Figure 3.8 are higher than those in the corresponding group partition disabled scenario as shown in Figure 3.7.

3.4.2.1.2 Average Control Packets per Packet Sent vs. Average Speed

Figure 3.9 and Figure 3.10 present the results which show how *average control packets by per packet sent* varies with mobility speeds.

Figure 3.9: Average control packets per packet sent vs. Average speed

--- group partition disabled

In Figure 3.9 where group partition is disabled, the *average control packets* raises slightly as the speed increases and an average of 0.55 and 0.77 control packet is generated per data packet sent for AODV and DSR respectively. It tallies with what has been discussed in subsection 3.2.2.2 that DSR uses more control overhead than AODV in route maintenance. The control packets are mainly consisted of two portions: control packets for inter-group communications and intra-group communications. When group partition is disabled, the control packets for intra-group communications will be relatively stable since the group membership is

not changed and hence routes can be valid for longer period of time during the simulation. On the other aspect, the control packets generated for inter-group communications will vary with the mobility speeds. When speed is increased, more control packets for route maintenance of inter-group communications will be generated, therefore, it results a slight increase in the number of control packets for AODV and DSR in this scenario.

Figure 3.10: Average control packets per packet sent vs. Average speed

--- group partition enabled

In the group partition enabled scenario as displayed in Figure 3.10, more control packets are generated in both DSR and AODV when speed is raised. As initially, all mobile nodes are deployed together in one group, the intra-group communications dominates at the beginning. It can be easily understood that intra-group communications are more stable than inter-group communications when group partitions do not occur. Therefore, the control packets are relatively low when the speed is as low as $1m/s$ (which we can say they barely move compared to the high

speed cases). Correspondingly, it also can be reflected by a high *packet delivery ratio* as illustrated in Figure 3.8 when mobility speed is at 1m/s. However, group partition takes place more frequently when the mobility speed increases as what we observed during the simulation. When group partitions take place, the previous stable intra-group communications would be disrupted, which results in massive *route error* messages and control packets for new *route discovery* process which would occur in both AODV and DSR. Thus, the control packets increase dramatically with the increase of mobility speeds as displayed in Figure 3.10.

The *average control packets* for the two scenarios as displayed in Figure 3.9 and Figure 3.10 also appear differently. The basic reason behind the trend of these simulation results is similar to what has been described for *packet delivery ratio* that group partitions lead to more topology changes which result more control packets generated for route maintenance.

3.4.2.1.3 End-to-end Delay vs. Average Speed

Figure 3.11 and Figure 3.12 illustrate the results of *end-to-end delay* of data packet delivery in group partition disabled and enabled scenarios respectively. In Figure 3.11, DSR generates much higher *end-to-end delay* than that of AODV, which is the same as what it does in Figure 3.12 (the *end-to-end delay* is small for both AODV and DSR when the speed is at 1m/s because all nodes are initially deployed in one group as discussed in the earlier subsections). The *end-to-end delay* in DSR is over 1.5 seconds whereas the *end-to-end delay* of AODV is only less than 0.3 second for both scenarios.

Figure 3.11: End-to-end delay vs. Average speed --- group partition disabled

Figure 3.12: End-to-end delay vs. Average speed --- group partition enabled

This great distinction in *end-to-end delay* between AODV and DSR is due to the difference of their working mechanisms. As introduced in subsection 3.2.2.2, caching is designed for keeping routing information in AODV and DSR. Caching is used in both *route discovery process* and *route recovery process* to increase the

possibility of finding a route without initiating flooding of messages. In AODV, only one cache entry is allowed to be kept for each source-destination pair, while all possible routes are cached in DSR. In a dynamic environment, network topology changes very fast resulting in caching information becoming obsolete more quickly. Therefore, when a data packet is sent via a broken link in AODV, the node detecting the disconnection would return a *route error* message immediately and requests the source to reinitiate a new *route discovery process*. However, in DSR, the node detecting the route breakage sends a *route error* to the source node and the source would not reinitiate a new *route discovery process* until all route entries to the destination in its cache are tried.

It is possible for a source in DSR to keep multiple route entries via nodes in different groups to its destination. If a group, with which its intermediate nodes affiliate, has moved away, many of its route entries would become invalid simultaneously. The source node has to repair the broken links by trying all possible routes in its cache which results in a much higher *end-to-end delay*. On the contrary for AODV, as only one route entry is maintained by the source, the source can reinitiate a new *route discovery process* as soon as the route is no longer available. Therefore, the delay generated during the route recovery in DSR does not occur in AODV and thus the *end-to-end delay* in AODV is much lower than that of DSR as illustrated in Figure 3.11 and Figure 3.12.

Comparing Figure 3.11 and Figure 3.12, we notice that DSR generates lower *end-to-end delay* when group partition is enabled, which reduces from 1.7 seconds to 1.5 seconds approximately when the speed is up to 15m/s. As the mobility speed increases, more group partitions would take place and with the aid of more

subgroups, the source in DSR can find more alternative routes(via different subgroups) to the destinations. Therefore, the *end-to-end delay* in DSR can be reduced slightly. However, route entries in the sources in AODV become obsolete quickly when group partitions take place more frequently, and hence it yields a slightly higher *end-to-end delay*, which is about 0.05 second, in the partition enabled scenario.

3.4.2.2 Varying the Node Density

3.4.2.2.1 Packet Delivery Ratio vs. Node Density

Figure 3.13: Packet delivery ratio vs. Node density --- group partition enabled

Figure 3.13 illustrates the changes in *packet delivery ratio* for AODV and DSR with respect to node densities. As it is shown, *packet delivery ratio* drops from 71% to 60% for AODV and from 68% to 54% for DSR when the node density increases from 200 *nodes/km²* to 400 *nodes/km²*. AODV yields an approximate 10% higher

packet delivery ratio than DSR. As discussed in the subsection 3.3.1, group coverage area is inversely proportional to the node density in the group. With a fixed number of mobile nodes, increasing node density in a group (where mobile nodes will be closer to other group members) will reduce the coverage area of the group. Therefore, links between connected groups may be broken because of the decrease of the group coverage areas which leads to a longer distance between those previously connected groups. As a result as illustrated in Figure 3.13, some data packets would not be delivered to the destinations via inter-group communications.

3.4.2.2.2 Average Control Packets per Data Packet Sent vs. Node Density

Figure 3.14 illustrates the changes of *average control packets per data packet sent* for AODV and DSR with respect to node densities. As the node density increases, group's coverage will be reduced as discussed in the subsection 3.3.1. As a result, less inter-group communications would be occurred and sources would not maintain as many route entries as that when more inter-group communications occur. Hence, the amount of maintenance control packets can be reduced when density is increased as shown in Figure 3.14. AODV uses nearly 50% of the control packets that are used in DSR because AODV only keeps one route entry whereas DSR has to maintain multiple route entries which requires more control packets in route maintenance as discussed in the subsection 3.4.2.1.3.

Figure 3.14: Average control packets per data packet sent vs. Node density

--- group partition enabled

3.4.2.2.3 End-to-end Delay vs. Node Density

Figure 3.15 presents the results which shows the *end-to-end delay* of AODV and DSR varied by node density. As discussed in the last subsection, when the node density increases, more links for inter-group communications may be broken (due to the coverage of groups become smaller which is inversely proportional to the node density). Therefore, more data packets would be lost which is illustrated in Figure 3.13 and retransmissions are required. Hence, more time will be spent on establishing new routes and it results in a higher *end-to-end delay* for both AODV and DSR. DSR generates a much higher *end-to-end delay* due to their different mechanisms to store and maintain route entries which has been discussed in the subsection 3.4.2.1.3.

Figure 3.15: End-to-end delay vs. Node density --- group partition enabled

3.4.3 Discussion

From the above comparisons and discussions, AODV shows its advantages in many aspects and outperforms DSR in group mobility network environment. The fundamental difference between AODV and DSR is the mechanism of maintaining routing cache tables. AODV keeps only one route entry in its routing cache. Whenever the route is stale, it will reinitiate a new *route recovery process*. However, in DSR, multiple route entries are kept in each route table, if any breakage of a link is detected, a maintenance process has to be triggered to repair the broken links first before a new route discovery takes place. In dynamic network environment, routes turn stale quickly. Thus, more control packets are generated during the route recovery operations in DSR and therefore it has a longer delay. AODV is more adaptive to the dynamic environment and it is able to response quickly to link breakage by reconstructing a new route.

When a group partition takes place, some connections may be broken and communications between sources and destinations are interrupted. It results in more packet loss and *packet delivery ratio* is reduced. As almost every routing protocol is implemented with a route recovery mechanism, it takes more control packets to repair the broken links which may prolong *end-to-end delay*. High speed movement can greatly affect network performance. Increase of node density can also weaken network performance since it can change the coverage of groups which affect the relative distance between connected groups in the network. AODV has proven itself as a more efficient network routing protocol than DSR in group mobility environment of MANETs.

In addition, some recent studies of the performance on AODV and DSR in group mobility models, such as in [71], concluded that routing protocols will not be affected significantly in group mobility with the findings that the received packets at the destinations are mainly due to intra-group communication and packets dropped are mainly due to the excessively long distance between two groups. These findings support our simulation results in packet delivery ratio of AODV and DSR in group mobility with group partition disabled as in Figure 3.7. It also demonstrates that the performances of DSR and AODV in RRGM perform better than those in RPGM because RRGM may have larger groups due to group mergers which introduce more reliable intra-group transmissions. This conclusion could also reflect the results in Figure 3.7 and 3.8 where the packet delivery ratio of routing protocols in RRGM is better than that in RPGM when the average speed is less than 10m/s. However, when the average speed is increased above 10m/s, the intra-group communication becomes less reliable and more group partitions will take place which degrades the performance of AODV and DSR in RRGM as shown

in Figure 3.8.

3.5 Conclusion

In this chapter, we carried out a number of experiments to study the impact of group partitions and mergers to the network performance of two typical reactive routing protocols, AODV and DSR. The network mobility patterns are generated by RRG model. AODV and DSR are compared by changing the speed and node density. Experimental results show that group partitions would have a significant impact on the network performance which has never been revealed before. The simulation results are captured in terms of *packet delivery ratio*, *average control packets*, and *end-to-end delay* for packet delivery. From these results, frequent group partitions can downgrade the performance of both AODV and DSR. However, AODV shows its advantages in tackling with such kinds of group operations better than DSR.

As a conclusion, it may be impossible to find a pervasive routing protocol that can be adaptive to any network environment. Every network routing protocol may only work well under some particular network circumstance. Selecting a suitable network routing protocol is very important for studying the operations and performance of MANETs. In this work, we demonstrate that AODV is a good choice to be used in group mobility with group partitions and mergers in MANETs. Moreover, AODV is also more adaptive to high speed environment. On the other hand, DSR is suitable to a network with less mobility where the load of route maintenance is not heavy. This study gives a deeper and better understanding in the network performance under the group mobility environment with group partitions and mergers.

CHAPTER 4: DISTRIBUTED GROUP MOBILITY ADAPTIVE CLUSTERING ALGORITHM

4.1 Introduction

Some people envisioned that a mobile ad hoc network (MANET) may possibly contain thousands of mobile nodes as well as connections [72]. The complexity of such a huge network grows exponentially as the number of mobile nodes increases. Thus a flat network structure poses scalability problems with respect to network routing and mobility management. If each node had a complete view of the entire network topology, the routing table maintained by each node would increase to an immense size. To reduce the network complexity, clustering algorithms are used to partition a network into a multi-level hierarchical structure which is composed of clusters. A cluster is a connected graph which is composed of a clusterhead (CH) and possibly a number of cluster member nodes. In a cluster structure, topology information is aggregated and routing messages are only exchanged at the higher hierarchical levels which are normally constructed by clusterheads. The higher hierarchical levels are responsible for the formation of clusters and the maintenance

of the network topology, whereas local cluster members only have the view of a fraction of the entire network. Clustering is a very important and effective approach of network management, especially for large scale dynamic networks, e.g. MANETs. By using clustering algorithms, the topology of a MANET can become “smaller” and “less dynamic”.

During the last two decades, various clustering schemes have been developed for traditional cellular mobile networks which rely on fixed infrastructures, such as based stations. However, since these fixed infrastructures are not available in MANETs, existing clustering schemes for cellular networks can not be applied to MANETs. Without any centralized controller, clustering schemes for MANETs are expected to work in a distributed, real-time and mobility-adaptive manner. A number of clustering schemes are proposed for MANETs to improve network performance in terms of cluster stability, scalability, bandwidth utilization, and resource allocation efficiency, e.g. [73-80]. However, most of the existing clustering schemes for MANETs are not mobility-aware or with the assumption of rare mobility. Clustering algorithms under this kind of assumption are not meaningful in dynamic network environment, e.g. in MANETs. Mobility is characterized as an inherent phenomenon in MANETs. Therefore, mobility is a significant factor to be considered in the study of clustering algorithms of MANETs.

Furthermore, we also look into the issue of group mobility with the occurrence of group partitions and mergers for our clustering algorithm. Group partitions and mergers are very common in many practical group mobility scenarios in MANETs, such as military operations in battlefields, disaster recovery and scientific

explorations. In these scenarios, mobile nodes may form groups and nodes in the same group may have a similar mobility pattern, which is known as group mobility in MANETs. Clustering algorithms can be used to organize the connectivity between group members. In group mobility of MANETs, group partitions and group mergers may cause the network topology to change. This will in turn affects the structures of clusters such that new clusters would need to be formed.

To address this concern, we propose in this chapter a *Distributed Group Mobility Adaptive* (DGMA) clustering algorithm for group mobility in MANETs. The motivation of DGMA is to provide an efficient clustering approach which can adapt to group mobility in MANETs to ensure maximum cluster stability. The advantages of DGMA can be emphasized based on the following four aspects.

Firstly, DGMA is a mobility-aware clustering algorithm which is suitable for dynamic network environment, e.g. in MANETs. Based on the idea of *spatial dependency* which is a mobility metric introduced in [8], we propose a novel concept, namely *linear distance based spatial dependency* (*LDSD*) as the mobility metric for DGMA. In *LDSD*, historical mobility information of mobile nodes are used instead of their instant velocities. By taking the advantage of *LDSD*, DGMA can also work under high speed network environment. Secondly, DGMA is a real-time, distributed algorithm which can be run at each node with the knowledge of local information. In DGMA, “micro” movement will be ignored. Thus, mobile nodes are not required to be static when clustering formation is in progress which is an assumption made by many existing clustering algorithms [81]. Thirdly, DGMA is adaptive to group partitions and mergers, which are the common group operations in MANETs. To the best of our knowledge, our proposed algorithm is

the first clustering algorithm which addresses the issue of group partitions and mergers in MANETs. Lastly, we use a coloring approach for clustering formation and maintenance in DGMA, which makes the clustering algorithm easy to be implemented. Various colors are used to represent nodes in different roles such as clusterheads and cluster members.

In DGMA, we assume every mobile node has the knowledge of its own location with the aid of positioning systems, such as GPS. The location information of every mobile node is processed by each individual node but it may not be necessary to share this information locally among its neighbors in order to reduce the network communication overhead. On the other hand, *Received Signal Strength Indication (RSSI)* [82] is a measurement of the power present in a received radio signal. Based on the received signal strength, the receiver could identify the distance to the senders. In DGMA, the RSSI propagation model is applied to identify a node's neighborhood. Thus, a mobile node could have a complete view of its neighborhood.

The rest of the chapter is organized as follows. In Section 4.2, we will review some existing work on clustering algorithms for MANETs. In Section 4.3, we will introduce our proposed mobility metric (i.e. *LDSD*). Next, the mechanism of the DGMA clustering algorithm will be discussed in Section 4.4. In Section 4.5, we evaluate the performance of DGMA under a group partitioning environment using the RRGm model. In particular, we compare the performance of DGMA with two other clustering algorithms, namely Lowest ID [83] and MOBIC [84]. In Section 4.6, we evaluate the performance of DGMA under the entity mobility environment. Finally, Section 4.7 is the conclusion of this work.

4.2 Literature Review

In this section, we will first review some existing clustering algorithms in the subsection 4.2.1. Next, a mobility metric, namely *spatial dependency*, will be introduced in subsection 4.2.2, which will be used as the underlying concept in our proposed mobility metric for DGMA in the next section.

4.2.1 Clustering Algorithms in MANETs

Over recent years, various clustering algorithms have been proposed for MANETs [72, 81, 84-116]. The Lowest-ID clustering algorithm [111] is a well-known and commonly used clustering algorithm in the studies of MANETs. In the Lowest-ID clustering algorithm, each node is assigned with a unique ID number. The Lowest-ID clustering algorithm is a distributed clustering algorithm in the sense that if a node has a smaller ID number than all of its directly connected neighbors, it can declare itself to be the clusterhead. When two clusterheads encounter with each other, the one with larger ID should give up the role of clusterhead and become a cluster member. The advantage of this clustering algorithm is its simplicity for implementation. On the other aspect, the major shortcoming of this algorithm is that it only considers the ID numbers of mobile nodes which is not meaningful practically. For example, it is not reasonable to select a node with minimum ID number to be the clusterhead, which may be prone to drain out of power, or has a mobility pattern which is very different from other mobile nodes.

To improve the performance of the Lowest-ID clustering algorithm, a number of its variants are proposed, which use different factors as each respective metric to form clusters, such as neighborhood connectivity, transmission power, and energy consumption. Among all these variants, *Maximum Connectivity Clustering*

algorithm [117], and *Least Clusterhead Change algorithm* [113] are used more often in the studies of network clustering schemes in MANETs.

In the Maximum Connectivity Clustering (MCC) algorithm [117], clusterhead election is based on degree of connectivity instead of node ID such that clusterheads are mobile nodes with “local highest node degree” as referred to in [117]. This algorithm can ensure the clusterhead has the maximum connectivity with cluster members. However, it may not be suitable for dynamic network environment where structures of cluster may change very frequently due to the dynamic network topology.

The Least Cluster Change (LCC) algorithm [113] defines two conditions which cause a clusterhead to change. Firstly, two clusterheads come within range of each other leading to a situation where someone has to give up its clusterhead position according to the Lowest-ID scheme or the maximum-connectivity scheme. Secondly, a node becomes disconnected from any cluster. This is an improvement (in stability) over two previous algorithms, namely Lowest-ID algorithm and MCC, which reelect the clusterhead every time the cluster membership changes. The LCC algorithm uses either Lowest-ID or highest-connectivity for initialization and maintenance. However, the key difference here is that when a non-clusterhead node moves into an already established cluster, it cannot challenge the current clusterhead. If, on the other hand, a clusterhead moves into an existing cluster, then it may or may not prevail based on node ID, connectivity or some other well defined priority. However if any cluster member moves out of the range of the clusterhead, it will cause the reelection of the clusterhead in this cluster. Hence, this

clustering algorithm also has a similar problem as MCC to tackle with dynamic network environment, where cluster membership changes very frequently.

MOBIC is a mobility-aware clustering scheme [84]. Instead of using node ID, MOBIC uses the *relative mobility*, M_Y^{rel} , as the mobility metric as shown in equation (4.1):

$$M_Y^{rel}(X) = 10 \log_{10} \frac{RxPr_{X \rightarrow Y}^{new}}{RxPr_{X \rightarrow Y}^{old}} \quad (4.1)$$

where the received signal power, $RxPr$, is used for estimating the distance between nodes X and Y ($RxPr_{X \rightarrow Y}^{new}$ represents the new distance between nodes X and Y , while $RxPr_{X \rightarrow Y}^{old}$ represents the old distance between nodes X and Y). Therefore, $M_Y^{rel}(X)$ can indicate whether two nodes are moving apart or closer to each other. In MOBIC, a node updates *relative mobility* with each of its neighbors periodically and calculates the variance of *relative mobility* to every neighbor, which will be aggregated. A small value of aggregated variance indicates that the mobile node is relatively less mobile with respect to its neighbors [84]. Accordingly, a mobile node with the smallest *relative mobility* among all of its neighbors assumes to be the clusterhead. In MOBIC, a “clusterhead contention interval” is defined to defer the contentions between two clusterheads when they are encountered [84]. MOBIC uses the *relative mobility* to reflect the relative changes in distance between two nodes as the clustering metric. However, MOBIC, as a mobility-aware clustering algorithm, does not consider node mobility parameters, e.g. speeds and moving directions. In addition, it is not suitable to be used for networks in group mobility scenarios. In group mobility, it is very common for two nodes staying very close to each other during a short interval. With respect to equation (4.1), $RxPr_{X \rightarrow Y}^{new}$ could

be 0 for the extreme case, which results $M_y^{rel}(X)$ to a minus infinite number. When this happens, MOBIC algorithm will become invalid. On the other aspect, when the network topology changes very fast in a dynamic network environment, the relative distance between mobile nodes would be changed frequently which results the clusters in the network become very unstable.

4.2.2 Spatial Dependency

Spatial dependency is a mobility metric based on relative motion in speed and direction. It is the extent of similarity of the velocities of two nodes that are not too far apart [8]. Formally,

$$D_{spatial}(i, j, t) = RD(\vec{v}_i(t), \vec{v}_j(t)) * SR(\vec{v}_i(t), \vec{v}_j(t)) \quad (4.2)$$

where $D_{spatial}(i, j, t)$ is the *spatial dependency* between node i and j at time t , $RD(\vec{v}_i(t), \vec{v}_j(t))$ is relative direction (cosine of the angle within $[0, \pi]$) between two vectors $\vec{v}_i(t)$ and $\vec{v}_j(t)$, and $SR(\vec{v}_i(t), \vec{v}_j(t))$ is the speed ratio between two vectors $\vec{v}_i(t)$ and $\vec{v}_j(t)$, which is given as:

$$SR(\vec{v}_i(t), \vec{v}_j(t)) = \frac{\min(|\vec{v}_i(t)|, |\vec{v}_j(t)|)}{\max(|\vec{v}_i(t)|, |\vec{v}_j(t)|)} \quad (4.3)$$

When two nodes move in similar directions with similar speeds ($RD(\vec{v}_i(t), \vec{v}_j(t))$ and $SR(\vec{v}_i(t), \vec{v}_j(t))$ can be approximately equal to 1), the *spatial dependency* between these two nodes i and j is high (the maximum value is 1). In the other aspect, if they move in opposite directions, they should have a smaller value for their *spatial dependency* (the minimum value is -1). Based on the definition of *spatial dependency* as in equation (4.2), we can easily deduce that with an equivalent number of neighbors, if a node's movement depends on other

neighbors, such as in group mobility, the summation of its *spatial dependency* on all neighbors must be greater than that of a node in an arbitrary mobility manner.

4.3 Mobility Metric - Linear Distance Based Spatial Dependency

We notice that the movement of groups may not be continuous in MANETs in the sense that mobile groups may temporarily be stationary at an area to carry out tasks. For example, in a disaster recovery scenario, rescuers perform their operations within an area; or a group of tourists visit an exhibition hall and they walk around a booth to take photos. However in these applications, when a group is stationary, the movement of its group members may not be interrupted. Instead, they may still move within a small area (e.g. a disaster site or a booth), and their movement could still be in an arbitrary manner, in which their speeds and directions may change frequently. The mobility metric, i.e. *spatial dependency* (which depends on the relative speeds and directions), will be affected. Therefore, *spatial dependency* is not suitable to be used as the mobility metric in the group mobility of MANETs.

To address this concern, we propose the *linear distance based spatial dependency (LDSD)* as the mobility metric for our proposed clustering algorithm. In *LDSD*, “micro movement” of a mobile node is discarded so that frequent changes in speeds and directions can be ignored. Each node’s movement is captured by its linear displacement (*LD*) which is used to justify whether a “significant movement” in physical locations is made for a node. Let $\Delta x(i,t)$ and $\Delta y(i,t)$ be the increment in x and y coordinates respectively for node i from the location $x(i, \tilde{T}(i))$ and $y(i, \tilde{T}(i))$ where the last “significant movement” was made at the time instant $\tilde{T}(i)$.

The linear displacement in x and y of the node i from $\tilde{T}(i)$ to the current time t can be calculated by :

$$\Delta x(i, t) = x(i, t) - x(i, \tilde{T}(i)) \quad (4.4)$$

$$\Delta y(i, t) = y(i, t) - y(i, \tilde{T}(i)) \quad (4.5)$$

where $x(i, t)$ and $y(i, t)$ represent the coordinates of node i at time t . Based on equations (4.4) and (4.5), $LD(i, t)$ for node i from time $\tilde{T}(i)$ to t can be calculated by:

$$LD(i, t) = \sqrt{\Delta x(i, t)^2 + \Delta y(i, t)^2} \quad (4.6)$$

We can define a node has made a “significant movement” only if it has changed its physical location by more than a threshold of distance LD_0 , i.e. $LD(i, t) > LD_0$. The speed $\tilde{S}(i, t)$ and direction $\tilde{\theta}(i, t)$ based on LD can be calculated by:

$$\tilde{S}(i, t) = \frac{LD(i, t)}{t - \tilde{T}(i)} \quad (4.7)$$

$$\tilde{\theta}(i, t) = \begin{cases} \varphi \cdot \text{sgn}(\Delta y(i, t)) & \Delta x(i, t) > 0 \\ \pi / 2 \cdot \text{sgn}(\Delta y(i, t)) & \Delta x(i, t) = 0 \\ (\pi - \varphi) \cdot \text{sgn}(\Delta y(i, t)) & \Delta x(i, t) < 0 \end{cases} \quad (4.8)$$

where $\tan \varphi = \left| \frac{\Delta y(i, t)}{\Delta x(i, t)} \right|$ and $\tilde{\theta}(i, t) \in (-\pi, \pi)$. The data set $[t, x(i, t), y(i, t),$

$\tilde{S}(i, t), \tilde{\theta}(i, t)]$ will be stored as the historical mobility information during the interval from $\tilde{T}(i)$ to t . “Micro movement”, which is less than LD_0 , will not be considered. Hence, frequent changes of mobility attributes, i.e. speeds and directions, can be ignored.

Based on historical mobility information, we propose a novel mobility metric, namely *linear distance based total spatial dependency (LDTSD)*, which can be calculated by the following steps:

Step 1: Every mobile node periodically broadcasts a *hello* message to announce its presence, which piggybacks its latest historical mobility information $\tilde{S}(i, t)$ and $\tilde{\theta}(i, t)$.

Step 2: Upon receiving the latest historical mobility information, the node calculates the values of *relative direction (RD)* and *speed ratio (SR)* with each respective neighbor from which the historical mobility information is received. $RD(i, j, t)$ between nodes i and j is the cosine of the angle between i and j at time t , which is calculated as:

$$RD(i, j, t) = \cos(\tilde{\theta}(i, t) - \tilde{\theta}(j, t)) \quad (4.9)$$

and $SR(i, j, t)$ between nodes i and j is given as:

$$SR(i, j, t) = \frac{\min(\tilde{S}(i, t), \tilde{S}(j, t))}{\max(\tilde{S}(i, t), \tilde{S}(j, t))} \quad (4.10)$$

Step 3: According to the equation (4.2), $LDSD$ between nodes i and j at time t can be calculated by:

$$LDSD(i, j, t) = RD(i, j, t) * SR(i, j, t) \quad (4.11)$$

Step 4: Assuming node i has received $LDSD$ from n neighboring nodes, the node i calculates the summation of all $LDSD$ it has received from its neighbors, which is referred to as $LDTSD$, by:

$$LDTSD(i, t) = \sum_{j=1}^n LDSD(i, j, t) \quad (4.12)$$

A larger value of *LDTSD* implies that the node *i*'s mobility is relatively more dependent on its neighbors. In other words, the node *i* has a relatively higher degree of similarity in mobility with its neighbors.

4.4 Distributed Group Mobility Adaptive Clustering Algorithm

In this section, we will introduce our proposed clustering scheme which is especially apt for group mobility pattern in MANETs. The expectations and objectives of the scheme focus on the following aspects:

- a. Prolong the lifetime of each cluster;
- b. Adapt to high speed environment with frequent network topology changes;
- c. Reduce the number of clustering iterations, i.e. changes of clusterheads and cluster members' affiliations.

Figure 4.1 illustrates a network in group mobility pattern and its topologies in clusters formed by DGMA. These figures are varied by node densities in mobile groups. In the figures, the squares represent clusterheads and the circles represent cluster members and idle nodes. Solid line is the link between a clusterhead and a cluster member. A cluster member can be connected to more than one clusterhead which indicates it belongs to two different clusters simultaneously. Those nodes that are not directly connected with any clusterhead are considered as idle nodes. DGMA is implemented by employing a coloring mechanism which assigns specific colors, namely red, yellow and white, to nodes in different roles. Clusterheads are colored with red while cluster members are in yellow. White represents idle nodes which either in the initial state or in a distance of more than one hop from any

clusterhead. From Figure 4.1, it can be seen that when the node density of a group increases, the number of clusters is reduced.

Figure 4.1: Examples of clusters in different node densities formed by DGMA

Every node including clusterheads, cluster members and idle nodes maintain their respective status and clustering affiliations. Every node can leave or join a cluster anytime based on their physical locations. Each clusterhead has a complete view of its cluster membership. In DGMA, it is possible for a mobile node to belong

to a number of clusters simultaneously when it can connect to more than one clusterhead. Every node has a *membership_table*, which is used to maintain its connectivities. The *membership_table* contains two fields: *membership_table.CH* and *membership_table.member*. The field of *membership_table.CH* keeps the IDs of all the clusterheads that a cluster member connects to; while the field *membership_table.member* is for a clusterhead to log all its neighbors in a format of “<nodeID><nodecolor>”. Every clusterhead is responsible to maintain the field of *membership_table.member* with the periodical *hello* messages received. Every cluster member tracks all the connected clusterheads in the field of *membership_table.CH* to indicate which clusters it enrolls in. An idle node does not have to maintain the *membership_table* except setting the two fields to blank whenever it turns to be idle (white).

Next, we will introduce terminologies which will be used in DGMA.

- 1) *T-contention(clusterhead contention interval)*: it represents the duration that two red nodes, i.e. clusterheads, are directly connected.
- 2) *Ratio-n*: a ratio of one-hop neighbors in white over total number of one-hop neighbors (its usage will be specified in the following parts of this section):

$$Ratio - n = \frac{\text{number of one - hop neighbors in white}}{\text{total number of one - hop neighbors}} \quad (4.13)$$

- 3) n_1 : *Ratio-n* of a red node.
- 4) n_2 : *Ratio-n* of a white node.
- 5) n_{1_Input} : a threshold of n_1 (an input parameter).
- 6) n_{2_Input} : a threshold of n_2 (an input parameter).

DGMA clustering algorithm contains two main processes: the *nomination process* and the *cluster maintenance process*, which will be discussed in the following subsections.

4.4.1 The Nomination Process

In this subsection, we will introduce the *nomination process* which is carried out in five steps:

Step 1: Each node is initially colored in white.

Step 2: A white node, which is not connected with any red node, will broadcast a *nomination_request* to its directly connected neighbors in order to trigger a *nomination process*.

Step 3: Upon receiving the *nomination_request* from an applicant (the white node who initiates the *nomination process*), neighboring white nodes feed back each respective *LDTSD*. The applicant compares the received *LDTSD* with its own *LDTSD*.

Suppose node A initiated a *nomination_request*, which is received by node B.

- a. if $LDTSD(A, t) < LDTSD(B, t)$, the nomination process at node A can be terminated and node A remains in white;
- b. if $LDTSD(A, t) > LDTSD(B, t)$, node A continues comparing its *LDTSD* with other white neighbors.

Step 4: if node A has the largest *LDTSD* among all of its white neighbors, it turns to red to form a new cluster labeled with the ID number of node A as the cluster ID. Node A updates its cluster *membership_table* with ID and color of its neighbors, from whom node A received the *LDTSD*.

Step 5: node A broadcasts an *invitation_message* to its neighbors. The receivers in white would change their color to yellow and update the *membership_table* by adding the cluster ID to the *membership_table.CH* field. Note, if a cluster member node affiliates with multiple clusters simultaneously, there would be more than one cluster ID stored in the field of *membership_table.CH*.

The pseudo-code for the nomination process is given in Figure 4.2.

```

PROCEDURE Nomination ( $\mu$ );
BEGIN
  Precondition:
    Input node:  $\mu$ 
    Neighborhood set of  $\mu$ :  $\Theta$ 
    membership_table.CH : list of connected CHs
    membership_table.member : list of cluster members
  Effect:
    if  $\mu$ .color is white
      for each  $v \in \Theta$ 
        if  $\forall v$  that  $v$ .color is white,  $\mu$ .LDTSD >  $v$ .LDTSD is always true
           $\mu$ .color = red;
          set u.membership_table.CH to blank;
          update u.membership_table.member with  $v.ID < color >$  for all  $v$ 
            in  $\Theta$ ;
        endif
      endfor
    if  $\mu$ .color is red
      for each  $v \in \Theta$ 
        if  $v$ .color is white
           $v$ .color = yellow;
          add u.ID into v.membership_table.CH
        endif
      endfor
    endif
  END

```

Figure 4.2: Protocol specification of the *nomination process*

4.4.2 The Maintenance Process

In this subsection, we will describe the *maintenance process* in DGMA. For the maintenance process executed at each node, different maintenance steps are executed based on the node colors of red, yellow and white separately.

- i) For red nodes, i.e. clusterheads:
 - a) A red node updates its neighbor information in *membership_table.member* by receiving *hello* messages.
 - b) If the red node is connected with another red node over a time interval of *T-contention*, the one with smaller *LDTSD* turns into yellow. Both of them will update their own *membership_table*. The red node adds the yellow node's ID and color into its *membership_table.member* list; while the yellow node resets its *membership_table.member* to blank and updates its *membership_table.CH* with the red node's ID.
 - c) For the red node, if *Ratio-n* is greater than $n_{I-Input}$, i.e. $n_I \geq n_{I-Input}$, it turns into white and resets its *membership_table.member* to blank.
- ii) For yellow nodes, i.e. cluster members:
 - a) If the *LDTSD* of a yellow node is greater than the *LDTSD* of any clusterhead it directly connects with, it turns into white and resets its *membership_table.CH* to blank. It piggybacks its color in the *hello* message and red nodes receiving this message will update the color of this node in their *membership_table.member*.
 - b) If it fails to find any neighboring red node, it turns into white and resets its *membership_table.CH* to blank.

iii) For white nodes, i.e. idle nodes:

- a) If a white node finds a red node among its neighbors that has a larger *LDTSD* than itself, it turns to yellow and update its *membership_table.CH* with the red node ID. The red node also adds the yellow node ID into its *membership_table.member*;
- b) Else, if *Ratio-n* of a white node is greater than $n_{2-Input}$, i.e. $n_2 \geq n_{2-Input}$, it triggers a new cluster *Nomination* process.

The pseudo-code for the maintenance process is given in Figure 4.3.

```

PROCEDURE Maintenance ( $\mu$ );
BEGIN
  Precondition:
    Input node:  $\mu$ 
    Neighborhood set of  $\mu$ :  $\Theta$ 
    membership_table.CH : list of connected CHs
    membership_table.member : list of cluster members
  Effect:
    if  $\mu$ .color is red
      update u.membership_table.member;
      for each  $v \in \Theta$ 
        if  $\exists v$ .color is red & connecting_time ( $\mu, v$ ) > T-contention
          if  $\mu$ .LDTSD <  $v$ .LDTSD
             $\mu$ .color = yellow;
            reset u.membership_table.member to blank;
            update u.membership_table.CH with  $v$ .ID;
            update  $v$ .membership_table.member with  $u$ .ID<color>;
          else
             $v$ .color = yellow;
            reset  $v$ .membership_table.member to blank;
            update  $v$ .membership_table.CH with  $u$ .ID;
            update u.membership_table.member with  $v$ .ID<color>;
        endif
      endif

```



```

(continued)
    else if  $n_1 \geq n_{1-Input}$ 
         $\mu.color = white$ ;
        reset  $u.membership\_table.member$  to blank;
    endif
endfor
endif
if  $\mu.color$  is yellow
    for each  $v \in \Theta$ 
        if that  $\exists v.color$  is red is false
             $\mu.color = white$ ;
            reset  $u.membership\_table.CH$  to blank;
        else if  $\forall v$  that  $v.color$  is red,  $\mu.LDTSD > v.LDTSD$  is true
            remove  $v.ID$  from  $u.membership\_table.CH$ ;
            if  $u.membership\_table.CH$  is blank
                 $\mu.color = white$ ;
            endif
        endif
    endfor
endif
if  $\mu.color$  is white
    for each  $v \in \Theta$ 
        if  $\exists v.color$  is red &  $\mu.LDTSD < v.LDTSD$ 
             $u.color = yellow$ ;
            update  $u.membership\_table.CH$  with  $v.ID$ ;
            update  $v.membership\_table.member$  with  $u.ID<color>$ ;
        endif
    endfor
else if  $n_2 \geq n_{2-Input}$ 
     $Nomination(\mu)$ ;
endif
END

```

Figure 4.3: Protocol specification of maintenance process

4.4.3 Analysis of the DGMA Algorithm

In this subsection, the correctness and complexity of DGMA algorithm will be analytically discussed.

Lemma 1: Let μ and v be a pair of white nodes, Θ_u be the neighborhood set of a node μ , and $v \in \Theta_u$. If $\mu.color=red$ at the end of *nomination process*, then $v.color=yellow$.

Proof: Assume that $v.color \neq yellow$. That means node v may be either in red or in white. We discuss it in two cases separately.

Case 1: If $v.color=red$. Since $\mu.color=red$, and $v \in \Theta_u$, hence $\mu.LDTSD > v.LDTSD$. Similarly, because $v.color=red$, and $v \in \Theta_u \Rightarrow \mu \in \Theta_v$, hence $\mu.LDTSD < v.LDTSD$, which is a contradiction. Note that although in the maintenance process, two red nodes may directly connect temporally to avoid frequent clusterhead re-elections, this would not happen in the initial *nomination process*.

Case 2: If $v.color=white$. Since $v \in \Theta_u$ and $\mu.color=red$, according to the clustering protocol of the *nomination process*, $v.color=yellow$ is executed eventually. Therefore, the assumption is false.

Based on the discussions in **Case 1** and **Case 2**, node v 's color can not be either red or white. Hence, **Lemma 1** is proved.

Lemma 2: DGMA has a worst case processing time complexity of $O(n)$ per node, where n is the number of nodes in the network.

Proof: In the cluster nomination phase, the worst case happens when all nodes in the network are in the neighborhood set of a node μ assuming that μ is finally selected as the clusterhead. It takes a processing time of at most $2(n-1)$. Similarly, for the phase of cluster maintenance, it takes a processing time of at most $(n-1)$ at

each of red and yellow nodes and $2(n-1)$ at white node for the worst case. Therefore, the total processing complexity is $O(n)$.

Lemma 3: DGMA has a worst case communication (message exchange) complexity of $O(1)$ at each node and for a network with n nodes, the communication complexity for the worst case is $O(n)$.

Proof: During every maintenance iteration, each node broadcasts a message which contains the information of node ID, color and $LDTSD$. Only the newly elected red nodes broadcast an additional *invitation_message*. Thus, the complexity is exactly $O(1)$ at each node. The number of messages has a upper bound of $O(n)$ in a network of size n .

Lemma 4: For a connected graph $G(V,E)$, $|V| \geq 2$, there exists at least one clusterhead.

Proof: We will use the method of mathematical induction to prove this lemma.

Basis: For $|V|=2$, assume that $V=\{A, B\}$. If $\max.LDTSD(A, B)=A$, A becomes the clusterhead; otherwise B is the clusterhead. Thus the statement holds when $|V|=2$.

Inductive step: Assume for $G=(V, E)$ and $|V|=n$, there exists at least one clusterhead.

When $|V|=n+1$, let $G'=(V', E')$ be a subgraph of G where $V'=(1, 2, 3, \dots, i, \dots, n)$ and let node i be a clusterhead in G' . Thus $\max.LDTSD(V'_j | j=1, 2, \dots, n)=i$. For the node $n+1$, we discuss it in two cases.

Case 1: If node i and node $n+1$ are directly connected neighbors, and $\max.LDTSD(i, n+1)=n+1$, we can derive $\max.LDTSD(V''_j | j=1, 2, \dots, n, n+1)=n+1$ and the $(n+1)^{th}$ node becomes a clusterhead.

Case 2: If node i and node $n+1$ are not directly connected neighbors, node i is still the clusterhead.

Hence, it follows by induction that **Lemma 4** is true.

Lemma 5: For a complete graph, there is exactly one clusterhead after the cluster *nomination process*..

Proof: In a complete graph, $G=(V, E)$, each vertex has a degree of $n-1$, and $|E|=n(n-1)/2$, which means every pair of vertices are adjacent. It follows from **Lemma 4** that there must be at least one clusterhead in a complete graph. Next, we suppose there are two clusterheads, e.g. nodes A and B . Since both $A, B \in V$, A and B is an adjacent pair of vertices in V . The node with larger $LDTSD$ is assigned as clusterhead. Hence, it is easy to deduce that the statement in **Lemma 5** is true.

Observation: After the cluster *nomination process* is completed, if a white node is surrounded by only yellow nodes without any other white or red nodes, this white node is in two hops distance to the nearest clusterhead through its directly connected yellow nodes. (Note: if more than 2 red nodes are in two hops distance from it, this white node functions like a gateway node.)

Lemma 6: After the cluster *nomination process* is completed, the maximum distance from a node to its nearest clusterhead is two hops.

Proof: Consider a two-hop connection $A \sim B \sim C$ which means node A and B , B and C are directly connected neighbor pairs whereas A and C are not connected to each other. Assume $LDTSD.A > LDTSD.B > LDTSD.C$, after cluster *nomination process*, hence node A becomes red node and B turns to yellow, whereas node C still keeps as white according to the *nomination process*. Thus, the maximum

distance for a connection between a node and the clusterhead is within two hops. If there is a connection more than two hops, e.g. $A \sim B \sim C \sim D$, if node A is the clusterhead, node B would be yellow. According to Lemma 4, two connected white nodes can not coexist after the cluster *nomination process* is completed and the one with greater *LDTSD* will be elected as the clusterhead. Thus for this case, either node C or node D will finally turn to red. The maximum distance from a node to the nearest clusterhead is still within two hops distance. Similarly, assume there are a series of connections $A \sim B \sim \dots \sim C \sim D$ in which the maximum distance from a node to a clusterhead is two hops. We try to add another white node E into the connections. If the white node E is connected with a yellow node, the distance from node E to a clusterhead is two (the distance from a yellow node to a clusterhead is one hop); else if node E is connected with a white node, one of the white nodes will eventually be turned to be a clusterhead. Thus, we can deduce that the statement in **Lemma 6** is true for any number of connections.

4.5 Simulation and Discussions on DGMA under Group Mobility Pattern

Our proposed algorithm is implemented in a C++ environment. The nodes' mobility is generated with the Reference Region Group Mobility (RRGM) model. The mobility traces of nodes are captured and are used as input for the clustering algorithm. We compare the performance of our proposed algorithm, i.e. DGMA, with two existing clustering algorithms, namely MOBIC and Lowest-ID clustering algorithms. Four independent sets of experiment are conducted and each experiment is run for 20 times. The results of each run are averaged eventually. The

simulation is executed for 1500 seconds. 40 identical mobile nodes are deployed in a terrain of 1500m by 1500m.

In the following parts of this section, we will first introduce the experimental environment including the specifications of the mobility model, namely RRGGM, and the configuration of DGMA. Next, comparisons and analysis are given for the DGMA clustering scheme.

4.5.1. Mobility Model

The Reference Region Group Mobility model is selected as the mobility model because it is a generic model to describe group mobility in MANETs as introduced in Chapter 2. In particular, it can also simulate the operations of group partitions and mergers. Based on this model, the performance of clustering algorithms under group mobility environment can be studied. The configuration parameters of RRGGM model are listed in Table 4.1. (Note, those parameters that have been introduced in Chapter 2 Table 2-1 but not listed in Table 4.1 are in their default values and they should not have any significant impact on the simulation results).

Table 4.1: Mobility parameters

PARAMETERS	DEFAULT VALUE
NUMBER-OF-NEW-DESTINATIONS	2
NODE-DENSITY	1 node/(100m) ²
AVERAGE-NODE-SPEED	10.0 m/s
TRANSMISSION-RANGE	200 m
PARTITION-TIME INTERVAL	50.0 s

In RRGGM, all nodes start in a big group initially. Group partitions may take place every 50 seconds specified by “PARTITION-TIME-INTERVAL”, and the number of newly generated subgroups due to group partitions is controlled by the parameter of “NUMBER-OF-NEW-DESTINATIONS”. Group mergers may happen anytime

according to the mechanism of RRGGM. Node density in a group is $1 \text{ node}/(100\text{m})^2$. The average node speed is 10m/s which falls in a range of (5m/s, 15m/s). The radio transmission range of mobile nodes is 200 meters.

4.5.2. Configurations of DGMA

Table 4.2 shows the control parameters for DGMA. “CLUSTER-MOBILITY-DISTANCE” in Table 4.2 is corresponding to LD_0 as introduced in Section 4.3. It is the threshold of linear distance over which a node should update its LD based mobility information. For instance, if LD_0 is configured to 100m, a mobile node will periodically measure its physical linear displacement every second from the location where it has last updated its mobility information. If LD is greater than 100m, the node should update its mobility information derived from the linear displacement. The “CLUSTER-MAINTAINANCE-n1” parameter, which is used by clusterheads, represents the ratio between the number of neighboring white nodes and the total number of directly connected neighbors. If the ratio is greater than n_1 , the clusterhead has to turn to a white node (idle node). In our experiment, 0.5 is assigned as the default value for this parameter. With this value, if more than 50% members of a cluster have left the cluster, the corresponding clusterhead will change its role to a white node as it is not qualified to be the clusterhead. “CLUSTER-MAINTAINANCE-n2”, which is used by a white node, represents the ratio between the number of neighboring white nodes and the number of all neighboring nodes. If the ratio is greater than n_2 , then the cluster *nomination process* will be triggered. “CLUSTER-MAINTAINANCE-red-red” defines the “contention interval” for two connected red nodes (i.e. clusterheads), after which they have to compete for the role of clusterhead.

Table 4.2: Clustering parameters

PARAMETERS	DEFAULT VALUE
CLUSTER-MOBILITY-DISTANCE	100 m
CLUSTER-MAINTAINANCE-n1	0.5
CLUSTER-MAINTAINANCE-n2	0.5
CLUSTER-MAINTAINANCE-red-red	50s

4.5.3. Performance Evaluations of DGMA

We simulate and compare DGMA clustering scheme with MOBIC and Lowest-ID clustering schemes by investigating cluster stability with respect to the node's transmission ranges, node's mobility speeds, group partition intervals and node densities.

- **Node's transmission ranges:** node's transmission ranges can affect the size of a cluster which in turn makes an impact on the topology of clusters formed by different clustering algorithms.
- **Node's mobility speeds:** node's mobility speeds can affect the stability of clusters in that when node's mobility speeds are increased, the network would become more dynamic; particularly in the group mobility environment, more group partitions and mergers will take place, which will affect the stability and topologies of clusters formed by different clustering algorithms.
- **Group partition intervals:** Group partition is an inherent phenomenon in group mobility of MANETs. When group partitions take place more frequently, the network topology will become more dynamic and the structures of clusters formed by different clustering algorithms would be impacted.
- **Node densities:** Changes of the node densities will result in the changes of cluster's radius as illustrated in Figure 4.1. As a result, the connectivities

between cluster members and cluster's topologies will be affected, which will make an impact on the performance of clustering algorithms.

For DGMA, the control parameters n_1 and n_2 are varied from 0.3 to 0.5, and are represented as DGMA-0.3, DGMA-0.4, DGMA-0.5 respectively. We will study and evaluate DGMA as well as other two clustering schemes from four aspects which can reflect the stability of clusters formed by each respective clustering scheme, namely *Mean lifetime of CHs(clusterheads)*, *Mean resident time*, *Average number of CH changes*, and *Average number of cluster reaffiliations*.

- *Mean lifetime of CHs*, which is also known as cluster lifetime, is calculated by accumulating the lifetime duration of each CH divided by the total number of CHs, namely n , during a simulation, as elaborated in equation (4.14). A longer *Mean lifetime of CHs* implies a longer period of time a cluster can survive.

$$\text{Mean lifetime of CHs} = \frac{\sum_{i=1}^n \text{lifetime of } i}{n} \quad (4.14)$$

- *Mean resident time* of a node is the average time interval that a node affiliates with a certain cluster [90] as indicated by equation (4.15). It reflects the average duration that a cluster member stays in a cluster, which is calculated by accumulating the total resident time of all residents in clusters divided by the total number of residents in the simulation.

$$\text{Mean resident time} = \frac{\sum_{i=1}^n \text{resident time of } i}{n} \quad (4.15)$$

- *Average number of CH changes* indicates the number of clusterhead elections normalized over the simulation time. A larger value of this metric denotes frequent clusterhead replacement in the network. It is affected by the schemes used for selecting clusterheads, and also the mobility pattern of a network. Most

clustering algorithms aims to reduce the number of clusterhead changes so that a cluster could last for a longer period of time for better stability.

- *Average number of cluster reaffiliations* is the summation of the total number of mobile nodes joining into a cluster and the total number of clusterhead elections normalized by the simulation time. This metric can reflect the overall overhead cost of a clustering algorithm in maintaining the hierarchical cluster structure.

In a dynamic network, more cluster reaffiliations would be generated than that in a static network.

4.5.3.1 Varying Transmission Ranges

(a). Mean lifetime of CHs vs Transmission range

(b). Mean resident time vs Transmission range

(c). Average number of CH changes per sec. vs Transmission range

(d). Average number of cluster reaffiliations per sec. vs Transmission range

Figure 4.4: Performance metrics vs. Transmission ranges

Node transmission range is varied from 100m to 300m incremented by 50m each time. As shown in Figure 4.4(a), when the transmission range is increased from 100 meters to 300 meters, the *mean lifetime of CHs* obtained by DGMA also increases. In particular, the *mean lifetime of CHs* obtained by DGMA-0.3 increases from 14 seconds to 23 seconds. As the transmission ranges are increased, the *mean lifetime of CHs* obtained by DGMA-0.4 and DGMA-0.5 also exhibits in a similar pattern which is higher than DGMA-0.3 for about 5 seconds and 10 seconds on average respectively. On the contrary, the *mean lifetime of CHs* obtained by Lowest-ID grows slowly and its *mean lifetime of CHs* is only comparable to that of DGMA-0.3. The *mean lifetime of CHs* obtained by MOBIC drops dramatically when the transmission ranges are increased. In general, when transmission range is expanded, the radius of a cluster may also be enlarged which in turn results the cluster possibly having more mobile nodes. If the criteria to select clusterheads is not dependent on nodes' mobility (for the case of Lowest-ID), or not very sensitive to node instant mobility (for the case of *LDSD*), the lifetime of clusterheads would not be affected significantly. For example, in an extreme case, when a node's transmission range can cover the whole network, the clusterhead elected by Lowest-ID would not change in the sense that the one with smallest ID is always the clusterhead. If the membership in a cluster is rarely changed and their mobility is steady without apparent changes, clusterheads are not likely to change frequently and therefore they can have longer lifetime. However, since MOBIC relies on instant mobility information, the *relative mobility* between nodes is updated frequently although it may not be necessary, e.g. the cluster is stationary at an intermediate point only with node's local movement within the range of the cluster. In addition, when transmission range is enlarged, more mobile nodes will be consisted within a

cluster. Difference in mobility behaviors between clusterheads and cluster members may increase their *relative mobility* dramatically, which results in more replacement of clusterheads. Thus, the lifetime of clusterheads obtained by MOBIC drops when the transmission range is increased.

Figure 4.4(b) shows the *mean resident time* for each clustering scheme. With the increase of transmission range, the *mean resident time* obtained by each clustering algorithm appears in a similar increasing trend. In particular, the *mean resident time* obtained by DGMA-0.3 increases from 11 seconds to 22 seconds as the transmission ranges are raised from 100 meters to 300 meters. DGMA-0.4 and DGMA-0.5 yield a longer *mean resident time* than DGMA-0.3 for 2 to 4 seconds and 4 to 8 seconds on average respectively. However, both Lowest-ID and MOBIC clustering algorithms obtain a shorter *mean resident time* than DGMA-0.3, with one exception that MOBIC surpasses DGMA-0.3 for 1 second when the transmission range is 100 meters. In DGMA, if a cluster member has a larger *LDTSD* compared with other neighboring nodes, it will not trigger the *nomination process* immediately. Instead, it turns itself into a white node so that it will not affect other cluster members. On the other hand, a change in *relative mobility* of a cluster member in a cluster formed by MOBIC may trigger a clusterhead *nomination process* immediately. As a result, DGMA yields a longer *mean resident time* than MOBIC and Lowest-ID.

Figure 4.4(c) illustrates the *average number of CH changes* for each clustering algorithm. When the transmission ranges are increased from 100 meters to 300 meters, the *average number of CH changes* also decreases for each of the clustering algorithms. In particular, the *average number of CH changes* obtained by Lowest-ID decreases at a faster rate from 0.7 to 0.1 as compared to others. The

average number of CH changes obtained by DGMA-0.3, DGMA-0.4, DGMA-0.5 and MOBIC decrease from 0.87, 0.68, 0.49 and 0.57 to 0.21, 0.18, 0.15 and 0.2 respectively. When the transmission ranges are increased, larger clusters in terms of number of cluster members are formed in the network and a node with a lower ID could maintain its role for a longer period of time in the cluster formed by Lowest-ID. For example, a node with the lowest ID in a cluster would always be the clusterhead if the cluster's membership is fixed. However, the clusterhead may be changed when the mobility metrics are updated in DGMA and MOBIC even if the membership in the cluster is fixed. Therefore, DGMA and MOBIC yield more number of CH changes than that of Lowest-ID when the transmission range is larger than 250 meters. It is also notable DGMA-0.4 and DGMA-0.5 achieve less CH changes than that of Lowest-ID and MOBIC with a smaller transmission range, e.g. less than 200 meters.

Figure 4.4(d) illustrates the *average number of cluster reaffiliations per second* obtained by each clustering algorithm. When the transmission ranges are increased from 100 meters to 300 meters, the *average number of cluster reaffiliations per second* obtained by each clustering algorithm appears in a decreasing trend, which decreases from 4.3, 3.8 and 3.2 to 3.0, 2.5 and 2.1 for DGMA-0.3, DGMA-0.4 and DGMA-0.5 respectively, and from 4.8 and 3.5 to 3.6 and 2.2 for MOBIC and Lowest-ID respectively. From the results, MOBIC generates more cluster reaffiliation events than others in the sense that mobile nodes may join and leave a cluster more frequently, which reflects the fact that cluster members in a cluster formed by MOBIC are less dependent on their respective clusterheads. The *average number of cluster reaffiliations per second* obtained by Lowest-ID is comparable to

that of DGMA-0.4 when the transmission range is larger than 250 meters. DGMA-0.5 yields the smallest number of cluster reaffiliation events.

From the above analysis and experimental results, DGMA adapts to the increase of transmission ranges better than MOBIC and Lowest-ID.

4.5.3.2 Varying Speeds

In the second set of our experiment, node's average speed is varied from 5m/s to 20m/s, incremented by 5m/s each time. Figure 4.5(a) illustrates the *mean life time of CHs* for each of the clustering algorithms. With the increase of the speeds, the *mean life time of CHs* of each clustering algorithm decreases. In particular, the *mean life time of CHs* obtained by Lowest-ID decreases faster than others from 42 seconds to 9 seconds, while that of MOBIC drops slowly from 32 seconds to 24 seconds as the speed is increased from 5m/s to 20m/s. When the speed increases, network becomes more dynamic and group partitions may take place more frequently. As a result, the *mean life time of CHs* of each clustering algorithm decreases. However, MOBIC benefits from a "contention interval" which is defined to avoid frequent clusterhead replacement when two clusterheads are directly connected (the contention interval is set to 10 seconds in the simulation). With the aid of this "contention interval", the *mean life time of CHs* could be prolonged in MOBIC and it yields a longer *mean life time of CHs* than others when the speed is up to 20m/s.

(a) Mean lifetime of CHs vs. Speeds

(b) Mean resident time vs. Speeds

(c) Average number of CH changes per sec. vs. Speeds

(d) Average number of cluster reaffiliations per sec. vs. Speeds

Figure 4.5: Performance metrics vs. Speeds

Correspondingly as illustrated in Figure 4.5(b), the *mean resident time* obtained by Lowest-ID also drops faster (from 31 seconds to 7 seconds) than others when mobility speed is increased from 5m/s to 20m/s due to the fast decline in the *mean lifetime of CHs* obtained by Lowest-ID. With the increase of the speeds, MOBIC yields a lower *mean resident time* (which decreases from 21 seconds to 11 seconds) than that of DGMA-0.5, DGMA-0.4 and DGMA-0.3 (which decrease from 38 seconds, 34 seconds and 28 seconds to 14 seconds, 12 seconds, and 11 seconds respectively). This is due to the fact that MOBIC generates more cluster reaffiliation events which will be discussed with the Figure 4.5(d) in the following paragraphs. Hence, from these results we know that DGMA can ensure a longer

mean resident time for cluster members than MOBIC and Lowest-ID especially in a dynamic network environment with high mobility speeds.

Figure 4.5(c) illustrates the *average number of CH changes per second* by varying the speeds from 5m/s to 20m/s. The *average number of CH changes per second* obtained by Lowest-ID increases dramatically from 0.15 to 0.61 while that of MOBIC increases slowly from 0.28 to 0.34 as compared to others. When nodes move faster, the network topology will also become more dynamic and group partitions and mergers would take place more frequently. Therefore, the number of CH changes would be increased which is illustrated by Figure 4.5(c). However, since a “contention interval” for clusterheads is defined in MOBIC, re-clustering is deferred until the “contention interval” has expired [84]. Therefore, MOBIC yields a slower increasing rate in the *average number of CH changes per second* than the other two clustering algorithms.

Figure 4.5(d) illustrates the *average number of cluster affiliations per second* by varying the speeds from 5m/s to 20m/s. From this figure, the *average number of cluster affiliations per second* of each clustering algorithm appears in an increasing trend. In particular, DGMA-0.5 yields the smallest amount of *average number of cluster affiliations per second* than others which increases from 1.2 to 5.1 and DGMA-0.4 yields a little higher amount of the *average number of cluster affiliations per second* than DGMA-0.5 which increases from 1.4 to 6.1. When the speed is less than 15m/s, MOBIC generates more cluster reaffiliation events compared to Lowest-ID and DGMA-0.3. As the speed is increased to 20m/s, MOBIC, Lowest-ID and DGMA-0.3 yield an approximately equivalent amount of cluster reaffiliation events for about 7.1 to 7.2 times per second. In MOBIC, if any

node joins into a new cluster, re-clustering will not be triggered and when two clusterheads are directly connected, re-clustering will also not be triggered before the expiration of the “contention interval”. Thus, when group mergers take place more frequently in the network as the result of increasing mobility speed, multiple clusterheads may co-exist in one group which is formed by a number of subgroups in MOBIC. Therefore, nodes in this group will affiliate with multiple clusters simultaneously, which results in more cluster reaffiliation events being generated in MOBIC. However, when a group merger between two subgroups takes place in a network clustered by Lowest-ID and if the two clusterheads are directly connected, the one with a higher node ID will give up its role immediately. On the other hand, when a group merger between two subgroups takes place in a network clustered by DGMA, re-clustering may take place based on the mobility metric of every cluster member. Thus, the chance of co-existence of CHs within a group in Lowest-ID and DGMA is not as high as that in MOBIC. Therefore, as it is reflected in the figure, MOBIC generates more cluster reaffiliation events than DGMA and Lowest-ID when the speed is increased.

We note that DGMA-0.4 and DGMA-0.5 outperform Lowest-ID and MOBIC in tackling speed changes and DGMA-0.3 is also very competitive as shown by the simulation results in Figure 4.5. MOBIC as a kind of mobility adaptive clustering algorithm shows its capability to prolong the lifetime of CHs but at the expense of more cluster reaffiliation events and shorter resident time for cluster members. Lowest-ID is apparently not suitable in dynamic environment as it is not able to alleviate the impact caused by network mobility.

4.5.3.3 Varying Partition Intervals

(a). Mean lifetime of CHs vs. Partition intervals

(b). Mean resident time vs. Partition intervals

(c). Average number of CH changes per sec. vs. Partition intervals

(d). Average number of cluster reaffiliations per sec. vs. Partition intervals

Figure 4.6: Performance metrics vs. Group partition intervals

In a dynamic group mobility environment, group partitions would make an impact on the stability of clusters. Interval of group partitions is one of the factors that can affect the performance of a clustering algorithm. Unfortunately, to the best of our knowledge it has not been considered and studied in the existing works so far. We address this concern in our work so as to study the impact of group partitions to cluster's stability. In group mobility of MANETs, a short group partition interval will cause the network topology to change more frequently, which in turn affects the connectivities among mobile nodes. When a group partition takes place, new

subgroups will be formed and new clusterheads will be elected for each newly formed cluster.

Figure 4.6(a) illustrates the simulation results of the *mean lifetime of CHs* by varying group partition intervals from 30 seconds to 200 seconds. When group partition intervals are increased, some subgroups will become idle groups which do not have any destination assigned to the groups as described in Chapter 2, and small subgroups would merge into larger groups. As a result, more mobile nodes will be contained in each group and it becomes possible for a group to be composed by multiple clusters where multiple clusterheads may coexist in the group. When this happens, more CH contentions will take place in a group which leads to the decrease of the *mean lifetime of CHs* obtained by MOBIC and Lowest-ID clustering algorithms. However, the *mean lifetime of CHs* obtained by DGMA increases from 28s, 23s and 19s to 32s, 25s and 21s for DGMA-0.5, DGMA-0.4 and DGMA-0.3 respectively when the group partition interval is increased from 30 seconds to 100 seconds. It benefits from the mobility metric used in DGMA, i.e. *LDSD*, that there would not be any significant changes in *LDSD* of mobile nodes in an idle group. It is notable that the *mean lifetime of CHs* obtained by Lowest-ID decreases faster than DGMA and MOBIC when the group partition intervals are greater than 100 seconds. This is due to the fact that MOBIC and DGMA, which are based on mobility metrics, will not be affected greatly when mobile groups move slowly, e.g. in idle groups. Thus, DGMA and MOBIC can maintain a relatively stable *mean lifetime of CHs* when the group partition intervals are greater than 50 seconds. On the other hand, clusters formed by Lowest-ID would have more cluster re-election events taking place in a large group which is formed by multiple smaller subgroups.

Therefore, the *mean lifetime of CHs* obtained by Lowest-ID is decreased dramatically.

Figure 4.6(b) illustrates the simulation results of the *mean resident time* of cluster members by varying group partition intervals from 30 seconds to 200 seconds. With the increase in the *mean lifetime of CHs* obtained by DGMA as displayed in Figure 4.6(a), the *mean resident time* of cluster members obtained by DGMA also increases from 23s, 20s and 17s to 24s, 21s and 18s for DGMA-0.5, DGMA-0.4 and DGMA-0.3 respectively when the group partition intervals are increased from 30 seconds to 50 seconds. When the group partition interval is longer than 50 seconds, more smaller subgroups will merge into a larger group and it is possible for a group to contain multiple clusterheads simultaneously as discussed in Figure 4.6(a). As a result, cluster members may change their cluster affiliations to other clusters within the group and therefore the *mean resident time* of cluster members obtained by DGMA slightly drops for about 4 seconds on average for each of the DGMA schemes as displayed in Figure 4.6(b). As what we have discussed in subsection 4.5.3.2, the MOBIC clustering algorithm has a problem in ensuring a fairly good performance of the *mean resident time* of cluster members under the group mobility pattern if it is compared with DGMA. MOBIC yields a shorter *mean resident time* for 2 to 3 seconds than that of DGMA-0.3 when group partition intervals are increased. Lowest-ID yields the shortest *mean resident time* compared to others due to the short *mean lifetime of CHs* obtained by Lowest-ID.

Figure 4.6(c) presents the *average number of CH changes* per second by varying the group partition intervals. The *average number of CH changes* generated by DGMA are decreased from 0.39, 0.32 and 0.25 to 0.34, 0.24, and 0.19 for

DGMA-0.3, DGMA-0.4 and DGMA-0.5 respectively; and the *average number of CH changes* generated by MOBIC decreases from 0.33 to 0.3 when the group partition interval increases from 30 seconds to 50 seconds. When the group partition intervals are greater than 50 seconds, the *average number of CH changes* generated by MOBIC and DGMA are not changed significantly. On the other hand, the *average number of CH changes* generated by Lowest-ID increases dramatically as the group partition intervals are increased since more clusterhead contentions would take place in mobile groups.

Figure 4.6(d) presents the *number of cluster reaffiliations* per second for each clustering algorithm by varying the group partition intervals. It can be figured out that the *number of cluster reaffiliations* is increased when the group partition intervals are extended for each of these clustering algorithms. In particular, DGMA-0.4 and DGMA-0.5 yield less cluster reaffiliation events than other schemes, which increase from 2.7 to 4.3, and 2.4 to 3.6 respectively. Lowest-ID generates less cluster reaffiliation events than DGMA-0.3 when the group partition interval is smaller than 100 seconds, and after which they generate similar amount of the cluster reaffiliation events. The *number of cluster reaffiliations* generated by MOBIC is larger than others because cluster members may change their cluster affiliations frequently in a group with multiple clusterheads.

From these simulation results, Lowest-ID performs worst and is affected greatly by the group partitioning intervals. The performance of MOBIC is fairly good, which keeps a stable *mean lifetime of CHs* but also requests more cluster reaffiliation events. DGMA shows the best performance with predictably steady results which are not affected much by the group partitions.

4.5.3.4 Varying Node Densities

We further investigate the impact on the performance metrics by varying node densities in a group from $0.3 \text{ node}/(100\text{m})^2$ to $5 \text{ nodes}/(100\text{m})^2$. Node density can reflect the relative distance between cluster members. When the node density of a group is reduced, group members are distributed more sparsely in the group; and with a fixed number of group members, the radius of the group will be enlarged. We next investigate how distance between each pair of nodes would make an impact on the cluster's stability.

(a). Mean lifetime of CHs vs. Node density (nodes/(100m)²)

(b). Mean resident time vs. Node density (nodes/(100m)²)

(c). Average number of CH changes per sec. vs. Node density (nodes/(100m)²)

(d). Average number of cluster reaffiliations per sec. vs. Node density (nodes/(100m)²)

Figure 4.7: Performance metrics vs. node densities

As shown in Figure 4.7(a), the *mean lifetime of CHs* for both DGMA and Lowest-ID increase along with the increase of node density. DGMA levels up gradually with a small increment each time in contrast to Lowest-ID's rapid growth. The main factor that affects the results is the radius of a group which would be changed when node density varies. For a cluster with a lower node density, the relative distance between cluster members would be longer than that of a cluster with a higher node density. In such a case, it is possible for a group with a lower node density to be formed by multiple clusters if one clusterhead is not able to cover the whole group. Hence, clusterhead contentions may take place in the group. Increasing node densities can make the radius of the group become smaller, which reduces the chances for clusterhead contentions to take place in a group, and thus it can prolong the lifetime of clusterheads. For example, if the transmission range of a clusterhead elected by the Lowest-ID clustering algorithm can cover the whole group where only one cluster is formed, this clusterhead could keep its role for a very long period of time before encountering another node with a smaller node ID from any other groups. However, it is different in DGMA and MOBIC, where even when the membership of a cluster remains unchanged, clusterhead reelections would still take place due to the changes of their mobility metrics. Thus both DGMA and MOBIC are not sensitive to the density changes. It is notable when node densities are smaller than $2 \text{ nodes}/(100\text{m})^2$, the *mean lifetime of CHs* obtained by MOBIC decreases from 30 seconds to 25 seconds. As discussed earlier, it benefits from its "clusterhead contention interval" and its maintenance mechanism when two clusters are merged, re-elections of clusterheads would not be triggered. As a result, some clusterhead contentions taking place in DGMA and Lowest-ID would not happen in MOBIC. Thus MOBIC can achieve fairly good results in the

mean lifetime of CHs which is comparable to that of DGMA-0.5 when node density is below $0.5 \text{ node}/(100\text{m})^2$.

As shown in Figure 4.7(b), every clustering scheme yields an increasing *mean resident time* of cluster members when node density is raised. The *mean resident time* of cluster members in clusters formed by DGMA is longer than that of MOBIC and Lowest-ID, which is raised from 20s, 18s and 14s to 30s, 24s and 20s for DGMA-0.5, DGMA-0.4 and DGMA-0.3 respectively. MOBIC yields a little higher *mean resident time* than Lowest-ID for 1 second on average. As discussed earlier, an increase of the node density in a group shrinks the radius of the group. Hence, the chances for multiple clusterheads to coexist in a group is reduced. Therefore, the *mean resident time* of cluster members are increased for each of these clustering algorithms.

Discussions and results of Figure 4.7(a) and Figure 4.7(b) can also be supported by the *average number of CH changes* as illustrated in Figure 4.7(c). When the node density is low, group's radius is relatively large which produces a lot of overlapping areas between groups. It increases the possibility of co-existence of multiple clusterheads in a group. As a result, more changes of clusterheads would take place. DGMA-0.4 and DGMA-0.5 achieve less *average number of CH changes* than MOBIC. Lowest-ID displays a fast reducing rate and finally surpasses DGMA-0.5 in the number of cluster reaffiliations when density is increased beyond $3 \text{ node}/(100\text{m})^2$. This is due to the fact that the clusterhead formed by Lowest-ID is rather stable in a group with high node density, where the node with the lowest ID could keep its role for a longer time than mobility-metric based clustering algorithms. Analogous performance can be found on *average number of cluster*

reaffiliations as displayed in Figure 4.7(d). DGMA-0.4 and DGMA-0.5 generate less number of cluster reaffiliation events than MOBIC and Lowest-ID gains the fastest drop rate among these clustering algorithms.

From the simulation results, DGMA is superior to MOBIC and Lowest-ID as the node density increases in most circumstances. Lowest-ID performs well for dense network and its performance is close to that of DGMA-0.5 for such networks. However, it is not suitable for sparsely distributed mobile nodes. MOBIC behaves similarly with DGMA in tendencies of most results but it does not yield as good results as DGMA.

4.6 Performance Evaluation of DGMA Using Entity Mobility Model

Although DGMA is originally designed to support group mobility, in this section, we present our investigation on the performance of DGMA implemented under an entity mobility environment. In particular, the random waypoint mobility (RWP) pattern, which is an entity mobility model (as discussed in subsection 2.8.3), is used to evaluate the performance of DGMA. We compare the performance of DGMA-0.5 with the other two schemes, namely MOBIC and the Lowest-ID, by varying the node transmission ranges and node speeds separately. Nodes are uniformly distributed initially. Similar to the previous experiment, we set the default average speed to 10m/s (in the range of (5m/s, 15m/s)), the transmission range to 200m, and the pause time of node's mobility to zero.

4.6.1 Varying Transmission Ranges

(a). Mean lifetime of CHs vs Transmission range

(b). Mean resident time vs Transmission range

(c). Average number of CH changes per sec. vs Transmission range

(d). Average number of cluster reaffiliations per sec. vs Transmission range

Figure 4.8: Performance metrics vs. Transmission range with RWP mobility

It is not difficult to understand that when the transmission range is extended for mobile nodes with random mobility in a network, the radio coverage of a clusterhead can reach to more mobile nodes in the network. Thus, with their continuous mobility of the clusterheads in the network, the possibility for clusterhead contentions to take place is increased. Therefore, the *mean lifetime of CHs* obtained by each clustering algorithms is shorten as illustrated by Figure 4.8(a). MOBIC achieves a better performance in the *mean lifetime of CHs* than DGMA when the transmission range is less than 200 meters, but after which the

performance of MOBIC falls fast due to the frequent changes in the its mobility metric of clusterheads when the size of a cluster expands (the mobility metrics of each cluster member could be quite different from each other for cluster members in diverse mobility behaviors). Lowest-ID yields the shortest *mean lifetime of CHs* since it does not have any mechanism to defer the occurrence of clusterhead contentions. As illustrated in Figure 4.8(b), DGMA yields the longest *mean resident time* in these three clustering algorithms, which is increased from 10 seconds to 19 seconds when the transmission range is raised from 100 meters to 300 meters. MOBIC achieves a longer *mean resident time* than Lowest-ID when the transmission range is below 150 meters.

From Figure 4.8(c), it can be seen that DGMA and MOBIC yield less clusterhead changes on average than that of Lowest-ID when the transmission range is 100m. When the transmission range is small, many mobile nodes in the network would not have any neighbor and the value of the mobility metric for their nodes is 0. They would not form any cluster in such a case if DGMA or MOBIC is used. As a result, not too many clusterhead changes would take place. However in Lowest-ID, nodes without any neighbor will declare itself to be a clusterhead as a result it has more CH changes on average than DGMA and MOBIC. As the transmission range is raised to 150m, more small clusters will be formed in DGMA and MOBIC and therefore the number of CH changes increase from 0.35 to 0.5 and 0.4 to 0.6 for DGMA and MOBIC respectively. When the transmission range is greater than 150 meters, the size of clusters are increasing and the number of clusterheads in the network will decrease. Hence, the *average number of CH changes* starts to decrease in MOBIC and Lowest-ID. The analogue trend also appears to DGMA when the transmission range is greater than 200m. As nodes move in random mobility pattern,

the cluster's membership is very dynamic. With the increase of the transmission range, mobile nodes may be affiliated with multiple clusters simultaneously. Hence, the *number of cluster reaffiliations* grows when the transmission range is raised as illustrated in Figure 4.8(d).

4.6.2 Varying Speeds

(a). Mean lifetime of CHs vs Speeds

(b). Mean resident time vs Speeds

(c). Average number of CH changes per sec. vs Speeds

(d). Average number of cluster reaffiliations per sec. vs Speeds

Figure 4.9: Performance metrics vs. Speed with RWP mobility

Figure 4.9(a) to Figure 4.9(d) show the performance metrics by varying the average mobility speed from 5m/s to 20m/s. With a higher speed, the possibility for a clusterhead to encounter another clusterhead is increased in Lowest-ID, and the *mean lifetime of CHs* is hence shortened as illustrated in Figure 4.9(a). However,

both DGMA and MOBIC have the mechanism to defer clusterhead contentions. Hence their *mean lifetime of CHs* drops slower than Lowest-ID. As mobile nodes move in totally random behaviors, the time they stay in a cluster is shortened when their speeds are raised. Therefore, the *mean resident time* of cluster members in each of the schemes decreases, where DGMA yields the longest *mean resident time*.

Correspondingly, more CH changes are taking place in Lowest-ID when the mobility speed increases from 5m/s to 20m/s as illustrated in Figure 4.9(c). As discussed in Figure 4.9(a), DGMA and MOBIC benefits from the contention intervals for clusterheads, hence they yield less clusterhead changes. As illustrated in Figure 4.9(d), all these three clustering algorithms appear in a linear increasing trend in the *number of cluster reaffiliations* where DGMA has the least number of reaffiliations; however, the difference is not significant as the cluster members are equally dynamic no matter which clustering scheme is selected. The node distribution within a cluster and the position of the clusterheads may cause cluster reaffiliations of each scheme to deviate from each other slightly. Thus, it can be concluded that DGMA outperforms both MOBIC and Lowest-ID and MOBIC is better than Lowest-ID as speed increases in the entity mobility pattern.

From the above simulation results and discussion, we have shown that DGMA can also be applied to the entity mobility environment with a relatively good performance.

4.7 Conclusion

With the development of MANETs, large-scale MANETs will become one of the future research directions. The routing efficiency and network complexity would be greatly influenced by the number of mobile hosts. Clustering can provide large-scale MANETS with a hierarchical network structure to facilitate network operations. Thus, clustering approach will be an active research arena. In this chapter, we proposed a distributed clustering scheme for group mobility in MANETs, namely DGMA, as well as a novel group mobility metric, namely *linear distance based spatial dependency*. Based on the new mobility metric, it ensures the maximum cluster stability. As we have expected, the clusters formed by this scheme is more stable in terms of longer cluster lifetime, longer mean residence time for cluster members and less cluster reaffiliation events. The results reflect that the DGMA clustering algorithm outperforms the Lowest-ID and MOBIC clustering algorithms.

Group partition is an inherited group mobility behavior. With the DGMA clustering scheme, efficient group partition prediction algorithms can be developed based on the mobility information of the clusterheads and this topic will be discussed in the next chapter.

CHAPTER 5: A FRAMEWORK FOR PREDICTING GROUP PARTITIONS IN MANETS

5.1 Introduction

As it is well-known, dynamic network structure is a key feature of mobile ad hoc networks (MANETs). Network topology of MANETs changes rapidly and arbitrarily. It causes the prevalence of group partitions in many MANET applications such as military action, disaster recovery and conferences, where groups are dynamically formed and partitioned into smaller units. Group disconnection interrupts inter-group communication such that nodes in different groups may not be able to communicate with one another. However, if a potential group partition could be forecasted, remedial actions can be performed in advance to minimize the impact of loss of communications. For instance, deploying relay nodes between partitioning groups or replicating services to other nodes are applicable. In this chapter, we propose a framework to predict group partitions in MANETs . This prediction framework is composed of nine modules as illustrated in Figure 5.1.

The first module, called the *Group Mobility Generator*, is used to generate group mobility scenarios. A group mobility model is selected to generate random group partitions. Mobile nodes are initially deployed into groups of different sizes. The speed, direction and membership of every mobile node may vary from time to time. Furthermore, the number of groups, group size and mobile node affiliation are also changing dynamically in the selected group mobility model.

Figure 5.1: Architecture of group partition prediction system

The second module is called the *Clustering Algorithm module*. A clustering algorithm can partition a network into multiple clusters according to different clustering criteria, such as mobility metrics, power consumption and location information, etc. Clustering information is very useful in routing and group membership management. From experimental results of previous research work, network performance can be greatly improved by adopting effective clustering algorithms [72, 81, 90, 118]. Clustering algorithms, which are used to organize a network into multiple clusters with a hierarchical structure, are also useful in group

mobility predictions. Clusterheads can gather mobility information from each individual cluster member. A clusterhead can update its cluster's mobility information periodically and exchange such information with other connecting clusterheads for predicting group partitions.

The third module is the *Group Mobility Information Gathering module*. It is comprised of three parts: the core and two appendix functions, namely the *Mobility Validation function* and the *Group Mobility History Cache*. This module is responsible for calculating both mobile node and group information, including coordinates of group physical center, group size, velocity and direction, etc. These mobility information will be validated by the appendix functions. The processed data will be either pushed into the Group Mobility History Cache, discarded or converted into a command to empty the history cache.

The historical mobility information, which is stored in Group Mobility History Cache, will be retrieved by the 4th module, namely *the Mobility Prediction Module*, where the prediction algorithm is implemented. Group partition causes the decomposition of a given group into several subgroups of smaller sizes. Consequently, inter-group communication may be interrupted. Partition prediction algorithm can be used to alert network disconnections. Based on the mobility information extracted by the clustering algorithm, the group mobility tendency can be estimated. Thereafter, the clusterhead can broadcast an alarm of the group partition to its cluster members.

Given a series of historical mobility information, the approximate mobility trend can be estimated using Quadratic Regression³. We note that quadratic regression models mobility trajectory using a parabola, and is a more suitable method to be used for modeling non-linear movement as compared to the method of linear regression of least square. While using cubic regression or any other higher p^{th} order polynomial regression models may increase the accuracy of the estimation, it also results in an increase in computational complexity. In addition, a higher order polynomial regression can be converted into the form of lower order polynomial regressions. Thus the quadratic regression model is adopted for mobility prediction. With the aid of exponential smoothing, the accuracy of the prediction results obtained using quadratic regression algorithm can be further improved.

The fifth module is the *Group Safe Distance Analysis module*; this module computes the distance between adjacent groups to guarantee their inter-group communication. Huang *et al.* [22] introduced the concept of safe distance between two mobile nodes in MANETs; this distance is essentially the fundamental distance to guarantee the communication between two mobile nodes. However, to the best of our knowledge, the notion of safe distance between two groups has never been addressed yet. We define the *safe distance* between two groups as the fundamental distance to guarantee the communication between two groups (further details will be given in the subsection 5.3.3). Communications between groups could be one hop or multi-hops. The safe distance provides a criterion to justify whether two groups are connected to each other. Beyond the safe distance, the possibility of communication interruption between two groups is high due to group partitions.

³ Quadratic regression is an equation of a parabola that “best fits” a given set of data by using the least square method.

Based on the predicted mobility speed and direction and the safe distance derived by module 4 and module 5 respectively, the partition time between two connecting clusters can be estimated by module 6, the *Group Partition Time Estimation module*; this module uses geometric equations to calculate the approximate partitioning time between a couple of splitting groups.

The Module 7, namely the *Actual Partition Time Monitor module*, is used to monitor the actual time of group partitions. This module maintains a connectivity table with group id, and timing of group connectivity. The network connectivity information is updated every second. The predicted partition time and the actual partition time between a couple of groups will be compared in Module 8, namely the *Compare Estimated Results with Actual Value module*. Finally, the statistic and experimental data will be received and categorized according to different criteria to evaluate the performance of the predicted results in Module 9.

The chapter is organized as follows. Section 5.2 gives a literature review on some of the existing partition prediction algorithms for MANETs. Section 5.3 presents our proposed group partition prediction algorithm including the analysis of group's *macro movement*, group safe distance and other modules based on the system flow depicted in Figure 5.1. Experimental results in terms of prediction accuracy and deviations are discussed in Section 5.4. Section 5.5 compares two different clustering schemes under the same network environment and discusses their impact on the performance of the prediction results. This chapter concludes with Section 5.6.

5.2 Review of Partition Prediction Algorithms

In this section, we will review some of the existing work on partition prediction algorithms. Su *et al.* [18] proposed using mobility prediction to assist network routing. Based on the instantaneous speed of mobile nodes, the distance between a pair of nodes after a period time can be estimated by solving geometric equations. This algorithm was designed for an environment which assumes that a host moves with constant speed and direction; this means that any changes in speed and direction may cause previous predictions to become invalid. Wang and Li's work [35] aims for service replacement if a partition is predicted to take place in the near future. A network partition scenario was described with their Reference Velocity Mobility Model. They simplify the group mobility network topology with a collection of equal sized "circles" that are initially stacked on top of each other with the assumption that the velocities of mobile groups' are time-invariant; we note that this is not considered as a typical characteristic of ad hoc networks. Some other existing work focuses on predicting the link availability information or on detecting broken links based on some ideal scenarios or snapshots of static network topologies[17, 20]. Clients and servers are connected by multiple disjoint paths so that link robustness can be observed for partition detection[20]. The sectorized ad hoc mobility prediction scheme[15] partitions a region into a number of sectors and calculates future sector indexes for each mobile node with its current location and mobility direction probabilistically. This is the traditional method used for handover in cellular networks.

Despite all these works, we observe that none of existing work provides a solution on the prediction of partitioning time. Hence, in this chapter we target on designing an innovative group partition prediction algorithm that can be used under

more complex and generic scenarios. Mobile node's affiliation and group membership are not fixed; instead they may be changing frequently in our proposed scheme. Group mobility is also without predefined routes and checkpoints while group partition and merger happen randomly.

5.3 Group Partition Prediction

We have generally introduced the prediction framework by describing each module of the framework in Section 5.1. In this section, we will specifically introduce and discuss each of modules of the prediction framework.

5.3.1. Group Mobility Generator

Mobility models and effective mobility management schemes are very important in reflecting network mobility patterns. Mobility models provide simulation platforms for research in MANETs. Mobility models are generally categorized as entity mobility models and group mobility models. Although many group mobility models have been proposed in the literature for MANETs [7, 26, 34, 119], group membership in these models are fixed as mobile nodes are not allowed to change their membership arbitrarily. Hence these mobility models are not able to describe our desired mobility patterns where group partitions will result in a change of group membership.

In our experiment, we adopted our proposed *Reference Region Group Mobility* (RRGM) model to generate group partition and group merger mobility pattern. As discussed in Chapter 2, RRGM model is a generic mobility model that can be used for describing both entity mobility and group mobility. When it is used as a group mobility model, the groups are eligible to partition and merge with other groups. In

RRGM, group membership is not fixed and it would change dynamically. Every group is initially assigned with a destination and the group is then associated with a reference region which is used to guide the group's movement towards the target region. Every node belonging to this group randomly picks a point inside the reference region and moves towards that point. After being stationary at an intermediate location for a period of time, the reference region will be relocated upon the arrival of all group members. As such, the reference region moves gradually towards the group destination step by step and nodes in the group finally reach the destination. Destinations are generated regularly for mobile groups in RRGM. If multiple destinations are assigned to a group, this group will be partitioned into a number of smaller subgroups, each with a new reference region associated with a different destination. After the members of the various partitioned subgroups reach their destinations and have completed their tasks, they may merge into larger groups by merging their respective reference regions together. The merger between groups is then completed.

5.3.2. Clustering Algorithm

A clustering algorithm is used to organize a network into a number of clusters. To predict group partitions, regional mobility information needs to be exchanged between different clusters. Our proposed *Distributed Group Mobility Adaptive Clustering algorithm (DGMA)* as described in Chapter 4 is adopted as the clustering scheme in this framework for its ability in coping with group mobility pattern and utilizing mobility attributes as the clustering criterion whereas other similar clustering algorithms may not have such features.

In DGMA, spatial dependency represents the similarity in mobility of a mobile node with other surrounding neighbors with respect to speed and direction. If a mobile node moves in different direction as compared to another node, their relative spatial dependency would be small; this means that their mobility is independent from each other. On the contrary, if a node's movement is closely related to another node, such that they are moving in a similar direction and speed, they are expected to have a higher value in spatial dependency. In DGMA, it makes sense that a node having the largest total spatial dependency to other neighboring nodes is elected to be the clusterhead. By using DGMA, when the coverage area of a group expands due to node mobility, a mobile group is sectorized into a number of small clusters with each of them having their respective clusterhead. It is possible for a big group to be composed of one or multiple clusters according to its group size. The clusterhead can monitor and calculate the average speed and direction of all its cluster members, which is in turn used to represent the whole cluster's mobility information.

5.3.3. Gathering Group Mobility Information

The *Gathering Group Mobility Information module* is used to retrieve group mobility information based on clustered network structure. After the network is logically partitioned into different clusters by DGMA, each cluster's mobility information is processed by the respective clusterhead in this module. The protocol is illustrated in Figure 5.2. Clusterheads elected by DGMA are designated to monitor and calculate the change of location in coordinates per second.

Figure 5.2: Procedure of gathering group mobility information

We introduce a novel criterion called ‘*macro movement*’ to update historical mobility information. Let D_{thr} denotes the threshold to justify *macro movement*. Let A denotes the most recent entry to the mobility historical cache (A is referred to as a *checkpoint*). Let D denotes the linear distance between the coordinates of checkpoint A and the current group coordinates.

If the displacement D of a group from the last checkpoint to its current position is greater than D_{thr} , it indicates that the group has made a *macro movement* from the last checkpoint. Once a *macro movement* is made, a checkpoint will be inserted into its mobility history cache to identify the location and time.

Figure 5.3 gives an example to illustrate the idea of *macro movement*. Suppose a group moves along the solid curve from checkpoint A to checkpoint K . The dotted line denotes the trajectory of *macro movements*, such as $A-B$, $B-C$, and $C-D$, etc. Thus, complex non-linear movement trajectory is simplified into pieces of linear segments of equidistance (which is D_{thr}). At intermediate points between checkpoints, clusterheads calculate the displacement to the last checkpoint of their respective group periodically and compare it with D_{thr} . If confirmed as a *macro movement* after comparison, group historical mobility information with respect of speed and direction of the last *macro movement* segment will be calculated.

Observe that the distance between checkpoint A and point A' is within the range of $[0, D_{thr})$, hence the path A-A' is not a *macro movement*.

Figure 5.3: Example of group *macro movement*

We assume that every clusterhead has the complete knowledge of its cluster, i.e. the coordinates of group's physical center, speed and direction of movement, etc. The group's physical center is used to represent the current location of the group. We define a group member set as $Mem=\{1,2,\dots,i,\dots,N\}$ including the clusterhead i itself. The coordinates of group physical center $(\bar{x}_i, \bar{y}_i)$, which are the mean of all group members' coordinates, are calculated by:

$$\bar{x}_i = \frac{\sum_{n=1}^N x_n}{N}, \text{ and } \bar{y}_i = \frac{\sum_{n=1}^N y_n}{N} \quad (5.1)$$

Then, the distance D between the current location $(\bar{x}_i, \bar{y}_i)$ and the last historical group location $(\hat{x}_i(t), \hat{y}_i(t))$ at time t is calculated as follows:

$$D = \sqrt{(\bar{x}_i - \hat{x}_i(t))^2 + (\bar{y}_i - \hat{y}_i(t))^2} \quad (5.2)$$

If $D \geq D_{thr}$, the group historical speed and direction from the last historical time t to the present time t' , $\hat{S}_i(t')$ and $\hat{\theta}_i(t')$, can be calculated using $(\hat{x}_i(t), \hat{y}_i(t), t)$ and t' as follows:

$$\hat{S}_i(t') = \frac{D}{t' - t} \quad (5.3)$$

$$\hat{\theta}_i(t') = \begin{cases} \varphi \cdot \text{sgn}(\bar{y}_i - \hat{y}_i(t)) & \bar{x}_i - \hat{x}_i(t) > 0 \\ \pi / 2 \cdot \text{sgn}(\bar{y}_i - \hat{y}_i(t)) & \bar{x}_i - \hat{x}_i(t) = 0 \\ (\pi - \varphi) \cdot \text{sgn}(\bar{y}_i - \hat{y}_i(t)) & \bar{x}_i - \hat{x}_i(t) < 0, \text{ and } \bar{y}_i - \hat{y}_i(t) \neq 0 \\ \pi & \bar{x}_i - \hat{x}_i(t) < 0, \text{ and } \bar{y}_i - \hat{y}_i(t) = 0 \end{cases} \quad (5.4)$$

where

$$\tan \varphi = \frac{|\bar{y}_i - \hat{y}_i(t)|}{|\bar{x}_i - \hat{x}_i(t)|}, \quad (0 \leq \varphi < \frac{\pi}{2}) \quad (5.5)$$

and $\hat{\theta}_i(t') \in (-\pi, \pi]$. The values of t' , $\hat{S}_i(t')$, $\hat{\theta}_i(t')$ and $(\bar{x}_i, \bar{y}_i)$ will be stored into the history cache for group i as $t', \hat{S}_i(t'), \hat{\theta}_i(t'), (\hat{x}_i(t'), \hat{y}_i(t'))$, in the form of a queue⁴. As the group speed and direction at each historical checkpoint are tracked, the change in direction of group i from time t to time t' is calculated by:

$$\Delta \hat{\theta}_i(t') = \hat{\theta}_i(t') - \hat{\theta}_i(t) \quad (5.6)$$

This difference is kept by the group's clusterhead and updated at each *macro movement*. It is necessary to notice that sudden turn may happen occasionally. In such a case, previous historical mobility information will be rendered useless to forecast future mobility trend. Therefore, invalid mobility historical information

⁴ $(\hat{x}_i(t'), \hat{y}_i(t'))$ is the format of historical data to store the value of $(\bar{x}_i, \bar{y}_i)$ in the queue.

should be discarded. We introduce another parameter θ_{thr} to discard invalid mobility information (which takes place when a sharp turn occurs). If $\Delta\hat{\theta}_i(t')$ is greater than θ_{thr} , historical mobility information will be rendered invalid and previous mobility information is cleared from the history cache, thus making it vacant. The value of θ_{thr} should be selected properly. If this threshold is configured too small, group mobility history cache would be vacated frequently which results in insufficient historical information for prediction and thus the predicted results would be inaccurate.

5.3.4. Mobility Prediction --- Quadratic Regression Trend Model

We assume we have the last $k-1$ historical mobility records. $S_i(t_k)$ and $\theta_i(t_k)$ represent the actual speed and direction of group i at time t_k , while $\hat{S}_i(t_k)$ and $\hat{\theta}_i(t_k)$ represent the predicted speed and direction at time t_k . The relation between actual and predicted value is:

$$S_i(t_k) = \hat{S}_i(t_k) + \varepsilon_i(t_k) \quad (5.7)$$

where $\varepsilon_i(t)$ is the error term between the predicted value and the actual value at time t_k , that can be observed at the next time instant t_{k+1} . This error term is a random fluctuation that causes the $S_i(t_k)$ value to deviate from $\hat{S}_i(t_k)$. The predicted speed at time instance t_k can be expressed using the quadratic regression model as follows:

$$\hat{S}_i(t_k) = \beta_0 + \beta_1 t_k + \beta_2 t_k^2 \quad (5.8)$$

β_0 , β_1 , and β_2 in the equation (5.8) are parameters used to define the position and slope of a parabola that have a best fit of all known historical data and to minimize the sum of squared residuals, Y , which is defined in the form of:

$$Y = \sum_{j=1}^k [S_i(t_j) - \hat{S}_i(t_j)]^2 \quad (5.9)$$

Combining (5.8) and (5.9), we have:

$$Y = \sum_{j=1}^k [S_i(t_j) - (\beta_0 + \beta_1 t_j + \beta_2 t_j^2)]^2 \quad (5.10)$$

The values of β_0 , β_1 , and β_2 can be derived by solving partial differential equations. The minimum value of Y occurs when the gradient is zero. Since there are 3 parameters, we will need 3 gradient equations[120]:

$$\frac{\partial Y}{\partial \beta_0} = 0, \frac{\partial Y}{\partial \beta_1} = 0, \frac{\partial Y}{\partial \beta_2} = 0 \quad (5.10.1)$$

From (5.10.1), we can deduce:

$$\begin{bmatrix} k & \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 \\ \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 \\ \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 & \sum_{j=1}^k t_j^4 \end{bmatrix} * \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^k S_i(t_j) \\ \sum_{j=1}^k t_j S_i(t_j) \\ \sum_{j=1}^k t_j^2 S_i(t_j) \end{bmatrix} \quad (5.10.2)$$

(5.10.3) can be replaced by:

$$\begin{aligned} T\beta &= S \\ \beta &= (TT^T)^{-1}T^T S \end{aligned} \quad (5.10.3)$$

Since all t_j and $S_i(t_j)$ are known, β_0 , β_1 , and β_2 can be found by solving the matrix (5.10.3)⁵.

The estimate of $\hat{S}_i(t_k)$ made in time period t_k could be refined using exponential smoothing [121]. We note that recent historical records may be more relevant to the current prediction; hence using the notion of the weighted moving average [121], we derive the following smoothing equation with the smoothing factor α_0 in the range of $[0, 1]$:

$$\hat{S}_i(t_k)' = \alpha_0 \hat{S}_i(t_k) + (1 - \alpha_0) \sum_{j=1}^{k-1} \mu_j S_i(t_{k-j}) \quad (5.11)$$

The first half of the RHS of equation (5.11) represents the current predicted value while the second half represents the historical data. When the value of α_0 is closer to 1, the equation has a less of smoothing effect and the current prediction $\hat{S}_i(t_k)$ gives more weight to the result $\hat{S}_i(t_k)'$. On the contrary, if the value of α_0 is closer to 0, $\hat{S}_i(t_k)'$ is less responsive to the current prediction. Considering the fact that the current predicted value should have a greater influence on the smoothed result, we define α_0 to be 0.7. Weighted moving average assigns different weights to historical observations. μ_j is a weighting factor for each of the historical terms. In practice, the weighting factors are often chosen to give more weight to the more recent data and less weight to older data. Hence, the value of μ_j decreases as j increases, i.e. $\mu_j > \mu_{j+1}$, since remote historical data may become enervated to the current estimation. Thus, we define μ_j as:

⁵ Refer more details in the Appendix-A.

$$\mu_j = \frac{k-j}{1+2+\dots+k-1} = \frac{k-j}{\frac{k(k-1)}{2}} = \frac{2(k-j)}{k(k-1)} \quad (5.12)$$

$$\begin{aligned} \sum_{j=1}^{k-1} \mu_j &= \sum_{j=1}^{k-1} \frac{2(k-j)}{k(k-1)} \\ &= 2 \sum_{j=1}^{k-1} \left[\frac{1}{k-1} - \frac{j}{k(k-1)} \right] \\ &= 2(k-1) \cdot \frac{1}{k-1} - \frac{2}{k(k-1)} \sum_{j=1}^{k-1} j \quad (5.13) \\ &= 2 - \frac{2}{k(k-1)} \cdot \frac{k(k-1)}{2} \\ &= 1 \end{aligned}$$

Thus $\hat{S}_i(t_k)'$ can be rewritten into:

$$\hat{S}_i(t_k)' = \alpha_0 \hat{S}_i(t_k) + (1 - \alpha_0) \sum_{j=1}^{k-1} \frac{2(k-j)}{k(k-1)} S_i(t_{k-j}) \quad (5.14)$$

Moreover, a $100(1-\alpha) \%$ prediction interval for $S_i(t_k)$ is $[-\varepsilon_i(t_k), \varepsilon_i(t_k)]$, which is represented by:

$$\varepsilon_i(t) = t_{[\alpha/2]}^{(k-1)} s \sqrt{1 + (1/k)} \quad (5.15)$$

where $s = \sqrt{(\sum_{t=1}^k [S_i(t_k) - \bar{S}_i]^2) / (k-1)}$ and $\bar{S}_i$ is the mean of historical actual speeds.

Thus, equation (5.7) can be expressed as:

$$S_i(t_k) \in \alpha_0 \hat{S}_i(t_k) + (1 - \alpha_0) \sum_{j=1}^k \frac{2(k-j+1)}{k(k+1)} S_i(t_{k-j}) \pm t_{[\alpha/2]}^{(k-1)} s \sqrt{1 + \left(\frac{1}{k}\right)} \quad (5.16)$$

The way in which $\theta_i(t_k)$ is derived is similar to what we have done for $s_i(t_k)$;

thus we will skip the details and $\theta_i(t_k)$ can be expressed as follows :

$$\theta_i(t_k) \in \alpha_0 \bar{\theta}_i(t_k) + (1 - \alpha_0) \sum_{j=1}^k \frac{2(k-j+1)}{k(k+1)} \theta_i(t_{k-j}) \pm t_{[\alpha/2]}^{(k-1)} s' \sqrt{1 + \left(\frac{1}{k}\right)} \quad (5.17)$$

where $s' = \sqrt{\frac{\sum_{i=1}^k [\theta_i(t_k) - \bar{\theta}_i]^2}{k-1}}$ and $\bar{\theta}_i$ is the mean of historical actual angles.

5.3.5. Group Safe Distance Analysis

The ‘Group Safe Distance Analysis’ module calculates the distance between two groups to guarantee their inter-group communication. It relates to the number of members in a group (i.e. group size) and their density in deployment. It provides a criterion to justify within what distance two groups are connected to each other. Beyond the safe distance, the possibility of communication interruption between two groups is high due to group partitions.

When two connected groups are going to spilt, the overlapping area between these two groups is decreasing. When the intersection area vanishes, the two groups are physically partitioned. Centered by the average coordinates of group members’, its coverage could be assumed to be a circular area with radius r , which depends on the number of group members n , (i.e. group size), and the distributed density, ρ , of these nodes, which is the average number of nodes n per area unit s :

$$\rho = \frac{n}{s} = \frac{n}{\pi r^2} \quad (5.18)$$

Then r can be denoted as:

$$r = \sqrt{\frac{n}{\rho \pi}} \quad (5.18.1)$$

where n is a positive integer. Given that all mobile nodes are equipped with compatible wireless transmitters of equal radio range Tr , and we assume group members are uniformly distributed in the circular area of the group, the group safe distance is defined as the maximum distance at which communication can be guaranteed to take place between these pair of groups. The group safe distance d between two groups is represented by:

$$d = Tr + r_1 + r_2 \quad (5.19)$$

where Tr is the unique transmission range, and r_1 and r_2 are the group radius of each group respectively. It can be illustrated by Figure 5.4 that it displays the extreme case for inter-group communication where the distance of the two groups equals to $Tr + r_1 + r_2$ (the distance of two groups refers to the distance between the physical centers of the two groups in this article). The inter-group communication can be ensured if the distance is less than or equal to d .

Figure 5.4: Example of the safe distance between two groups

Since $r = (n\rho^{-1}\pi^{-1})^{1/2}$, equation (5.19) can be rewritten as:

$$\rho\pi(d - Tr)^2 = n_1 + n_2 + 2\sqrt{n_1n_2} \quad (5.19.1)$$

We assume that n_1 and n_2 are both approximately equal to the average group size; therefore $(n_1 n_2)^{1/2}$ can be approximated by $(n_1 + n_2)/2$. Equation (5.19) can be further simplified as:

$$d = Tr + \sqrt{\frac{2(n_1 + n_2)}{\rho\pi}} \quad (5.19.2)$$

For example, consider the case where Tr is 200, $\rho=0.0001$, and n_1 and n_2 are 5 and 8 respectively. Using equation (5.19) and (5.19.2), the safe distance d will be 485.5 and 487.7 respectively. We observe that the estimation error is negligible in general and could thus be ignored.

During group partitions, an overlapping area between two partitioning groups is commonly seen before they completely split apart. During this transitional period of group partition, some nodes may be located in the intersection area. These nodes are counted as part of n_1 and n_2 in the equation (5.19.2). Let n_{12} , denotes the number of nodes in the intersection area. We next modify equation (5.19.2) to address the overcounting of nodes:

$$d = Tr + \sqrt{\frac{2(n_1 + n_2 - n_{12})}{\rho\pi}} \quad (5.19.3)$$

We use the term $d(t)$ to represent the distance between partitioning groups at time t . On the assumption that nodes are evenly distributed within their group area and they are not moving very fast, group communication can be guaranteed if $d(t) \leq d$.

5.3.6. Group Partition Time Estimation

The time that two groups may partition depends on their locations, distance, group size, and their respective speed and moving direction. Each group has a clusterhead which commits to exchange the information of group physical centers' coordinates and other mobility attributes, such as predicted speed and direction, to other connecting clusterheads periodically. Assume this information is represented as $\{(\bar{x}, \bar{y}), \hat{S}(t), \hat{\theta}(t)\}$. After retrieving neighboring groups mobility parameters, the estimated connecting duration and partition time could be calculated.

Figure 5.5: Illustration of partition time estimation

Suppose two groups i and j will partition after time τ , at time $T(t+\tau)$ as illustrated in Figure 5.5. Assume the group center's coordinates of group i , j at time t are $(\bar{x}_i(t), \bar{y}_i(t))$, and $(\bar{x}_j(t), \bar{y}_j(t))$ respectively. We can compute their coordinates at time $T(t+\tau)$:

$$\begin{aligned}
 \bar{x}_i(t + \tau) &= \bar{x}_i(t) + \hat{S}_i(t) \cdot \cos \hat{\theta}_i(t) \cdot \tau \\
 \bar{y}_i(t + \tau) &= \bar{y}_i(t) + \hat{S}_i(t) \cdot \sin \hat{\theta}_i(t) \cdot \tau \\
 \bar{x}_j(t + \tau) &= \bar{x}_j(t) + \hat{S}_j(t) \cdot \cos \hat{\theta}_j(t) \cdot \tau \\
 \bar{y}_j(t + \tau) &= \bar{y}_j(t) + \hat{S}_j(t) \cdot \sin \hat{\theta}_j(t) \cdot \tau
 \end{aligned} \tag{5.20}$$

The time τ can be derived as follows: First we note that

$$[\bar{x}_i(t + \tau) - \bar{x}_j(t + \tau)]^2 + [\bar{y}_i(t + \tau) - \bar{y}_j(t + \tau)]^2 = d^2 \quad (5.20.1)$$

Combining (5.20) and (5.20.1), we have the following:

$$\begin{aligned} & [\bar{x}_i(t) + \hat{S}_i(t) \cdot \cos \hat{\theta}_i(t) \cdot \tau - (\bar{x}_j(t) + \hat{S}_j(t) \cdot \cos \hat{\theta}_j(t) \cdot \tau)]^2 + \\ & [\bar{y}_i(t) + \hat{S}_i(t) \cdot \sin \hat{\theta}_i(t) \cdot \tau - (\bar{y}_j(t) + \hat{S}_j(t) \cdot \sin \hat{\theta}_j(t) \cdot \tau)]^2 \\ & = d^2 \end{aligned} \quad (5.20.2)$$

Let

$$\begin{aligned} A &= \hat{S}_i(t) \cos \hat{\theta}_i(t) - \hat{S}_j(t) \cos \hat{\theta}_j(t) ; B = \bar{x}_i - \bar{x}_j \\ C &= \hat{S}_i(t) \sin \hat{\theta}_i(t) - \hat{S}_j(t) \sin \hat{\theta}_j(t) ; D = \bar{y}_i - \bar{y}_j \end{aligned} \quad (5.20.3)$$

Thus (5.20.2) can be further simplified into:

$$(A\tau + B)^2 + (C\tau + D)^2 = d^2 \quad (5.20.4)$$

Solving (5.20.4), which is a quadratic equation in τ , will give the estimated partitioning time τ as follows:

$$\tau = \frac{-(AB + CD) \pm \sqrt{(A^2 + C^2)d^2 - (AD - BC)^2}}{A^2 + C^2} \quad (5.20.5)$$

Let τ_1 and τ_2 be the solutions of equation (5.20.5). The case where τ_1 or τ_2 is less than zero will be ignored since a negative value of τ is meaningless for a partition time. So, next we consider the case where both τ_1 or τ_2 are non-negative, and $\tau_1 \geq \tau_2 > 0$. Because if two groups are moving in opposite directions (the included angle is within $[\pi/2, \pi]$), they may partition in shorter period of time than moving in

similar directions (the included angle is within $[0, \pi/2]$). Therefore, the estimated partition time is τ_1 if $|\hat{\theta}_i(t) - \hat{\theta}_j(t)| < \frac{\pi}{2}$, otherwise it would be τ_2 .

5.3.7. Prediction Result Evaluation

This process contains three related modules 7, 8 and 9 for monitoring actual partition time, comparison with the predicted results and experimental results. The Actual Partition Time Monitor, is a stand-alone module which is running in parallel with modules 3 to 6. The function of this module is to monitor the cluster's connectivity synchronously so that the predicted results could be compared to the actual partition time. It records every connecting cluster pairs by their cluster IDs and the time when their connection is set up as well as the time when they disconnect. It is updated periodically in every second. The actual partition time will be compared with the predicted time and finally reflected in the experimental results via statistical analysis by module 9.

5.4 Simulation

As introduced in the subsection 5.3.1, group partition mobility behavior is generated by RRGm model. Mobile nodes are deployed to network and they compose a number of groups randomly. Their speed, direction and membership vary from time to time during the simulation. Thus the number of groups, the group size and group constituent are dynamically changing. Figure 5.6 illustrates group partition and merger generated by this model. From the graph, an apparent group partition can be seen from $T=10s$ to $T=40s$ when a larger group has split into two smaller subgroups. At $T=60s$, each subgroup continuously partitions into two further smaller subgroups. At $T=80s$, subgroups begin to merge with one another. It

is worth to notice that the newly formed groups are composed of mobile nodes which originate from different groups. Thus group membership has changed. $T=90s$ shows a stable stage that nodes move within the respective range of their groups. ‘Sudden turn’ problem needs to be identified during prediction to discard invalid mobility history where a threshold $\pi/2$ is assigned to θ_{thr} . If any turn is greater than $\pi/2$, which is considered as a sudden turn, existing historical records will be cleared from the mobility cache. Group mobility information of speed and direction is updated by clusterhead in every 15 meters of group displacement. Group safe distance is given by the length of transmission range and group radius as discussed in equation (5.19.4) for securing inter-group connections.

Figure 5.6: Example of group partitions generated by RRG

Configuration of the parameters for mobility and prediction is given in Table 5.1.

It is worth to notice that the node speed is distributed in a range from 1m/s to 3m/s.

The unique group density varies from $0.5 \text{ node}/(100\text{m})^2$ to $1 \text{ node}/(100\text{m})^2$, which means in an area of $(100\text{m})^2$, there is 0.5 node or 1 node on average. E.g., for a group having 10 nodes, its coverage is a circular area in the radius of 252.38 meters with the density $0.5 \text{ node}/(100\text{m})^2$; the radius is 178.46m when the density is $1 \text{ node}/(100\text{m})^2$.

Table 5.1 Prediction Parameters

TITLE	SYMBOL	Value
<u><i>Mobility Parameters</i></u>		
Simulation Time	$Time$	5000s
Terrain Size	$X \times Y$	2000m×1800m
Number of Nodes	N	40
Transmission Range	Tr	100m, 150m, 200m
Speeds	$S_{min} \sim S_{max}$	1~3m/s
Group Density	P	0.5 , 1 node/ $(100\text{m})^2$
<u><i>Prediction Parameters</i></u>		
Sharp Turn Threshold	Θ_{thr}	$\pi/2$
Linear Distance	D_{thr}	15m
Safe Distance	D	$Tr + (\frac{2(n_1 + n_2 - n_{12})}{\rho\pi})^{1/2}$

Figure 5.7 shows the accuracy of predictions when the network density is $0.5 \text{ node}/(100\text{m})^2$ and $Tr=100\text{m}$. The x axis exhibits the time interval before partitions by every three-second time slot. For example, the point of zero is the time that a partition happens and the label ‘6-9s’ stands for 6 to 9 seconds before a partition happens. The y axis represents the accuracy of predictions in percentage.

Figure 5.7: Prediction Accuracy, $\rho=0.5$ node/(100m)², $Tr=100$ m

In order to estimate the prediction accuracy, we propose a new variable, namely a prediction time window $y=(\mu t)^{1/2}$ to provide a range to evaluate predictions. In particular, if the difference between the predicted value and the actual value falls within the prediction time window, the predicted value will be considered as a valid prediction. In addition, the size of the time window can be adjusted by changing the value of the parameter μ . For example, by assuming $\mu=1$, the range of the time window for evaluating predictions is $[0, t^{1/2}]$. If the difference between the predicted partition time and the actual partition time falls within the range of $[0, t^{1/2}]$, then this prediction will be reflected as valid prediction in the statistics. This can be illustrated with a numerical example: suppose the time before a partition takes place $t=16$ s, and the predicted time to partition is 13s. We note that the difference between the real partition time and predicted partition time is 3s and it falls in the range of the defined time window $[0, (16)^{1/2}=4]$ when $\mu=1$; hence under this condition, this prediction is counted as a valid prediction. When we change μ to 2 or

3, the time window will be 1.4 (the result of $2^{1/2}$) times and 1.7 (the result of $3^{1/2}$) times larger accordingly as compared to the time window when $\mu=1$.

From Figure 5.7, we observe that the prediction accuracy increases when the time interval before partition taking place decreases. This is due to the fact if the time interval before partition taking place is large, moving groups have a higher possibility to change their directions due to their random mobility pattern, which results in earlier predictions becoming outdated. When we enlarge the prediction time window to $(2t)^{0.5}$ and $(3t)^{0.5}$, the prediction accuracy improves significantly as displayed in the first and second curves in the Figure 5.7. For instance, the prediction accuracy is nearly 90% for the [6-9s] interval before partitions occur under the time window $(3t)^{0.5}$.

Figure 5.8: Prediction Accuracy, $\rho=1$ node/ $(100m)^2$, $Tr=100m$

Figure 5.9: Prediction Accuracy, $\rho=1$ node/ $(100m)^2$, $Tr=150m$

Figure 5.10: Prediction Accuracy, $\rho=1$ node/ $(100m)^2$, $Tr=200m$

Similarly, when the node density is fixed at $1 \text{ node}/(100\text{m})^2$ and changing the transmission range from 100m to 150m and 200m, the performance of the prediction algorithm varies as shown in Figure 5.8, 5.9, and 5.10. When transmission range increases, safe distance for connections also gets longer. It can hedge some of the inaccurate predictions caused by direction changes. It can be seen from the graphs, the prediction accuracy increases as the transmission range grows larger, especially in the early stage of predictions. This is due to the fact that more valid historical mobility information is available for making predictions if the transmission range is large. When the acceptable prediction range is increased to $1.4 (2^{1/2})$ and $1.7(3^{1/2})$ times of $t^{0.5}$, the performance improves for more than 20% and 40% respectively; we also note that more than 70% predictions fall in the acceptable prediction range for the $[0, 15\text{s}]$ interval when μ is 3.

Figure 5.11: Prediction deviation, $\rho=1 \text{ node}/(100\text{m})^2$

Figure 5.11 and Figure 5.12 illustrate the prediction deviation from the actual values. Figure 5.11 displays the deviation by varying different transmission ranges.

As the transmission range increases, deviation decreases in the sense that prediction becomes more precise. Because when the safe distance is increased, a group may need longer time to split. As a result, more valid historical mobility information could be captured and used in the prediction algorithm. Thus the deviation is reduced along with the increase of transmission ranges.

Figure 5.12: Prediction deviation, Tr=100m

In the Figure 5.12, when the group density is decreased by 50% from 1 node/(100m)² to 0.5 node/(100m)², the deviation reduces by approximately 20%. This is due to the fact that as the group density decreases, the safe distance increases. In conclusion, larger transmission range and smaller group density help in improving the accuracy of the predicted results.

From the above statistic, the quadratic regression trend model demonstrates its feasibility and efficiency in predicting group partitions.

5.5 Discussion on the Impact of Clustering Schemes on the Prediction Results

Our empirical results in Section 5.4 show that there are a number of factors that can affect the efficiency of the group partition prediction system, e.g. the transmission range, group density, and sudden change in mobility directions. In addition to these, we note that different clustering schemes may also influence the accuracy of the prediction scheme. Clustering scheme is a key module in the prediction system because clustering scheme determines how clusters are formed, based on which the cluster information is directly utilized for predictions. A suitable clustering scheme can improve the prediction outcomes and vice versa. Hence in this section, we investigate two different types of clustering schemes, called DGMA and Lowest-ID, to get a better understanding on how clustering scheme could affect the prediction output.

Figure 5.13 illustrates the difference in the prediction output for the Lowest-ID clustering scheme and the DGMA clustering scheme. It reflects the prediction accuracy in term of the absolute value of estimation error from the actual partition time. This is implemented in a smaller size network with 20 nodes in a 900m x 600m terrain. Other parameters are the same as in Table I. We investigate the prediction results obtained using the two clustering schemes for an interval of [0, 25s] before partition occurs to display the trend of changes.

Figure 5.13: Estimation error with respect to clustering schemes

From this figure, the estimation error of DGMA is smaller than Lowest-ID and it is in general about 2 seconds smaller as compared to Lowest-ID. This shows that DGMA has demonstrated its effectiveness over Lowest ID clustering scheme in partition prediction. According to experimental results of DGMA, the clusterhead changes are less than that in the Lowest-ID clustering scheme. Thus, cluster information kept by clusterheads has a longer lifetime that gives more valid and reliable mobility historical information for prediction. It can be concluded that suitable clustering schemes, e.g. considering mobility metrics or group topology, could improve the prediction accuracy of group partitions.

5.6 Conclusion

The motivation of this work rests with the aim to provide continuous communication in MANETs by implementing a network partition prediction scheme. By the presence of group partition prediction schemes, unaware network disconnection among adjacent groups can be predicted and possibly be prevented by certain actions, such as deploying communication sensors or enlarging radio

transmission range to maintain/restore connectivities [122]. In this chapter, we introduce a framework to carry out research in group partition prediction for MANETs. The involved modules include mobility models, clustering algorithms and safe distance for connectivity, each of which is a standalone research topic. The concept of the group safe distance based on group density is also discussed. The core module of this work is the mobility prediction algorithm which is based on the quadratic regression trend model. Its effectiveness and efficiency in predicting group partitions have been demonstrated from the experimental results. In addition, our discussion is extended by using different clustering algorithms, namely the Lowest ID and DGMA clustering schemes. It illustrates that using a suitable clustering algorithm does have a positive effect on prediction results. This work raises the issue of solutions for preventing and restoring network from disconnections as one of the future works.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

In this chapter, conclusions regarding the research work reported in this thesis will be made in Section 6.1. Based on the conclusions made, some recommendations for further research will be suggested in Section 6.2.

6.1 Conclusions

This thesis focuses on mobility issues for group communications in mobile ad hoc networks. By studying the group mobility characteristics, we investigated the problems that arise from group partitions and proposed different schemes for topology control and group partition predictions.

In Chapter 2, we investigated the mobility of groups in various applications of MANETs. Practically, mobile nodes with similar mobility patterns can form a group which can be further partitioned into multiple smaller subgroups or merged with other groups. We proposed a generic mobility model, which is referred to as the Reference Region Group Mobility model, to simulate the mobility of groups and entities in MANETs. The novelty of this model is its ability to model group operations, such as group partitions and mergers, for simulation studies in

MANETs. Based on this model, simulations of MANET applications in different scenarios are carried out, which include group mobility with randomly generated destinations, group mobility with predefined destinations, and entity's mobility. This mobility model generates the mobility environment that we want to investigate in the simulation studies of the following chapters in this thesis.

To address the issue of the impact of group partitions and mergers to the network performance of routing protocols in MANETs, we carried out a comprehensive evaluation and comparison for the performance of two well-known routing protocols, i.e. AODV and DSR, by varying mobility speed and node density respectively in Chapter 3. In addition, we reviewed a number of routing protocols proposed for MANETs which are generally classified into two categories, namely table-driven and on-demand routing protocols, and discussed their pros and cons. The simulation results show that the performance of both AODV and DSR are affected greatly by frequent group partitions. ADOV is more adaptive to mobility and topology changes than DSR in a relatively dynamic network environment. DSR yields a high end-to-end delay and large amount of control packets which in turn affects the packet delivery ratio. On the other hand, DSR is suitable to a network with less mobility where the load of route maintenance is not heavy. Furthermore, change of node density in a group will affect the relative distance between mobile nodes and hence it eventually generates an impact on the network performance of routing protocols.

As a step forward, we studied the issues in the area of network topology control, and proposed a clustering algorithm namely the *Distributed Group Mobility Adaptive* (DGMA) clustering algorithm, for mobile nodes in group mobility in

Chapter 4. DGMA is a distributed, mobility-aware clustering algorithm, and it is also adaptive to the group operations, i.e. group partitions and mergers, in a network with group mobility. A novel feature of this algorithm is the concept of *Linear Distance Based Spatial Dependency*, which is used as the mobility metric to capture a node's "significant movement" and therefore "micro" movement of mobile nodes could be discarded. With the aid of this mobility metric, the DGMA clustering algorithm can improve clusters' stability in terms of longer cluster lifetime, longer mean residence time for cluster members and less cluster reaffiliation events. In addition, our experimental studies show that DGMA outperforms the Lowest-ID and MOBIC clustering algorithms in all the above mentioned performance measures.

After proposing the group mobility model to mimic group operations, such as group partitions and mergers, and the network clustering algorithm for topology control in group mobility environment, we proposed a framework to investigate and study the issue of group partition prediction in MANETs in Chapter 5. This framework are composed by different modules which include the group mobility generator by the RRGm mobility model, the clustering algorithm using DGMA, group mobility information gathering, mobility prediction using quadratic regression trend model, safe distance analysis for connectivity between groups, and group partition time estimation. From the simulation results, this framework is effective and efficient in predicting potential group partitions in MANETs. In particular, our discussion is further extended by comparing the prediction results by using different clustering algorithms, namely Lowest-ID and DGMA, in order to investigate the impact of clustering algorithms on this framework. Our studies show

that DGMA can help to improve the prediction accuracy for 2 seconds on average as compared to the results obtained by Lowest-ID.

6.2 Recommendation for Future Work

Currently, much research work has been done with regard to network communications in MANETs. However efforts to improve group communications in MANETs have not kept pace. Considering the nature of MANET applications which are proliferated in many group related scenarios, MANETs require more effective and efficient support for group communications. There are some recommendations on further research activities that can be carried out.

One important aspect that researchers can pursue more actively is to improve the reliability of connectivities of mobile nodes in order to support group communications. It becomes extremely challenging to obtain a reliable network for group communications when network size and mobility increase. Thus, to enhance the connectivities between mobile nodes is one of the critical approaches. Research can be focused on the study of the mobility behaviors of mobile nodes and mobility metrics to quantitatively describe the mobility of nodes, It can be used to identify those “inactive” nodes existing in a network in order to establish a backbone of “reliable” connectivities in the network.

As the network size and mobility increase, any network routing protocol will become less reliable and effective. Although flooding can be adapted for network communications in most circumstance, it is definitely not a good choice since it requires too much network resource and generates extraordinary heavy traffic load. Thus, routing protocols to support group communications would be another future

research direction. To the best of our knowledge, there is no existing routing protocol that are proposed to support group communications in MANETs. However, it would not easy to design such kinds of routing protocols as it would be possibly concerned with both unicasting and multicasting, and the mobility behavior of a group is much more complex than the random mobility of an entity.

Based on “reliable” connectivities in the network, the DGMA clustering algorithm could be improved with the aim of more stable cluster structures, which will in turn help to design routing protocols to support group communications in MANETs. Stable cluster structures can also increase the accuracy of group partition predictions. Hence, remedial methods to avoid network from disconnections could also be studied, e.g., deploying intermediate nodes or on-demand wakeup of nodes in the network to maintain connectivities between mobile groups.

APPENDIX-A

The purpose of this appendix is to explain how the equation (5.10.3) is deduced.

As we have (5.10.1)

$$\frac{\partial Y}{\partial \beta_0} = 0, \frac{\partial Y}{\partial \beta_1} = 0, \frac{\partial Y}{\partial \beta_2} = 0 \quad (5.10.1)$$

Thus,

$$\begin{aligned} \frac{\partial Y}{\partial \beta_0} = 0 &\Rightarrow \\ -2 \sum_{j=1}^k S_i(t_j) + 2 \sum_{j=1}^k \beta_0 + 2 \sum_{j=1}^k \beta_1 t_j + 2 \sum_{j=1}^k \beta_2 t_j^2 &= 0 \\ 2 \sum_{j=1}^k \beta_0 + 2 \sum_{j=1}^k \beta_1 t_j + 2 \sum_{j=1}^k \beta_2 t_j^2 &= 2 \sum_{j=1}^k S_i(t_j) \\ k \beta_0 + \beta_1 \sum_{j=1}^k t_j + \beta_2 \sum_{j=1}^k t_j^2 &= \sum_{j=1}^k S_i(t_j) \end{aligned} \quad (5.10.1.1)$$

Similarly,

$$\begin{aligned} \frac{\partial Y}{\partial \beta_1} = 0 &\Rightarrow \\ \beta_0 \sum_{j=1}^k t_j + \beta_1 \sum_{j=1}^k t_j^2 + \beta_2 \sum_{j=1}^k t_j^3 &= \sum_{j=1}^k t_j S_i(t_j) \end{aligned} \quad (5.10.1.2)$$

$$\frac{\partial Y}{\partial \beta_2} = 0 \Rightarrow$$

$$\beta_0 \sum_{j=1}^k t_j^2 + \beta_1 \sum_{j=1}^k t_j^3 + \beta_2 \sum_{j=1}^k t_j^4 = \sum_{j=1}^k t_j^2 S_i(t_j) \quad (5.10.1.3)$$

From the above three equations:

$$\begin{bmatrix} k & \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 \\ \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 \\ \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 & \sum_{j=1}^k t_j^4 \end{bmatrix} * \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^k S_i(t_j) \\ \sum_{j=1}^k t_j S_i(t_j) \\ \sum_{j=1}^k t_j^2 S_i(t_j) \end{bmatrix} \quad (5.10.2)$$

We define:

$$T = \begin{bmatrix} k & \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 \\ \sum_{j=1}^k t_j & \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 \\ \sum_{j=1}^k t_j^2 & \sum_{j=1}^k t_j^3 & \sum_{j=1}^k t_j^4 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \text{ and } S = \begin{bmatrix} \sum_{j=1}^k S_i(t_j) \\ \sum_{j=1}^k t_j S_i(t_j) \\ \sum_{j=1}^k t_j^2 S_i(t_j) \end{bmatrix}$$

(5.10.2) can be represented as:

$$T\beta = S$$

$$(TT^T)\beta = ST^T \quad (5.10.3)$$

$$\beta = (TT^T)^{-1}T^T S$$

AUTHOR'S PUBLICATIONS

- Yan Zhang, Chor Ping Low, Jim Mee Ng, and Ting Wang, "Predicting Group Partitions in Mobile Ad Hoc Networks", *Wiley: International Journal of Communication Systems*, Wiley Online Library, DOI: 10.1002/dac.1333, 2011.
- Yan Zhang, Chor Ping Low, Jim Mee Ng, "Performance Evaluation of Routing Protocols on the Reference Region Group Mobility Model for MANETs", *Journal of Wireless Sensor Networks*, Scientific Research, vol.3, No.3, March 2011, pp. 92-105.
- Yan Zhang, Jim Mee Ng, Chor Ping Low "A Distributed Group Adaptive Clustering Algorithm for Ad Hoc Networks", *Computer Communications*, Elsevier, vol. 32, no. 1, 2009, pp. 189-202.
- Yan Zhang, Chor Ping Low, Jim Mee Ng, and Ting Wang, "An efficient group partition prediction scheme for MANETs", The 2009 IEEE conference on Wireless Communications & Networking Conference (IEEE WCNC 2009), Budapest, Hungary, 2009, pp. 2816-2821.
- Yan Zhang and Jim Mee Ng, "A Distributed Group Mobility Adaptive Clustering Algorithm for Mobile Ad Hoc Networks", IEEE International Conference on Communications (IEEE ICC 2008), Beijing, 2008, pp. 3161-3165.

- Jim Mee Ng, Yan Zhang, “Realistic Group Mobility Modeling and Network Performance Evaluation”, the International Conference on Late Advances in Networks, 2006 (ICLAN'2006), Paris, France, 2006, *pp.* 1-5.
- Jim Mee Ng, Yan Zhang, “Impact of Group Mobility on Ad hoc Networks Routing Protocols”, the IEEE 8th International Conference on Advanced Communication Technology, 2006 (IEEE ICACT 2006), Korea, 2006, *pp.*1129-1134.
- Jim Mee Ng, Yan Zhang, “A Mobility Model with Group Partitioning for Wireless Ad hoc Networks”, the IEEE International Conference on Information Technology and Applications 2005 (IEEE ICITA 05), Sydney, 2005, *pp.*289-294.
- Jim Mee Ng, Yan Zhang, “Reference Region Group Mobility Model for Ad hoc Networks", the Second IEEE International Conference on Wireless and Optical Communications Networks (IEEE WOCN 2005), Dubai, 2005, *pp.* 290-294.

BIBLIOGRAPHY

- [1] L.Kleinrock and F.Kamoun, "Hierarchical routing for large networks", *Computer Networks*, vol.1, no.3, 1977, pp.155-174.
- [2] Jie Wu and I. Stojmenovic, "Ad hoc networks", *IEEE Computer*, vol.37, no.2, Feb 2004, pp.29 - 31.
- [3] IETF MANET Working Group,
<http://www.ietf.org/html.charters/manetcharter.html> and
<http://www.ietf.org/rfc/rfc2501.txt>.
- [4] Freebersyser and B. Leiner, "A DoD Perspective on Mobile Ad Hoc Networks", in *Ad Hoc Networking*, C. E. Perkins Addison-Wesley, 2001, pp. 29-51.
- [5] IETF MANET Working Group.
<http://www.ietf.org/html.charters/manetcharter.html>.
- [6] R. Ramanathan and J. Redi., "A brief overview of ad hoc networks: challenges and directions", *IEEE Communications Magazine*, vol.40, no.5, May 2002, pp.20-22.
- [7] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communication & Mobile Computing (WCMC), Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol.2, no.5, 2002, pp.483-502.
- [8] F. Bai, N. Sadagopan, and A. Helmy, "Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks", *IEEE Information Communications Conference (INFOCOM 2003)*, San Francisco, USA, Apr. 2003, pp. 85-91.

- [9] Osamah S. Badarneh and Michel Kadoch, "Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy", *EURASIP Journal on Wireless Communications and Networking*, vol.2009, no.1, 2009, pp.1-42.
- [10] H. Moustafa and H. Labiod, "A performance comparison of multicast routing protocols in ad hoc networks", 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications (PIMRC 2003), Beijing, China, 7-10 Sept. 2003, pp. 497-501.
- [11] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom98), ACM, Dallas, Texas, United States, October 25-30, 1998, pp. 85-97.
- [12] E. M. Royer and Chai-Keong Toh, "A review of current routing protocols for ad hoc mobile wireless networks", *Personal Communications, IEEE*, vol.6, no.2, 06 August 2002, pp.46-55.
- [13] Samba Sesay, Zhongkai Yang, Biao Qi, and Jianhua He, "Simulation Comparison of Four Wireless Ad hoc Routing Protocols", *Information Technology Journal*, vol.3, no.1, 2004, pp.219-226.
- [14] Liu George and Jr. Gerald Maguire, "A class of mobile motion prediction algorithms for wireless mobile computing and communication", *Special issue: routing in mobile communications networks*, vol.1, no.2, 1996, pp.113-121.
- [15] R. Chellappa, A. Jennings, and N. Shenoy, "A Comparative Study of Mobility Prediction in Cellular and Ad Hoc Wireless Networks", IEEE International Conference on Communications (ICC2003), Alaska, USA, May 2003, pp. 194-199.
- [16] K.H Wang and Baochun Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks", IEEE Information Communications Conference (INFOCOM 2002), New York, USA, June 2002, pp. 1089-1098.

- [17] Shengming Jiang, Dajiang He, and Jiangqiang Rao, "A Prediction-Based Link Availability Estimation for Routing Metrics in MANETs", *IEEE/ACM Transactions on Networking*, vol.13, no.6, Dec. 2005, pp.1302-1312.
- [18] W. Su, S.-J. Lee, and M. Gerla, "Mobility Prediction and Routing in Ad Hoc Wireless Networks", *International Journal of Network Management*, vol.11, no.1, 2001, pp.3-30.
- [19] B.S. Ghahfarokhi and N. Movahedinia, "Neural Network based Mobility Prediction for Handoff Management in Wireless MPLS Network", *Conference of Networks and Communication Systems 2006*, Chiang Mai, Thailand, 2006, pp. 74-80.
- [20] M. Hauspie, J. Carle, and D. Simplot, "Partition detection in mobile ad-hoc networks using multiple disjoint path set", *The 1st International Workshop on Objects Models and Multimedia Technologies (OMMT)*, Geneva, Switzerland, 2003, pp. 25-27.
- [21] Aruna Seneviratne Jonathan Chan, "A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks", *IEEE International Conference on Networks (ICON)*, Singapore, 1999, pp. 104-111.
- [22] Qingfeng Huang, Christine Julien, and Gruia-Catalin Roman, "Relying on Safe Distance to Achieve Strong Partitionable Group Membership in Ad Hoc Networks", *IEEE Transaction on Mobile Computing*, vol.3, no.2, 2004, pp.192-205.
- [23] R. Chellappa Doss, A. Jennings, and N. Shenoy, "A Review of Current Work on Mobility Prediction Techniques in Mobile Ad Hoc Networks", *2004 IASTED International Conference on Wireless Networks and Emerging Technologies*, Wireless Networks and Emerging Technologies, Canada, 2004, pp. 536-543.
- [24] R. Chellappa Doss, A. Jennings, and N. Shenoy, "Mobility Prediction based Routing for Minimizing Control Overhead in Mobile Ad hoc Networks," *International Conference on Wireless Networks (ICWN 2004)*, Las Vegas, USA, June 2004, pp. 27-31.

- [25] R. A. Guérin, "Channel Occupancy Time Distribution in a Cellular Radio System", *IEEE Transactions on Vehicular Technology*, vol.35, no.3, Aug. 1987, pp.89-99.
- [26] I. Stepanov, P. J. Marron, and K. Rothermel, "Mobility Modeling of Outdoor Scenarios for MANETs", The 38th Annual Simulation Symposium, Germany, Apr. 2005, pp. 312-322.
- [27] J. Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas, "Mobility modeling in third generation mobile telecommunication systems", *IEEE Personal Communications*, vol.4, no.4, August 1997, pp.41-56.
- [28] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks", ACM Mobicom, 2003, pp. 217-229.
- [29] C. E. Perkins and E. M. Royer., "Ad-hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, USA, February 1999, pp. 90-100.
- [30] D.Johnson and D.Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", in *Mobile computing*. vol.5 Kluwer Academic, 1996, pp. 153--181.
- [31] S. Gowrishankar, S. Sarkar, and T. G. Basavaraju, "Simulation Based Performance Comparison of Community Model, GFMM, RPGM, Manhattan Model and RWP-SS Mobility Models in MANET", First International Conference on Networks and Communications, 2009 (NETCOM '09), Chennai, Thailand, 2009, pp. 408-413.
- [32] M. Sanchez and P. Manzoni, "A Java-Based Ad Hoc Networks Simulator", Proceedings of the SCS Western Multi-conference Web-based Simulation Track, Jan.1999, pp. 287-294.
- [33] K Pearson, "The problem of the Random Walk", *Nature*, vol.72, no.1, 1905, pp.294.
- [34] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless networks", ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), Seattle, WA, USA, August 1999, pp. 53-60.

- [35] Karen Wang and Baochun Li, "Group Mobility and Partition Prediction in Wireless Ad-hoc Networks", IEEE International Conference on Communications (ICC 2002), New York City, USA, 2002, *pp.* 1017-1021.
- [36] Zhao Ming and Wang Wenye, "Design and Applications of A Smooth Mobility Model for Mobile Ad Hoc Networks", IEEE Military Communications Conference, 2006 (MILCOM 2006), Washington, Ellipse Group Mobilit DC, USA, 2006, *pp.* 1-7.
- [37] T. Soungalo, Dong Wang, and Yulan Liu, "Performance Evaluation of Campus Mobility Model in Mobile Ad Hoc Networks", International Conference on New Trends in Information and Service Science, 2009 (NISS '09), Beijing, China, 2009, *pp.* 743-748.
- [38] C. Papageorgiou, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "An obstacle-aware human mobility model for ad hoc networks", IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009 (MASCOTS '09), London, UK, 2009, *pp.* 1-9.
- [39] Di Wu, Xiaofeng Zhu, and Xin Wang, "Analysis of 3-D Random Direction Mobility Model for Ad Hoc Network", 6th International Conference on ITS Telecommunications, Chengdu, China, 2006, *pp.* 741-744.
- [40] Li Demin, Zhou Jie, Wang Jiacun, and Yu Peng, "Linking Generation Rate Based on Gauss-Markov Mobility Model for Mobile Ad Hoc Networks", International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09), Wuhan, Hubei, China, 2009, *pp.* 358-361.
- [41] S. Ahmed, G. C. Karmakar, and J. Kamruzzaman, "Geographic Constraint Mobility Model for Ad Hoc Network", IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems (MASCOTS 2008), Baltimore, MD, USA, 2008, *pp.* 1-10.
- [42] Ting Hu, XiaoHong Li, Dong Wang, and DaFang Zhang, "IIMM: An Interest Induced Mobility Model for the Simulation of Mobile Ad Hoc Networks", International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007), Shanghai, China, 2007, *pp.* 1461-1464.

- [43] Shiddhartha Raj Bhandari, Gyu MOUNG Lee, and Noel Crespi, "Mobility Model for User's Realistic Behavior in Mobile Ad Hoc Network", The Eighth Annual Communication Networks and Services Research Conference (CNSR 2010), Montreal, QC, Canada, 2010, *pp.* 102-107.
- [44] A. E. Kamal and J. N. Al-Karaki, "A New Realistic Mobility Model for Mobile Ad Hoc Networks", IEEE International Conference on Communications (ICC '07), Glasgow, Scotland, UK, 2007, *pp.* 3370-3375.
- [45] Chen Chun-Hung, Wu Ho-Ting, and Ke Kai-Wei, "A Novel Mobility Model of Diverse Average Speed and Uniform Node Distribution in Mobile Ad Hoc Networks", IEEE Region 10 Conference (TENCON 2006), Hong Kong, China, 2006, *pp.* 1-4.
- [46] Zhao Chen and M. L. Sichitiu, "N-Body: Social Based Mobility Model for Wireless Ad Hoc Network Research", The 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON 2010), Boston, MA, USA, 2010, *pp.* 1-9.
- [47] V. Vetriselvi and R. Parthasarathi, "Trace Based Mobility Model for Ad Hoc Networks", Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007), White Plains, NY, USA, 2007, *pp.* 81-81.
- [48] B. Zhou, K. Xu, and M. Gerla, "Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks", IEEE Military Communications Conference, 2004 (MILCOM 2004), Monterey, Canada, 2004, *pp.* 289-294.
- [49] Ning Li, Yan Guo, Chao Dong, and Jinlong Wang, "Diamond Group Mobility Model for ad hoc network in military", 9th International Conference on Signal Processing (ICSP 2008), Beijing, China, 2008, *pp.* 2754-2756.
- [50] K. Wu and Q. Yu, "A Multi-Group Coordination Mobility Model for Ad Hoc Networks", IEEE Military Communications Conference (MILCOM 2006) Washington, USA, 2006, *pp.* 1-5.
- [51] E. Ochirsuren, L. S. Indrusiak, and M. Glesner, "An Actor-Oriented Group Mobility Model for Wireless Ad Hoc Sensor Networks", 28th International

- Conference on Distributed Computing Systems Workshops (ICDCS '08), Beijing, China, 2008, *pp.* 174-179.
- [52] Chen Wenjie, Dou Lihua, Li Yujin, and Wei Yunchuan, "Ellipse Group Mobility model for tactical MANET", Chinese Control and Decision Conference (CCDC 2010), Xuzhou, China, 2010, *pp.* 2288-2293.
 - [53] M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Pattern", *IEEE Journal on Selected Areas in Communications*, vol.15, no.7, September 1997, *pp.*1239-1252.
 - [54] I. Rubin and C. Choi, "Impact of the Location Area Structure on the Performance of Signaling Channels in Wireless Cellular Networks", *IEEE Communications Magazine*, vol.35, no.2, February 1997, *pp.*108-115.
 - [55] Vincent Gauthier, "Mobility model in ad hoc network, <http://www-public.it-sudparis.eu/~gauthier/MobilityModel/mobilitymodel.html>," 2009.
 - [56] E. Royer, P.M. Melliar-Smith, and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks", IEEE International Conference on Communications (ICC), 2001, Helsinki, Finland, 2001, *pp.* 857-861.
 - [57] NS2-Nam, "<http://www.isi.edu/nsnam/ns/>."
 - [58] S. J. Lee, W. Su, and M. Gerla, "Wireless Ad hoc Multicast Routing with Mobility Prediction", *ACM/Kluwer Mobile Networks and Applications*, vol.6, 2000, *pp.*351-360.
 - [59] A. Alshanyour and U. Baroudi, "Random and realistic mobility models impact on the performance of bypass-AODV routing protocol", 1st IFIP Wireless Days, 2008 (WD '08), Dubai, UAE, 2008, *pp.* 1-5.
 - [60] M. Bouhorma, H. Bentaouit, and A. Boudhir, "Performance comparison of ad-hoc routing protocols AODV and DSR", International Conference on Multimedia Computing and Systems (ICMCS '09), Canada, 2009, *pp.* 511-514.
 - [61] E. M. Royer and Toh Chai-Keong, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *Personal Communications, IEEE*, vol.6, no.2, 1999, *pp.*46-55.
 - [62] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", The

- SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, London, UK, August 1994, *pp.* 234–244.
- [63] Murthy Shree and J. J. Garcia Luna Aceves, "An efficient routing protocol for wireless networks", *Mobile Networks and Applications - Special issue: routing in mobile communications networks*, vol.1, no.2, 1996, *pp.*183-197.
 - [64] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks", *IEEE Technology for the 21st Century (INMIC 2001)*, Pakistan, 2001, *pp.* 62-68.
 - [65] G. Pei, M. Gerla, X. Hong, and C. C. Chiang, "A wireless hierarchical routing protocol with group mobility", *IEEE Wireless Communications and Networking Conference (WCNC 1999)* New Orleans, LA , USA, 1999, *pp.* 1538-1542.
 - [66] P. Samar, M. R. Pearlman, and Z. J. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks", *Networking, IEEE/ACM Transactions on*, vol.12, no.4, 2004, *pp.*595-608.
 - [67] Zhou Biao, Lee Yeng-Zhong, and M. Gerla, "Direction assisted geographic routing for mobile ad hoc networks", *IEEE Military Communications Conference (MILCOM 2008)*, San Diego, CA, USA, 2008, *pp.* 1-7.
 - [68] A. N. Ai-Khwildi, S. R. Chaudhry, Y. K. Casey, and H. S. Ai-Raweshidy, "Mobility Factor in WLAN-Ad hoc On-Demand Routing Protocols", *IEEE 9th International Multitopic Conference (INMIC 2005)*, Karachi, 2005, *pp.* 1-6.
 - [69] M. Bansal and G. Barua, "Performance comparison of two on-demand routing protocols for mobile ad hoc networks", *2002 IEEE International Conference on Personal Wireless Communications*, New Delhi, India, 2002, *pp.* 206-210.
 - [70] www.scalable-networks.com.
 - [71] Zhang Mingyang and P. Chong, "Performance Comparison of Flat and Cluster-Based Hierarchical Ad Hoc Routing with Entity and Group Mobility", *IEEE Wireless Communications and Networking Conference*, 2009, Budapest, Hungary, 5-8 April 2009, *pp.* 1-6.
 - [72] Inn ER and Winston K.G. Seah, "Mobility-based d-hop clustering algorithm for mobile ad hoc networks", *IEEE Wireless Communications and*

- Networking Conference (WCNC 2004), Atlanta, USA, 2004, pp. 2359-2364.
- [73] E. Sakhaee and A. Jamalipour, "Stable Clustering and Communications in Pseudolinear Highly Mobile Ad Hoc Networks", *IEEE Transactions on Vehicular Technology*, vol.57, no.6, 2008, pp.3769-3777.
 - [74] Hou Ting-Chao and Tsai Tzu-Jane, "A access-based clustering protocol for multihop wireless ad hoc networks", *IEEE Journal on Selected Areas in Communications*, vol.19, no.7, 2001, pp.1201-1210.
 - [75] Ryu Jung-hee, Song Sanghwa, and Cho Dong-Ho, "New clustering schemes for energy conservation in two-tiered mobile ad hoc networks", *IEEE Transactions on Vehicular Technology*, vol.51, no.6, 2002, pp.1661-1668.
 - [76] Leng Supeng, Zhang Liren, Fu Huirong, and Yang Jianjun, "A Novel Location-Service Protocol Based on K-Hop Clustering for Mobile Networks", *IEEE Transactions on Vehicular Technology*, vol.56, no.2, 2007, pp.810-817.
 - [77] Wang Zhigang, Liu Lichuan, Zhou MengChu, and N. Ansari, "A Position-Based Clustering Technique for Ad Hoc Intervehicle Communication", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.38, no.2, 2008, pp.201-208.
 - [78] Leng Supeng, Zhang Yan, Chen Hsiao-Hwa, Zhang Liren, and Liu Ke, "A novel k-hop Compound Metric Based Clustering scheme for ad hoc wireless networks", *IEEE Transactions on Wireless Communications*, vol.8, no.1, 2009, pp.367-375.
 - [79] H. Gharavi and K. Ban, "Master-slave cluster-based multihop ad-hoc networking", *Electronics Letters*, vol.38, no.25, 2002, pp.1756-1757.
 - [80] R. T. Goonewardene, F. H. Ali, and E. Stipidis, "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks", *IET Intelligent Transport Systems*, vol.3, no.2, 2009, pp.148-158.
 - [81] S Basagni, "Distributed Clustering for Ad Hoc Networks", The 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99), Washington, DC, USA, 1999, pp. 310 - 315.

-
- [82] Received signal strength indication (RSSI),
http://en.wikipedia.org/wiki/Received_signal_strength_indication.
- [83] A.Ephremides, J.E.Wieselthier, and D.J.Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling", *Proceedings of the IEEE*, vol.75, no.1, 1987, pp.56-73.
- [84] Basu, N. Khan, and T.D.C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing, Phoenix, AZ, USA, April 2001, pp. 413 - 418.
- [85] S.K. Dhurandher and G.V. Singh, "Weight based adaptive clustering in wireless ad hoc networks", 2005 IEEE International Conference on Personal Wireless Communications, New Delhi, India, 2005, pp. 95-100.
- [86] J. Lian, G. Agnew, and K. Naik, "A Variable-degree Based Clustering Algorithm for Networks", IEEE International Conference in Computer Communication and Networking, November 2003, Dallas, Texas, USA, 2003, pp. 465 - 470.
- [87] H. Cheng, Cao, J., Wang, X., and Das, S. K., "Stability-based multi-objective clustering in mobile ad hoc networks", The 3rd international Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Waterloo, Ontario, Canada, 2006, pp. 27-32.
- [88] Venkataraman Gayathri, Emmanuel Sabu, and Thambipillai Srikanthan, "Size-restricted cluster formation and cluster maintenance technique for mobile ad hoc networks", *International Journal of Networking Management*, vol.17, no.2, 2007, pp.171-194.
- [89] Colette Johnen and Le Huy Nguyen, "Self-Stabilizing weight-based Clustering Algorithm for Ad hoc sensor Networks", *Lecture notes in computer science*, 2006, pp.83-93.
- [90] C. Bettstetter and R. Krausser, "Scenario-based stability analysis of the distributed mobility-adaptive clustering (DMAC) algorithm", The 2nd ACM international symposium on mobile ad hoc networking & computing, Long Beach, CA, USA, 2001, pp. 232-241.

- [91] C. C. Chiang, "Routing in Clustered Multi-hop, Mobile Wireless Networks with Fading Channel", The IEEE Singapore International Conference on Networks, Singapore, 1997, *pp.* 197-211.
- [92] 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, Aug 1999, *pp.*1466-1487.
- [93] S. Chinara and S. K. Rath, "Mobility Based Clustering Algorithm and the Energy Consumption Model of Dynamic Nodes in Mobile Ad Hoc Network", International Conference on Information Technology (ICIT 2008), Rome, Italy, 2008, *pp.* 171-176.
- [94] Y. Fernandess and D. Malkhi, "K-clustering in wireless ad hoc networks", The second ACM international workshop on Principles of mobile computing, POMC'02, France, Oct. 2002, *pp.* 31-37.
- [95] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks", *IEEE Transactions on Mobile Computing*, vol.3, no.4, 2004, *pp.*366-379.
- [96] C. C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks", *Journal of Cluster Computing: Special Issue on Mobile Computing*, vol.1, no.2, 1998, *pp.*187-196.
- [97] A. Fathi and H. Taheri, "Enhance topology control protocol (ECEC) to conserve energy based clustering in wireless ad hoc networks", 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), Beijing, China, *pp.* 356-360.
- [98] S. Chinara and S. K. Rath, "Energy Efficient Mobility Adaptive Distributed Clustering Algorithm for Mobile Ad Hoc Network", 6th International Conference on Advanced Computing and Communications (ADCOM 2008), Chennai, India, 2008, *pp.* 265-272.
- [99] Ni Minming, Zhong Zhangdui, Wu Hao, and Zhao Dongmei, "An Energy Efficient Clustering Scheme for Mobile Ad Hoc Networks", 2010 IEEE 71st Vehicular Technology Conference (VTC 2010-Spring), Taipei, Taiwan, *pp.* 1-5.

- [100] S. H. Bouk and I. Sasase, "Energy Efficient and Stable Weight Based Clustering for mobile ad hoc networks", 2nd International Conference on Signal Processing and Communication Systems (ICSPCS 2008), Gold Coast, AUS, 2008, *pp.* 1-10.
- [101] Feng Wei and J. M. H. Elmirghani, "Energy Efficiency in the Cluster-Based Linear Ad-hoc Wireless Networks", Third International Conference on Next Generation Mobile Applications, Services and Technologies, 2009 (NGMAST 2009), Cardiff, Wales, 2009, *pp.* 395-400.
- [102] Lei Zhang and Xuehui Wang, "An Energy Balanced Clustering Algorithm for Mobile Ad Hoc Networks", 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009 (WiCom 2009), Beijing, China, 2009, *pp.* 1-4.
- [103] E. Astier, A. Hafid, and A. Benslimane, "Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks", IEEE Wireless Communications and Networking Conference, 2009 (WCNC 2009), Budapest, Hungary, 2009, *pp.* 1-6.
- [104] T. J. Kwon and M. Gerla, "Efficient flooding with Passive Clustering (PC) in ad hoc networks", *Proceedings of the IEEE*, vol.91, no.8, 2002, *pp.*1210 - 1220.
- [105] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Hong Kong, China, 2004, *pp.* 629-640.
- [106] F. D. Tolba, D. Magoni, and P. Lorenz, "Connectivity, Energy and Mobility Driven Clustering Algorithm for Mobile Ad Hoc Networks", IEEE Global Telecommunications Conference, 2007 (GLOBECOM '07), Washington, DC, USA, 2007, *pp.* 2786-2790.
- [107] Geng Chen, F.G. Nocetti, J.S. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks", the 35th Annual Hawaii International Conference on System Sciences, 2002 (HICSS 2002), Hawaii, USA, 7-10 Jan. 2002, *pp.* 2450 - 2459.
- [108] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless Networks", IEEE Twentieth Annual Joint Conference of

- the IEEE Computer and Communications Societies (INFOCOM 2001), Anchorage, AK, USA, 2001, *pp.* 1028-1039.
- [109] Wei Dali and H. A. Chan, "Clustering Ad Hoc Networks: Schemes and Classifications", 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON 2006), Reston, VA, USA, 2006, *pp.* 920-926.
 - [110] C. Bettstetter, "The cluster density of a distributed clustering algorithm in ad hoc networks", the IEEE International Conference on Communications (ICC 2004), New York, USA, 2004, *pp.* 4336 - 4340.
 - [111] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks", *IEEE Journal on Selected Areas in Communications*, vol.15, no.7, 1997, *pp.*1265-1275.
 - [112] T. C. Hou and T. J. Tsai, "An access-based clustering protocol for multihop wireless Networks", *IEEE Journal on Communications*, vol.19, no.7, July 2001, *pp.*1201-1210.
 - [113] Chiang Ching-Chuan and M Gerla, "Routing and multicast in multi-hop, mobile wireless networks", IEEE 6th International Conference on Universal Personal Communications, San Diego, CA, USA, 1997, *pp.* 546-551.
 - [114] R. Palit, E. Hossain, and P. Thulasiraman, "Mobility-aware pro-active low energy (MAPLE) clustering in ad hoc mobile wireless networks", IEEE Global Telecommunications Conference, 2004 (GLOBECOM '04), Dallas, USA, 2004, *pp.* 3426-3430.
 - [115] Ryu Jung-hee, Song Sanghwa, and Cho Dong-Ho, "New clustering schemes for energy conservation in two-tiered mobile ad hoc networks", *IEEE Transactions on Vehicular Technology*, vol.51, no.6, 2002, *pp.*1661-1668.
 - [116] F. Foroozan and S. Datta, "A Low-Maintenance Energy-Aware Clustering Algorithm for Wireless Ad-hoc Networks", IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2006), Montreal, Canada, 2006, *pp.* 457-462.
 - [117] A. K. Parekh, "Selecting routers in ad hoc wireless networks", the SBT/IEEE International Telecommunications Symposium, ITS, Rio de Janeiro, August 1994, *pp.* 1-18.

- [118] Yu Jane Y and Peter H.J.Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", *IEEE Communications Surveys & Tutorials*, vol.7, no.1, 2005, pp.32-48.
- [119] Jing Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation", 35th Annual Proceedings of Simulation Symposium, 2002, California, USA, 14-18 April 2002, pp. 337 - 344.
- [120] Wikipedia, "Non-linear least squares," in http://en.wikipedia.org/wiki/Non-linear_least_squares, 2009.
- [121] Bowerman, O' Connell, and Koehler, *Forecasting, Time Series, and Regression*, Fourth Edition, USA: Thomson Brooks/Cole, 2005.
- [122] Huang Jiun-Long and Chen Ming-Syan, "On the effect of group mobility to data replication in ad hoc networks", *IEEE Transactions on Mobile Computing*, vol.5, no.5, 2006, pp.492-507.