Clustering and routing in mobile ad hoc networks (MANETs)

Yu, Yang

2008

Yu, Y. (2008). Clustering and routing in mobile ad hoc networks (MANETs). Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41844>

<https://doi.org/10.32657/10356/41844>

Clustering and Routing in Mobile Ad Hoc Networks (MANETs)

YU Yang

School of Electrical & Electronic Engineering

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

2008

Statement of Originality

I hereby certify that the work embodied in this Thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

17/10/2008

Date

YU Yang

YU Yang

Acknowledgements

First of all, I would like to express my sincere thanks and great gratitude to my supervisor, Assistant Professor Chong Han Joo Peter. Without his intellectual guidance, constant attention and strict demands, this work would have not been completed. His extensive knowledge, enlightened direction and continuous encouragement make my Thesis work much easier. In particular, I wish to thank him for teaching me some of the philosophical aspects of research and some intangible skills, which I believe are the most invaluable knowledge that I have acquired as a PhD student.

I would like to thank the centre director, Associate Professor Shum Ping, technicians and many labmates, especially Dr. Li Xuejun and Dr. Yang Ming, in Network Technology Research Centre for their guidance, support and help in the past few years.

I would like to express my gratitude to my family, especially my husband and parents, for their consistent support and encouragement during my PhD study and my life.

Last, but not the least, I would like to thank Nanyang Technological University for providing all the study facilities and research scholarship.

Abstract

Mobile Ad Hoc Networks (MANETs), without any fixed infrastructures, allow wireless communication terminals to build communication networks anytime and anywhere. Hence, MANETs bear great application potential and become a hot research topic in recent years. However, MANETs still face a lot of challenging problems, which may greatly affect their performance and application in practical situations. In this Thesis, we mainly focus on two challenging problems of MANETs: scalability and energy limitation, and come up with some clustering and routing protocols to solve the addressed problems.

A MANET with flat structure encounters scalability problem with increased network size, in particular in face of node mobility. Cluster structure, as a typical hierarchy architecture, is essential for solving the scalability problem of MANETs and achieving performance guarantee in a MANET with moderate to large size. In this Thesis, we first give a comprehensive survey on some typical clustering schemes proposed in MANET research. We give the fundamental knowledge for MANET clustering, show the pros and cons of cluster based MANETs, categorize those proposed clustering schemes based on their objectives, and discuss their mechanisms, cost and feasible applications in detail.

Secondly, we propose a clustering scheme, named efficient clustering scheme (ECS), for large and dense MANETs. With the introduction of a new cluster-related status, named clusterguest (CG), and a cluster deletion mechanism, ECS can effectively eliminate small and unnecessary clusters and hence effectively reduce the cluster

overlapping in a MANET with moderate to high node density, which is helpful in simplifying the network structure. Results show that ECS can provide a stable cluster structure as well for upper layer applications.

Following that, we implement the routing mechanisms of Cluster Based Routing Protocol (CBRP) on the top of the cluster structure built and maintained by ECS. To make full use of the cluster structure maintained by ECS, some routing mechanisms are modified accordingly so that the routing space can be further reduced without affecting the routing efficiency as compared to CBRP. Also, an information table update mechanism by utilizing routing events and data delivery events are introduced to enhance the clustering performance and routing performance. Such a routing scheme is named Efficient Cluster Based Routing Protocol (ECBRP) in the Thesis. The performance of ECBRP in terms of packet delivery ratio, normalized routing overhead and end-to-end transmission delay is compared with that of CBRP and dynamic source routing (DSR).

Energy limitation poses a severe challenge for the performance of typical MANETs, in which mobile nodes are supplied with battery power. Routing is greatly related with the energy consumption of mobile nodes in a MANET because routing decides the mobile nodes that are involved as data relays. Hence, energy-related metrics should be integrated with routing to achieve wise energy utilization for an energy-limited MANET. The final part of this Thesis proposes a routing scheme, named hybrid energy-aware routing (HEAR), by introducing a time delay mechanism in the route discovery, an adaptive Hello mechanism and a local route adjustment and salvage mechanism. The results show that HEAR can achieve satisfying performance in an energy-limited MANET in terms of network throughput and network lifetime.

List of Abbreviations

AMC	Adaptive Multi-hop Clustering
AODV	Ad-hoc On-demand Distance-Vector Routing
AP	Access Point
BRP	Bordercast Resolution Protocol
CAT	Cluster Adjacency Table
CBRP	Cluster Based Routing Protocol
CDS	Connected Dominating Set
CG	Clusterguest
CGSR	Clusterhead Gateway Switch Routing
CGW	Clustergateway
CH	Clusterhead
CID	Cluster ID
CM	Clustermember
CMC	Combined-Metrics-based Clustering
CMMBCR	Conditional Max-Min Battery Capacity Routing
CRID	Calculated Route ID
DCF	Distributed Coordinator Function
DDCA	Distributed Dynamic Clustering Algorithm
DLBC	Degree Load Balancing Clustering
DS	Dominating Set
DSC	Dominating-Set-based Clustering

List of Abbreviations

DSR	Dynamic Source Routing
DSDV	Destination-Sequenced Distance Vector Routing
DV	Distance Vector
ECBRP	Efficient Cluster Based Routing Protocol
ECBRP-S	Simple version of ECBRP
ECS	Efficient Clustering Scheme
EEC	Energy-Efficient Clustering
FSR	Fisheye State Routing
GAF	Geographic Adaptive Fidelity
GF	Geographic Forwarding
GID	Gateway ID
GPS	Global Positioning System
HCC	Highest Connectivity Clustering
HCCM	Modified Version of HCC
HEAR	Hybrid Energy-Aware Routing
HYB	Hybrid Routing
IARP	Intra-zone Routing Protocol
IDLBC	ID Load Balancing Clustering
IERP	Inter-zone Routing Protocol
ISPs	Internet Service Providers
LBC	Load-Balancing Clustering
LCC	Least Cluster Change
LEAR	Localized Energy Aware Routing

List of Abbreviations

LIC	Lowest ID Clustering
LMC	Low-Maintenance Clustering
LRM	Local Replacement Mechanism
LS	Link State
MAC	Mobility-Aware Clustering
MANETs	Mobile Ad Hoc Networks
MTPR	Minimum Total Transmission Power Routing
NCID	Neighbor CID
NT	Neighbor Table
ODWCA	On-Demand Weighted Clustering Algorithm
OMM	Online Max-Min Routing
PC	Passive Clustering
PRC	Primary Route Cache
QoS	Quality of Service
RC	Route Cache
RCC	Random Competition-based Clustering
RCHE	Route_Cache message
RENG	Route Energy
RERR	Route Error
RREP	Route Reply
RREQ	Route Request
SRC	Secondary Route Cache
SNR	Signal-to-Noise Ratio

List of Abbreviations

TCID	Traversed/Traversing CID
TCL	Traveled Cluster List
TDOR	Time Delay On-demand Routing
WCDS	Weakly Connected Dominating Set
ZRP	Zone Routing Protocol

Table of Contents

Acknowledgements..... iii

Abstract iv

List of Abbreviations vi

Table of Contents x

List of Figures xiv

List of Tables xvii

Chapter 1 Introduction..... 1

 1.1 Background 1

 1.2 Motivation and Objectives2

 1.3 Major Contributions of the Thesis4

 1.4 Organization of the Thesis7

Chapter 2 Literature Survey of Clustering Techniques in MANETs 9

 2.1 Fundamental Knowledge of MANET Clustering9

 2.1.1 What is clustering?..... 10

 2.1.2 Why do MANETs require clustering? 10

 2.1.3 What is the cost of clustering? 11

 2.1.4 How to classify clustering schemes? 13

 2.2 Clustering Scheme Study 15

 2.2.1 DS-based Clustering (DSC) 15

 2.2.2 Low-maintenance Clustering (LMC)21

 2.2.3 Mobility-aware Clustering (MAC)28

Table of Contents

2.2.4	Energy-efficient Clustering (EEC)	32
2.2.5	Load-balancing Clustering (LBC)	36
2.2.6	Combined-metrics-based Clustering (CMC)	41
2.3	Conclusions.....	43
Chapter 3	Literature Survey of Routing Protocols in MANETs	46
3.1	Classification of MANET Routing Protocols	46
3.2	Flat Routing Vs. Hierarchical Routing	47
3.2.1	Flat Routing	47
3.2.2	Hierarchical Routing	53
3.3	Energy-Aware Routing	57
3.3.1	Transmission Power Control Protocols	59
3.3.2	Load Distribution Protocols.....	62
3.3.3	Sleep/Power-down Mode Protocols.....	65
3.4	Conclusions.....	68
Chapter 4	An Efficient Clustering Scheme for Large and Dense MANETs.....	71
4.1	Introduction.....	71
4.2	Efficient Clustering Scheme (ECS)	75
4.2.1	Concepts Clarification	75
4.2.2	Cluster Formation	77
4.2.3	Cluster Maintenance	78
4.2.4	Summary of the Newly Proposed Mechanisms in ECS	84
4.3	Simulation Models and Performance Metrics	86
4.3.1	Simulation Models.....	86
4.3.2	Performance Metrics.....	88

Table of Contents

4.3.3	Modifications of HCC.....	89
4.4	Simulation Results and Discussions	90
4.4.1	Effects of Node Mobility	90
4.4.2	Effects of Network Size	98
4.4.3	Effects of Hello Frequency	102
4.5	Conclusions.....	107
Chapter 5 Implementation and Performance Evaluation of CBRP Based on ECS in MANETs109		
5.1	Introduction.....	109
5.2	Implementation of ECBRP	112
5.2.1	Neighbor Information Discovery and Maintenance	113
5.2.2	Route Discovery.....	115
5.2.3	Route Maintenance	121
5.3	Simulation Models and Performance Metrics.....	126
5.3.1	Simulation Models	126
5.3.2	Performance Metrics	128
5.4	Simulation Results and Discussions	128
5.4.1	Effect of Traffic Load	129
5.4.2	Effect of Node Mobility.....	136
5.4.3	Effect of Network Size.....	139
5.4.4	Effect of Hello Frequency on ECBRP	142
5.5	Conclusions.....	149
Chapter 6 HEAR: Hybrid Energy-Aware Routing for Energy-Limited MANETs..... 151		
6.1	Introduction.....	151

Table of Contents

6.2	Hybrid Energy Aware Routing (HEAR).....	155
6.3	Simulation Models and Performance Metrics	163
6.3.1	Simulation Models.....	163
6.3.2	Performance Metrics.....	164
6.4	Simulation Results and Discussions	165
6.4.1	Network Lifetime Performance	165
6.4.2	Network Throughput Performance	168
6.4.3	Routing Overheads Performance	174
6.5	Conclusions.....	177
Chapter 7	Conclusions and Future Work	179
7.1	Conclusions.....	179
7.2	Recommendations for Future Work	182
	Author's Publications	186
	Bibliography	188

List of Figures

Figure 2.1 Cluster Structure Illustration. 10

Figure 2.2 Dominating Set Illustration. 15

Figure 2.3 WCDS Illustration. 18

Figure 2.4 Clustering Mechanisms of PC. 25

Figure 2.5 Flowchart of DDCA. 30

Figure 2.6 Illustration of Cluster Merging in AMC. 38

Figure 2.7 Flowchart of Clustering Formation Procedure in ODWCA. 42

Figure 3.1 Illustration of Route Discovery in DSR. 51

Figure 3.2 Illustration of Route Discovery in ZRP. 56

Figure 3.3 Min-Power Path and Max-Min Path in OMM. 62

Figure 3.4 Illustration of RCHE Message Utilization in LEAR. 65

Figure 3.5 Virtual Grid Structure in GAF. 67

Figure 3.6 Illustration of Master Election in GAF. 68

Figure 4.1 Illustration of Cluster Structure after Cluster Formation. 78

Figure 4.2 Illustration of Cluster Structure in Cluster Maintenance. 81

Figure 4.3 T_c vs. v_{max} with $N = 500$ 91

Figure 4.4 T_m vs. v_{max} with $N = 500$ 92

Figure 4.5 N_c and S_c vs. v_{max} with $N = 500$ 94

Figure 4.6 Clustering Status Distribution of HCCM vs. Simulation Time with $v_{max} =$
10m/s and $N = 500$ 95

List of Figures

Figure 4.7 Clustering Status Distribution of ECS vs. Simulation Time with $v_{max} = 10\text{m/s}$ and $N = 500$	95
Figure 4.8 O_c-H vs. v_{max} with $N = 500$	96
Figure 4.9 O_c vs. v_{max} with $N = 500$	98
Figure 4.10 T_c vs. N with $v_{max} = 10\text{m/s}$	99
Figure 4.11 T_m vs. N with $v_{max} = 10\text{m/s}$	100
Figure 4.12 N_c and S_c vs. N with $v_{max} = 10\text{m/s}$	100
Figure 4.13 O_c-H vs. N with $v_{max} = 10\text{m/s}$	101
Figure 4.14 O_c vs. N with $v_{max} = 10\text{m/s}$	102
Figure 4.15 T_c Vs. h_B with $N=500$	103
Figure 4.16 T_m Vs. h_B with $N=500$	104
Figure 4.17 O_c-H vs. h_B with $N=500$	105
Figure 4.18 Clustering Status Distribution with $v_{max} = 10\text{m/s}$ and $N = 500$	107
Figure 5.1 Cluster Based MANET Topology.....	113
Figure 5.2 Illustration of the RREP mechanism.....	121
Figure 5.3 Illustration of the Local Route Repair Mechanism.....	124
Figure 5.4 r_p vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$	130
Figure 5.5: NO_v-H vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$	133
Figure 5.6: NO_v vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$	134
Figure 5.7 d_t vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$	135
Figure 5.8 r_p vs. v_{max} with $n_t=15$ and $N=50$	137
Figure 5.9 NO_v vs. v_{max} with $n_t=15$ and $N=50$	138
Figure 5.10 d_t vs. v_{max} with $n_t=15$ and $N=50$	139
Figure 5.11 r_p vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$	140

List of Figures

Figure 5.12 NO_v vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$.	141
Figure 5.13 d_t vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$.	142
Figure 5.14 Number of RERRs vs. v_{max} with $n_t=15$.	144
Figure 5.15 Number of RERRs vs. v_{max} with $n_t=20$.	145
Figure 5.16 r_p vs. v_{max} with $n_t=15$.	146
Figure 5.17 r_p vs. v_{max} with $n_t=20$.	146
Figure 5.18 d_t vs. v_{max} with $n_t=15$.	148
Figure 5.19 d_t vs. v_{max} with $n_t=20$.	149
Figure 6.1 MANET Topology.	157
Figure 6.2 Illustration of LRM.	162
Figure 6.3 T_L vs. v_{max} with $n_t=15$ and $r_s=4$.	166
Figure 6.4 n_L vs. Simulation Time with $v_{max}=10\text{m/s}$ and $n_t=15$.	167
Figure 6.5 r_p vs. v_{max} with $n_t=15$ and $r_s=4$.	169
Figure 6.6 N_p vs. v_{max} with $n_t=15$ and $r_s=4$.	170
Figure 6.7 r_p vs. r_s with $v_{max}=10\text{m/s}$ and $n_t=15$.	171
Figure 6.8 N_p vs. r_s with $v_{max}=10\text{m/s}$ and $n_t=15$.	171
Figure 6.9 r_p vs. n_t with $v_{max}=10\text{m/s}$ and $r_s=4$.	173
Figure 6.10 N_p vs. n_t with $v_{max}=10\text{m/s}$ and $r_s=4$.	173
Figure 6.11 O_v vs. v_{max} with $n_t=15$ and $r_s=4$.	174
Figure 6.12 O_v vs. n_t with $v_{max}=10\text{m/s}$ and $r_s=4$.	175
Figure 6.13 Distribution of Routing Overheads with $v_{max}=10\text{m/s}$, $n_t=15$ and $r_s=4$.	177

List of Tables

Table 2.1: Description of Cost Terms for Cluster Based MANETs. 12

Table 2.2: Summary of Six Clustering Categories. 14

Table 2.3: Summary of DSC Schemes. 20

Table 2.4: Cost Comparison of DSC Schemes. 20

Table 2.5: Distributed Cluster Formation Scheme in Lin’s Algorithm. 23

Table 2.6: Summary of LMC Schemes. 26

Table 2.7: Cost Comparison of LMC Schemes. 26

Table 2.8: Summary of MAC schemes..... 31

Table 2.9: Cost Comparison of MAC schemes. 31

Table 2.10: Summary of EEC Schemes..... 35

Table 2.11: Cost Comparison of EEC Schemes. 35

Table 2.12: Summary of LBC Schemes. 39

Table 2.13: Cost Comparison of LBC Schemes. 40

Table 2.14: Summary of ODWCA Scheme..... 43

Table 2.15: Cost of ODWCA Scheme..... 43

Table 3.1: Category of Energy-Aware Routing Protocols..... 59

Table 4.1: Summary of Clustering Messages for ECS. 82

Table 4.2: Functions and Information of Clustering Messages for ECS. 83

Table 4.3: Information Table for Mobile Node n_g 84

Table 4.4: Default Simulation Parameters. 87

Table 4.5 O_c vs. h_B with $N=500$ for RCC and ECS. 106

List of Tables

Table 5.1: 1-hop NT of Node n_1	114
Table 5.2: 2-hop NT of Node n_1	114
Table 5.3: CAT of Node n_4	115
Table 5.4: Partial RREQ Header Format.	117
Table 5.5: Partial RREP Header Format.....	119
Table 5.6: Default Simulation Parameters.	126
Table 5.7: Number of Dropped Packets at IP Level Due to Queue Size Full of Mobile Nodes.	130
Table 5.8: Number of Triggered Hello Messages.....	137
Table 6.1: NT of Node n_1	157
Table 6.2: Default Simulation Parameters.	163
Table 6.3: Average Communication Expiration Time (T_{exp}).	167

Chapter 1

Introduction

1.1 Background

In recent years, mobile computing has enjoyed a tremendous rise in popularity because of rapid technological advance and market growth. The great popularity of Internet services makes more people enjoy and depend on the networking applications. As wireless network nodes proliferate and as Internet services become more and more essential in people's daily life, people will expect to use networking applications even in situations where the Internet itself is not available [1]. Mobile Ad Hoc Networks (MANETs), without any fixed infrastructures, allow mobile terminals with compatible wireless communication devices to build up a short-lived network for communication needs of the moment. In MANETs, wireless computing devices are able to communicate with each other when no routers or base stations or Internet service providers (ISPs) can be found. In the absence of such fixed infrastructure, the wireless computing devices themselves take on the missing functions [1]. Hence, MANETs bear great application potential in scenarios, such as disaster and emergency relief, mobile conferencing, home networking, sensor dust and battle field communication [1, 2].

However, MANETs still face a lot of challenging problems that may greatly affect their performances and applications, such as scalability, energy limitation, wireless data rate limitation, security threats, and protocol standardization [1]. In this Thesis, we

mainly focus on providing solutions for two of the above addressed problems: scalability and energy limitation.

1.2 Motivation and Objectives

Routing within today's Internet depends on the ability to aggregate reachability information to IP nodes. This aggregation is based on the assignment of IP addresses to nodes so that all the nodes on the same network link share the same routing prefix. The aggregation is effective in helping to reduce the number of routing prefixes that have to be advertised across the Internet, thus enabling today's router hardware to continue to maintain the Internet's global addressability. Hence, we can say that aggregating routing information is the key to Internet scalability [1]. However, such aggregation is typically not available for MANETs because mobile nodes in a MANET are often assumed to have IP addresses that are assigned in a way that is not directly related to their current position relative to the network topology. Hence, MANETs are vulnerable to scalability problems. Node mobility makes MANETs less scalable due to constant changes in the network topology that cause additional overhead in routing. Since routes between a pair of mobile nodes change as any end node or intermediate node moves, the control messages are likely to be transmitted often to indicate connectivity change among mobile nodes. And this may cause overloads on mobile nodes and saturation on network bandwidth. It has been noticed that a MANET based on flat structure cannot perform well in the face of node population growth and node mobility increase [1, 3-5]. Therefore, as a typical self-organizing control structure, the hierarchy structure is essential to achieve guaranteed performance in a large MANET because it can enable efficient use of network resources and routing [1, 4, 6]. Cluster structure is considered as one of the most

popular hierarchy structures. Hence, building and maintaining a cluster based structure has become an important technique for conquering the scalability problem of MANETs. Besides, how to efficiently utilize the cluster structure for routing and data delivery is also a key issue for achieving satisfying performance in data communication of MANETs. Therefore, we put much effort in studying and discussing various issues related with MANET clustering, including clustering algorithms and cluster based routing schemes, in this Thesis.

In a typical MANET, mobile nodes operate on battery power and may not be recharged during operation. Although mobile computing devices are getting smaller and more efficient with rapid advances in wireless communications, advances in battery technology have not yet reached the stage where a mobile device can operate for days without recharging [7]. Hence, the energy limitation poses a severe challenge for the performance of such a MANET. Since the radio transceiver often consumes significant amount of power of a mobile computing device, and its power consumption is always related with all kinds of communication events in the network, such as dynamic routing and packet delivery, it is necessary to take power conservation into consideration when designing and implementing communication protocols in various layers [7, 8]. To maximize the lifetime of a MANET, the power consumption rate at each mobile node should be evenly distributed because the lack of mobile nodes can result in partitioning of the network and cause communication interruption between mobile nodes [9, 10]. In a MANET, besides serving as data sources and sinks, mobile nodes also take the responsibility of routers by relaying data packets for other mobile nodes. Excessively serving as data relays for others will make a mobile node overloaded and drain its battery energy quickly, and consequently “die” early because of energy depletion. Hence

properly choosing mobile nodes as data relays with enough residual energy, which is decided by the routing protocol, is important for avoiding possible network partition and guaranteeing performance for an energy-limited MANET. Finally, in this Thesis, we propose an energy-aware routing protocol for an energy-limited MANET in order to prolong its network lifetime as well as to maximize its network throughput.

Thus, the objectives of this Thesis are to improve the performance of a MANET in face of scalability and energy limitation challenge by proposing several clustering and routing protocols.

1.3 Major Contributions of the Thesis

1) Clustering, as a key technique to solve the scalability problem of a MANET, has been put much effort by researchers in recent years. A large variety of approaches for MANET clustering have been proposed, whereby different approaches may focus on different metrics. In our Thesis, we first give a survey on various existing MANET clustering schemes. This survey is the first comprehensive survey for clustering schemes in MANET research so far. In this survey, we provide the basic concept about MANET clustering, show the pros and cons of cluster based MANETs, categorize some proposed clustering schemes based on their specific objectives, analyze the clustering cost of each clustering scheme qualitatively or quantitatively, and discuss their mechanisms and feasible applications in detail. With this survey, people can have more thorough understanding about MANET clustering and up-to-date knowledge about research trends and progress in MANET clustering.

2) In a MANET with moderate to high node density, a typical 1-hop clustering scheme likely forms highly overlapped clusters. Highly overlapped cluster structure may

be inefficient in controlling routing traffic, bring difficulty in spatial reuse of wireless channels, and produce longer routes for data communications [6, 11, 12]. Besides, some existing clustering schemes may invoke ripple effect of re-clustering when the underlying network topology changes. This adversely affects the cluster stability, likely introduces large clustering control overheads and adversely affects the protocols and applications relying on the cluster structure [4, 13]. In this Thesis, we propose a clustering scheme, named efficient clustering scheme (ECS), for large and dense MANETs. With the introduction of a new cluster-related status, clusterguest (CG), and a cluster deletion mechanism, ECS can eliminate small and unnecessary clusters and hence reduce the cluster overlapping for a MANET with moderate to high node density. By limiting the re-clustering situations, reducing the cluster re-affiliation rate [14] and eliminating the ripple effect of re-clustering, ECS can provide a stable structure for upper layer applications. Simulation results show that ECS can perform well even in a MANET with a large number of highly mobile nodes in terms of cluster stability and cluster overlapping.

3). A cluster structure is to serve for a routing protocol residing on it. Hence, to judge the efficiency of a clustering scheme, it is necessary to evaluate the performance of its upper layer routing protocol. On the other hand, the upper layer routing protocol should be able to make full use of the cluster structure to improve its performance. In this Thesis, we implement the major routing mechanisms of Cluster Based Routing Protocol (CBRP) [15] on the top of our proposed clustering scheme, ECS, because CBRP is considered as a promising cluster based routing scheme and ECS can provide a more stable and less overlapping cluster structure. Some routing mechanisms of CBRP are modified accordingly so that they can better utilize the underlying cluster structure

maintained by ECS. An information table update mechanism by utilizing the routing and packet delivery events are proposed. Such a cluster based routing scheme is named efficient cluster based routing protocol (ECBRP) in this Thesis. By reducing the routing backbone size due to the less overlapping underlying cluster structure, ECBRP can effectively reduce the routing overheads in the network. By providing a more stable cluster structure because of ECS and introducing the information table update mechanism, ECBRP can provide better routing efficiency. Hence, ECBRP shows satisfying performance in terms of data delivery ratio, routing overheads and end-to-end delay in various simulation scenarios.

4). Energy limitation is a challenging issue for the performance and application of MANETs. The early “death” of partial mobile nodes due to energy depletion may cause serious problems, such as network partition and communication interruption [9], and thus greatly deteriorates the network performance. Routing behavior decides the data relays for data communication in a MANET, and consequently may affect the energy dissipation rate of a mobile node because excessively serving as data relays may consume a large amount of battery energy of a mobile node. Hence, the energy metrics should be taken into consideration in designing MANET routing protocols. In this Thesis, we propose a routing scheme, named hybrid energy-aware routing (HEAR), for an energy-limited MANET. In HEAR, a time delay mechanism is introduced in the route discovery procedure of the on-demand routing component to find energy-rich paths between communication pairs. Energy-based adaptive Hello messages are exchanged among direct neighbors to update the neighbor table (NT) and reduce the side effect of Hello on draining mobile nodes’ battery energy. And a local replacement mechanism (LRM) is introduced to dynamically replace energy-poor nodes along active routes.

HEAR avoids over-utilization of mobile nodes as data relays based on their energy levels to minimize network disconnection and service interruption due to early energy-depletion of partial mobile nodes, and thus improves network lifetime and network throughput for an energy-limited MANET.

1.4 Organization of the Thesis

This Thesis is organized as follows:

In Chapter 2, the comprehensive survey about clustering schemes in MANET research is given. Various issues and comparison metrics related with MANET clustering are discussed and studied.

In Chapter 3, a review of some routing protocols in MANETs is presented. Several typical cluster based routing schemes and energy-aware routing schemes are included.

In Chapter 4, mechanisms for cluster formation and maintenance in ECS are described in detail. The performance of ECS is studied and compared with random competition-based clustering (RCC) [5] and a modified version of highest connectivity clustering (HCC) [16] in terms of clustering overheads, clusterhead (CH) lifetime, cluster number and cluster size. Simulation results show that ECS successfully achieves its objectives at reducing the cluster overlapping, maintaining a stable cluster structure as well as producing moderate clustering overheads.

In Chapter 5, mechanisms for the route discovery and route maintenance in ECBRP are described and studied in detail. Then its performance is compared with dynamic source routing (DSR) [17] and CBRP in terms of packet delivery ratio, routing overheads, end-to-end transmission delay and so on. Simulation results show that

ECBRP can outperform the other two schemes as the network traffic load, node mobility or network size increases.

In Chapter 6, mechanisms for the route discovery and route maintenance in HEAR are described and studied in detail. The performance of HEAR is compared with DSR, time-delay on-demand routing (TDOR) [18], hybrid routing (HYB) in terms of packet delivery ratio, number of data packet received, routing overheads, average lifetime of mobile nodes and average lifetime of communication pairs. Simulation results show that HEAR can outperform the other three schemes in network throughput and network lifetime in such an energy-limited MANET.

In Chapter 7, the final conclusions about this Thesis and some recommendations for future work are provided.

Chapter 2 Literature Survey of Clustering Techniques in MANETs

2.1 Fundamental Knowledge of MANET Clustering

Dynamic routing is an important issue in MANETs, and it has been proved that a flat structure exclusively based on proactive or reactive routing schemes cannot perform well in a large dynamic MANET [3-5, 19]. In other words, a flat structure encounters scalability problem with increased network size, in particular in face of node mobility. The communication overhead of link-based proactive routing protocols increases with the square of the number of mobile nodes in a MANET. For reactive routing schemes, the disturbing route request (RREQ) flooding and the considerable route setup delay become intolerable as network size and mobility increase. Hence, hierarchical architecture is essential for achieving performance guarantee in a large-scale MANET [1]. Since cluster structure is a typical hierarchy, many papers focus on presenting effective and efficient clustering schemes for MANETs. In this Chapter we give a comprehensive survey on clustering schemes for MANET research, and classify and study these schemes based on their objectives.

2.1.1 What is clustering?

Clustering is to divide the mobile nodes in a MANET into virtual groups, and to allocate mobile nodes geographically adjacent into the same cluster according to some rules. A typical cluster structure is shown in Figure 2.1. In a cluster, mobile nodes may be assigned different status or function, such as CH, clustergateway (CGW) or clustermember (CM). A CH normally serves as a local coordinator for its cluster, in charge of intra-cluster transmission arrangement and inter-cluster routing, and its ID is usually utilized to identify the corresponding cluster. A CGW is a member node with inter-cluster links, so that it can connect to neighboring clusters and forward information between clusters. A CM is a member node without inter-cluster links.

Figure 2.1 Cluster Structure Illustration.

2.1.2 Why do MANETs require clustering?

A cluster structure, as an effective topology control means [20], provides at least three benefits [21-23]. Firstly, a cluster structure facilitates the spatial reuse of wireless band resources to increase the system capacity [22, 24]. With the non-overlapping multi-cluster structure, two clusters may deploy the same frequency channel or code set if they

are not neighboring clusters [23]. The second benefit is on routing because CHs and CGWs can form a virtual backbone for inter-cluster routing, and only those backbone nodes are involved in routing. Hence, the routing space and the broadcast routing traffic are effectively reduced [25, 26]. Lastly, a cluster structure makes a MANET appear smaller and more stable in the view of mobile nodes because a mobile node only records information about its neighbors and cluster(s) [27, 28]. Thus, information processed and stored by each mobile node is greatly reduced.

2.1.3 What is the cost of clustering?

A cluster based MANET has its side effects because constructing and maintaining a cluster structure usually needs additional cost. The clustering cost is a key issue to validate the effectiveness and scalability enhancement of a cluster structure. By analyzing the cost of a clustering scheme in different aspects qualitatively or quantitatively, its advantages and drawbacks can be specified. The clustering cost terms are described as follows.

- To maintain a cluster structure in a dynamically changing scenario often requires explicit clustering message exchange among mobile nodes [11-14, 16, 22, 29-35], which may consume considerable bandwidth and drain mobile nodes' energy quickly so that the clustering events will compete for the limited wireless band resources with upper layer applications, such as routing and data delivery.
- Ripple effect of re-clustering indicates the cluster structure rebuilt over the whole or partial network due to some local events, such as the movement of a mobile node or the re-election of a CH [11, 13, 14, 16, 29, 31, 33, 34, 36]. Hence, the ripple effect

may greatly affect the cluster structure stability [13] and the performance of upper layer protocols, which may be adversely affected when the cluster topology changes.

- Some schemes assume mobile nodes to be kept static in the initial cluster formation [11, 12, 16, 23-25, 27-30, 37]. Because those schemes require that the initially elected CHs are with specific attributes in their respective neighborhoods, and this can only be guaranteed with information exchange in the neighborhood with unchanged topology. However, this assumption may not be applicable in an actual scenario [36, 38], where mobile nodes may move randomly all the time.
- The computation round is the number of rounds that a cluster formation procedure can be completed and reflects how fast a cluster structure can be deployed in a MANET. In some clustering schemes, the cluster-related status for a mobile node cannot be determined by itself alone but may depend on neighbors' status, and thus the time required for the cluster construction for those schemes normally cannot be bounded.

The explicit clustering message, the ripple effect of re-clustering and the frozen period for cluster formation are the additional costs of a cluster based MANET compared with a flat MANET. In this Chapter, we will discuss the cost of a clustering scheme in the aspects as shown in Table 2.1.

Table 2.1: Description of Cost Terms for Cluster Based MANETs.

Cost of Clustering	Definition and Description
Explicit clustering message	Clustering requires explicit clustering-related message exchange among mobile nodes.
Ripple effect of re-clustering	The cluster structure is rebuilt over the whole network due to some local events, such as the movement of a mobile node or the re-election of a CH.

Frozen period for cluster formation	Mobile nodes must be static in the cluster formation phase so that mobile nodes can obtain accurate neighbor information and the cluster structure can be guaranteed with specific attributes.
Constant Computation round	Computation round is the number of rounds that a cluster formation procedure can be completed. The non-constant computation round of a clustering scheme indicates its unbounded time complexity.
Communication (message) complexity	Communication complexity indicates the total number of clustering-related messages exchanged for the cluster formation of a clustering scheme or re-clustering in the cluster maintenance of a clustering scheme with ripple effect.

2.1.4 How to classify clustering schemes?

Clustering schemes of MANETs can be classified based on different criteria. Depending on whether a special mobile node with extra functions, such as a CH, is required for a cluster, clustering protocols can be classified as CH-based clustering [11-14, 16, 29-35] and non-CH-based clustering [21, 22]. Based on the hop distance between node pairs in a cluster, clustering schemes can be divided into 1-hop clustering [11-14, 16, 29-31, 34] and multi-hop clustering [32, 35].

In this Chapter, we classify the clustering protocols based on their objectives. According to this criterion, the discussed schemes in this Chapter can be grouped into six categories as shown in Table 2.2. Dominating-Set-based (DS-based) clustering (DSC) [11, 12, 39, 40] tries to find a DS to form the routing backbone for a MANET so that the number of mobile nodes participating in routing is reduced. Low-maintenance clustering (LMC) schemes [13, 22, 30, 36] aim at providing stable cluster architecture with low cost by limiting re-clustering situations or reducing explicit clustering control messages. Mobility-aware clustering (MAC) [21, 31, 32] groups mobile nodes with low relative speed into a cluster, and thus the intra-cluster links can be tightened and the cluster

structure can be stabilized accordingly. Energy-efficient clustering (EEC) [33, 34] eliminates unnecessary energy consumption or balances energy consumption for the clustering events of mobile nodes to avoid the early energy depletion of partial mobile nodes. Load-balancing clustering (LBC) [33, 35] attempts to limit the number of mobile nodes in a cluster in a preset range so that clusters are with similar size and the network loads can be more evenly distributed in each cluster. Combined-metrics-based clustering (CMC) [14] considers multiple metrics in cluster construction and CH decision. With the consideration of several parameters, CHs can be more properly chosen without giving bias to mobile nodes with specific attributes. Based on this classification, we can more easily study the common criteria underlying each category, and the similarity and difference for schemes in the same category. At the same time, we can find out the feasible application scenario for each clustering category. A brief summary of these categories is given in Table 2.2.

Table 2.2: Summary of Six Clustering Categories.

Clustering Category	Objectives
DSC	Finding a DS to reduce the number of mobile nodes participating in routing;
LMC	Providing a stable cluster infrastructure with low clustering maintenance cost;
MAC	Utilizing mobile nodes' mobility behavior for clustering and assigning mobile nodes with low relative speed to the same cluster to stabilize the inter-cluster links;
EEC	Avoiding unnecessary energy consumption or balancing energy consumption for clustering events of mobile nodes to prevent early energy depletion of mobile nodes;
LBC	Distributing the network load more evenly into clusters by limiting the number of mobile nodes in each cluster in a preset range;
CMC	Considering multiple metrics in cluster configuration and CH selection.

2.2 Clustering Scheme Study

In this Section, typical clustering schemes of MANETs are classified into six categories as shown in Table 2.2 and studied based on their objectives.

2.2.1 DS-based Clustering (DSC)

Routing based on a set of dominating nodes [11, 39, 40], which function as the CHs for a group of mobile nodes, is a typical technique in MANETs. Such a set of nodes is called a DS. The idea of finding a DS for a MANET comes from that the routing process is only aggregated on mobile nodes in the DS [11]. Hence, when table-driven routing is applied, only nodes in the DS are required to maintain the routing tables. When on-demand routing is adopted, the route search space is mainly limited to the DS.

Figure 2.2 Dominating Set Illustration.

Taking a MANET as an un-weighted graph G , a vertex (node) subset S_G of G is a DS if each vertex in G either belongs to S_G or is adjacent to at least one vertex in S_G . For example, in Figure 2.2(a), the black vertices are dominating nodes and they form a DS. A DS is called a connected DS (CDS) if all the dominating nodes are directly connected. In

Figure 2.2(b), the black vertices form a CDS and the black dash lines indicate the backbone that can be utilized for routing.

2.2.1.1 Wu's CDS Algorithm

Wu [11] proposed a distributed algorithm to find a CDS in order to design efficient routing schemes for a MANET. Initially, every mobile node v exchanges its neighbor list with all its neighbors. A mobile node sets itself as a dominating node if it has at least two unconnected neighbors. This is called a marking process. Then some extension rules are implemented to reduce the size of a CDS generated from the marking process. A mobile node deletes itself from the CDS when its close (open) neighbor set is completely included in the neighbor set(s) of a (two connected) neighboring dominating node(s) and it has smaller ID than the neighboring dominating node(s). The open neighbor set of a mobile node only includes a mobile node's direct neighbors, whereas the close neighbor set is the open neighbor set plus itself. A CDS with smaller size reduces the number of nodes involving in routing task. The extension rules proposed in Wu's algorithm are effective for reducing the DS size [11]. Wu's algorithm guarantees the cluster construction completed in just two rounds, one for marking procedure and the other for extension rules, so that its time complexity can be bounded.

The connection relationship of mobile nodes may change and the CDS needs to be updated if any one or more of these events occur, including switch-on, switch-off and movement of mobile nodes. For the CDS maintenance, any moving mobile node needs to send beacon message in every τ seconds during its movement. Other related mobile nodes keep monitoring the message from the moving node. Therefore, a single mobile node's movement may suppress many mobile nodes from transmitting or receiving their

own packets. Also, if many mobile nodes in the network are in movement, the network topology may be greatly affected and thus, the complete re-calculation of a CDS with large amount of message exchange is required. Hence, the cluster maintenance events in this algorithm are expensive in terms of transmission efficiency and control overhead. Since no simulation performances for moving scenarios are provided, the effectiveness of the cluster maintenance mechanisms of Wu's scheme under a dynamic environment is with doubt.

2.2.1.2 Chen's Weakly CDS (WCDS) Algorithm

Chen [12] pointed out that although a CDS provides explicit information for inter-cluster routing, the number of clusters produced by a CDS clustering is rather large. Hence, Chen proposed schemes to build a weakly CDS (WCDS) by relaxing the requirement of direct connection between neighboring dominating nodes so that a DS with smaller size can be formed. The induced subgraph of a WCDS includes the dominating nodes, those non-dominating nodes, each of which can link two unconnected dominating nodes together just by itself, and the links between these nodes. In Figure 2.3(a), the black vertices form a WCDS and the black dash lines indicate paths that can be used for inter-cluster routing. The size of the DS is decreased from 8 to 5 as compared to Figure 2.2(b).

Figure 2.3 WCDS Illustration.

Five algorithms have been proposed in [12]. We only discuss algorithm I and V because of space limitation. In both algorithms, each vertex in a graph G is associated with a color (white, gray, or black) and a piece. Initially all vertices are white-colored. When a vertex is colored black, all its direct white neighbors are changed to gray. Piece is a term to describe a substructure of a graph. A white piece is a single white-colored vertex. A black piece includes a black vertex plus all its direct gray-colored neighbors. If two black vertices can be connected through a single gray vertex, their corresponding black pieces can be merged into a larger one. Hence, a black piece refers to a maximal set of black vertices, which can form a WCDS, plus all their gray-colored neighbors. The idea of both algorithms is to elect the optimal white/gray vertex (the vertex with maximum improvement) to dye black, and gradually grow the vertices of a graph into a single black piece. Finally the black vertices form a WCDS. Here, the improvement of a gray/white vertex indicates the number of pieces that would be merged into a single black piece if that vertex is colored black.

In algorithm I, each round a gray/white vertex with the maximum improvement over the whole network is chosen to change to black until all vertices are included into a single black piece. For example, Figure 2.3(b) shows the piece structure of a graph after

the 3rd iteration of coloring. Then we can see that dyeing vertex 4 to black would merge 4 pieces together, whereas dyeing other vertices, 5, 6, or 7, would merge only 3 pieces. Thus, vertex 4 is selected to dye black in the 4th iteration. Later, Chen proposed a fully distributed approach, algorithm V for the WCDS construction. A mobile node can be considered as a candidate of its piece for dyeing black if it has the maximum improvement in its closed neighborhood. Therefore, a black piece may have more than one candidate at a time. In each run, by choosing the best candidate in each piece to color black, the dyeing procedure can be performed in parallel in the network. The algorithm ends when a new black piece finds that its best candidate has an improvement value of 1.

The DS size in algorithm I and V is reduced as compared to a typical CDS. However, how to maintain the WCDS structure as the network topology changes is not addressed in [12]. Algorithm I only allows one mobile node to be colored black in each run. This may cause slow convergence on WCDS construction. Algorithm V brings faster WCDS formation because it allows the dyeing procedure to perform in parallel. Both algorithms require comparatively large communication overheads because each grey/white vertex needs to exchange “improvement” information globally or in its piece with growing size after each round.

2.2.1.3 Summary of DSC

Features and objectives of the two DSC schemes are summarized in Table 2.3. Both schemes form 1-hop clusters. Compared with Wu’s algorithm, Chen’s algorithm can form less number of clusters resulting in less overlapping cluster architecture. Hence, Chen’s algorithm is more effective in simplifying a network structure for routing.

Table 2.3: Summary of DSC Schemes.

	With CH?	1-hop or multi-hop ¹	Objectives
Wu [11]	Yes	1-hop	Constructing a CDS and reducing the DS size without breaking the direct connection between neighboring dominating nodes;
Chen [12]	Yes	1-hop	Constructing a WCDS and reducing the DS size by relaxing the direct connection requirement between dominating nodes.

Table 2.4: Cost Comparison of DSC Schemes.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation round	Communication complexity
Wu [11]	Yes	Yes	Yes	Two Rounds	$O(2N)$
Chen [12]	Yes	N. A.	Yes	No	$>O(2N)$

The cost comparison of the two algorithms is shown in Table 2.4. The symbol N represents the number of mobile nodes in a network. When calculating the communication complexity of a clustering algorithm, the assumption is that each message is a local broadcast [39] and can be received by all direct neighbors of the sending node. Both Wu’s and Chen’s schemes require explicit message exchange for clustering, and have high restriction on the DS structure (connected/weakly connected). Hence, local network topology updates may require global adjustment of the DS structure, and hence bring large communication overhead for the maintenance. Thus, the DSC is more feasible for networks with low mobility. The communication complexity of

¹ Only CH-based clustering schemes are associated with the attribute of ‘1-hop’ or ‘multi-hop’ in this Chapter. The 1-hop clustering indicates that the hop distance between a CH and its member nodes is limited to 1. The multi-hop clustering indicates that there is no hop limit between a CH and its member nodes.

Wu's algorithm is $O(2N)$ if only considering the marking procedure. In this marking procedure each mobile node needs to send out only two messages, one for its neighbor list broadcast and the other for its cluster-related status claim. Wu's algorithm has lower computation and communication complexity than Chen's because Wu's algorithm could be completed in just two rounds but Chen's needs non-constant and probably larger number of rounds.

2.2.2 Low-maintenance Clustering (LMC)

The main criticism of a clustered MANET comes from the need of extra explicit message exchange among mobile nodes for maintaining the cluster structure. When network topology changes frequently resulting in frequent cluster structure change, the clustering control overheads increase drastically. Thus, clustering behavior may consume network bandwidth excessively, drain mobile nodes' energy quickly, and override its improvement on network scalability and performance [36]. Hence, it is important to reduce the clustering overheads. Most LMC protocols aim at providing stable cluster architecture by reducing re-affiliation rate and minimizing re-clustering situations, and thus the clustering control overheads can be reduced accordingly. Here, re-affiliation refers to a non-CH node changing its attached cluster. Some schemes eliminate the clustering control overheads completely by building and maintaining cluster architecture as a byproduct of data delivery [36], which means that the cluster status for mobile nodes can be refreshed in packet delivery procedure.

2.2.2.1 Least Cluster Change (LCC)

LCC [13] is considered as large enhancement of Lowest ID Clustering (LIC) [29] or HCC [16]. Before LCC is proposed, most clustering protocols execute the clustering procedure periodically, and re-cluster from time to time in order to satisfy some specific attribute of a CH in its neighborhood. For example, in HCC, the clustering scheme is performed periodically to check the “local highest node degree” attribute of a CH. When a CH finds a member node with higher degree, it is forced to relinquish its CH role. This mechanism, of course, involves frequent re-clustering.

In LCC the clustering algorithm is divided into two steps: cluster formation and cluster maintenance. The cluster formation simply follows LIC that initially mobile nodes with the lowest ID in their neighborhoods are chosen as CHs. Re-clustering is event-driven and invoked only in two cases: (1) when two CHs move into the reach range of each other, one gives up the CH role, or (2) when a mobile node cannot access any CH, it re-builds the cluster structure for the network based on lowest ID metrics. Hence, LCC significantly improves the cluster stability by relinquishing the requirement that a CH should always bear some specific attributes in its local area. But the second situation of re-clustering in LCC indicates that a single node’s movement may still incur the complete cluster structure re-computation, and once this happens, the large communication overhead for clustering may be invoked.

2.2.2.2 Lin’s Algorithm

Lin [22] proposed an adaptive clustering scheme to form non-overlapping cluster architecture without CHs because CHs bear extra work as compared to member nodes and likely become the bottlenecks of a network.

In Lin's scheme, every mobile node i keeps its own ID and the ID of its direct neighbors in a set Γ_i . The cluster ID (CID) of a cluster is the ID of its corresponding CH. Initially, mobile nodes with the lowest IDs in their local area become CHs. A mobile node i can broadcast its CID information, including its ID and CID, only when it is with the lowest ID in its set Γ_i . When a mobile node i receives CID information from a neighbor j , it deletes j from its set Γ_i . If the CID information from j is a CH claim (j 's ID and CID are the same), mobile node i checks its own CID attribute. If its CID is unspecified (it is not involved in any cluster yet) or larger than the ID of j , it sets j as its CH. The process continues till every mobile node involves in some cluster. This mechanism promises small overheads for constructing clusters because each mobile node needs to broadcast only one CID message in this procedure. The procedures of Lin's distributed clustering formation algorithm can be summarized as in Table 2.5 [22]:

Table 2.5: Distributed Cluster Formation Scheme in Lin's Algorithm.

Γ_i : my ID and IDs of my 1-hop neighbors
{
if(my_id == min(Γ_i))
{my_cid = my_id; / broadcast cluster(my_id, my_cid); / $\Gamma_i = \Gamma_i - \text{my_id}$;}
for(; ;)
{ on receiving cluster(id, cid)
{set the CID of node id to cid ;
if(id == cid and (my_cid == UNKNOWN or my_cid > cid))
my_cid = cid;
$\Gamma_i = \Gamma_i - id$;
if(my_id == min(Γ_i))
{ if(my_cid == UNKNOWN) my_cid == my_id;
broadcast cluster(my_id, my_cid); / $\Gamma_i = \Gamma_i - \text{my_id}$; }
}
if($\Gamma_i == \text{NULL}$) stop;
}
}

In Lin's algorithm, mobile nodes in a cluster are not assigned different status, like CH and CM but are required to be mutually reachable within two hops in the maintenance phase. When a mobile node i finds out that the distance between itself and some node j in the same cluster is larger than two hops, it invokes the maintenance mechanism. If i is a direct neighbor of the mobile node with highest intra-cluster connectivity in its cluster, it remains in the cluster and removes j . Otherwise i joins a neighboring cluster or forms a new cluster to cover itself in case no proper cluster to join. Lin's algorithm likely forms new clusters but has no corresponding cluster elimination or merge mechanisms, so that the number of clusters increases and more mobile nodes form single-node clusters as time advances. Eventually, the cluster structure disappears [23].

2.2.2.3 Passive Clustering (PC)

PC is a clustering protocol that does not use dedicated clustering-protocol-specific control packets or signals [36]. In PC, a mobile node can be one of the following four statuses: initial, CH, CGW and CM. All the mobile nodes are with initial status at the beginning. Only mobile nodes with initial status have the potential to be CHs. When a mobile node with initial status has something to send, such as flood search, it declares itself as a CH by piggybacking its status in the packet. Neighbors can learn the CH claim by monitoring the cluster-related status in the packet. Then, they record the CID and the packet receiving time. A mobile node that receives claim from just one CH becomes a CM, and a mobile node that hears more claims becomes a CGW. Since PC does not send any explicit clustering-related message to maintain the cluster structure, each mobile node is responsible for updating its own cluster-related status by keeping a timer. For example, when a CM does not receive any packet from its CH for a given period, its

status reverts to initial. Also, PC proposes a heuristic to control the number of CGWs in a local area without breaking its passive nature. The main idea is that a mobile node, who can hear from more than one CH, serves as a CGW only when the number difference between CGWs and CHs, $|D_{H-G}|$, in its coverage area is beyond some range; otherwise it stays with CM status. The clustering-related status change of mobile nodes running the PC scheme can be summarized in Figure 2.4.

Figure 2.4 Clustering Mechanisms of PC.

PC can form and maintain its cluster structure without explicitly exchanging the clustering control packets. Thus it can completely eliminate the control overhead caused by clustering. Since a potential CH can send out CH claims without bearing neighbors' information, PC does not require frozen period for initial neighboring learning and cluster construction as in many active clustering schemes. By adopting the gateway control heuristic, PC keeps $|D_{H-G}|$ of a local area at a constant level. This mechanism can reduce the duplicated flooding traffic and keep good connectivity between clusters. However, how to decide the optimal $|D_{H-G}|$ according to operation situations, such as traffic pattern and channel quality, is not addressed in [36].

2.2.2.4 Summary of LMC

Table 2.6 summarizes the features and objectives of the three LMC schemes. They three build 1-hop clusters with the involvement of CHs in the cluster formation phase. In the cluster maintenance phase, Lin’s algorithm does not assign different cluster-related status for mobile nodes in order to distribute traffic load more evenly among all mobile nodes, whereas the other two still keep CH-based cluster structure.

Table 2.6: Summary of LMC Schemes.

	With CH?	1-hop or Multi-hop	Objectives
LCC	Yes	1-hop	Limiting re-clustering situations and reducing clustering control overhead.
Lin [22]	Only for cluster formation	N. A.	Limiting re-clustering situations and reducing clustering control overhead; eliminating CHs in cluster maintenance to avoid overloaded mobile nodes.
PC	Yes	1-hop	Eliminating explicit control message for clustering and maintaining a cluster structure only when there are data traffics in the network; suppressing the number of CGWs to achieve flooding efficiency.

Table 2.7: Cost Comparison of LMC Schemes.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation round	Communication complexity
LCC	Yes	Yes	Yes	No	$O(mN)$
Lin [22]	Yes	No	Yes	No	$O(N)$
PC	No	Yes	No	No	Zero

Table 2.7 shows the cost comparison of the three schemes. LCC and Lin’s algorithm are active clustering because they require explicit clustering-related control

message exchange for clustering. Clustering in PC, however, is a byproduct of packet delivery and thus PC completely eliminates the clustering control overheads. LCC and Lin's algorithm requires frozen period of motion of mobile nodes for the initial cluster construction because CHs are required to be with the lowest ID in their local area in the two schemes. All the three schemes need non-constant number of rounds to complete the cluster formation because a mobile node's cluster-related status may be affected by its neighbors' status in these schemes. Hence, their time complexity is difficult to determine. The three schemes re-cluster only when the current cluster structure violates some basic maintenance rules, for example, two CHs move into the reach range of each other. Hence, the cluster structure can be stabilized and the clustering-related control overheads can be reduced by limiting the CH change situations. Lin's algorithm guarantees that the cluster formation procedure can be completed with each mobile node sending out only one message [22] so that the communication complexity of Lin's cluster formation mechanism can be kept at a low level. However, Lin's algorithm likely forms small and new clusters when topology changes and the cluster size decreases with time, and thus the cluster structure of Lin's algorithm may finally disappear. PC should be suitable for a network with high mobility because the cluster maintenance of PC is traffic-dependent and immune from increased control overhead caused by frequent change of network topology. But for a network with burst traffic, the cluster structure in PC is difficult to maintain and cannot be guaranteed ready for serving upper layer routing and data forwarding.

2.2.3 Mobility-aware Clustering (MAC)

Mobility is a prominent characteristic of MANETs, and is the main factor to bring topology change and route invalidation [31]. Some believe that it is necessary to consider mobile nodes' movement in the cluster construction in order to decrease its influence on cluster topology and form a stable cluster structure. MAC tries to group geographically near mobile nodes with low relative speed into the same cluster. Hence, the intra-cluster links become stabilized, and the re-affiliation and re-clustering rate can be decreased accordingly.

2.2.3.1 MOBIC

MOBIC [31] suggests that the cluster formation, especially the CH election, should take mobility into consideration. It also points out that the CH election is a local activity so that a CH should be determined only by its neighbors and itself. MOBIC proposes an aggregate local mobility metric for the cluster formation so that mobile nodes with relatively low speed to their neighbors have the chance to become CHs.

In MOBIC, by calculating the variance of a mobile node's relative speed to each of its neighbors, the aggregate local speed of a mobile node can be estimated. The main rationale behind calculating this variance is that a low variance indicates that this mobile node is relatively less mobile to its neighbors [31]. Consequently, mobile nodes with low variance in their neighborhoods take the CH responsibility. For the cluster maintenance, MOBIC follows that of LCC except that MOBIC avoids unnecessary CH relinquishing when CHs incidentally pass each other: only when two CHs becomes mutually reachable longer than a predefined period, one needs to give up its CH role. However, mobility behaviors of mobile nodes are not always considered in cluster maintenance in MOBIC.

So a CH is not guaranteed to bear low relative mobility characteristic to its members during maintenance phase. As time advances, the mobility criterion is somewhat ignored.

It is easy to see that MOBIC is feasible and effective for MANETs with group mobility behaviors, where a selected CH can guarantee the low mobility with respect to its member nodes. However, MOBIC may not perform well in a MANET with mobile nodes moving randomly all the time.

2.2.3.2 Distributed Dynamic Clustering Algorithm (DDCA)

DDCA [32] attempts to partition mobile nodes into multi-hop clusters based on (α, t) criteria. The (α, t) criteria indicate that a mobile node has a path to another mobile node that will be available over some time period t with a probability $\geq \alpha$ regardless of the hop distance between them. The purpose of DDCA is to support robust and efficient routing, and adaptively adjust its dominant routing scheme depending on the network mobility manner. How to adaptively choose the values for α and t can be referred to [21].

As a distributed dynamic clustering scheme, DDCA requires no periodic re-clustering. When a mobile node is powered-on, it becomes un-clustered and seeks a cluster to join by broadcasting *JoinRequest* message. A mobile node can join a cluster if it has a mutual path to satisfy (α, t) criteria between itself and the CH of that cluster. If a mobile node receives multiple *JoinResponse* messages from different clusters indicating the availability of these paths, it chooses the cluster with the highest path availability probability to join. If a mobile node does not receive any such message after a certain period of time, it creates a new cluster to cover itself. As a CH of such a single-node cluster, the mobile node monitors *JoinRequest* messages from un-clustered nodes or

other single-node clusters or periodically sends out *JoinRequest* message until it successfully expands its own cluster or joins some other cluster as a member node. The distributed clustering procedure of DDCA running at each mobile node can be summarized in Figure 2.5.

Figure 2.5 Flowchart of DDCA.

DDCA can adaptively adjust its cluster size considering the same level of stability α . In a network with low mobility, it forms large-size clusters. But in a highly mobile network, it diminishes the cluster size. DDCA suggests using table-driven schemes for intra-cluster routing whereas on-demand schemes for inter-cluster routing. Thus, a network can adaptively adjust its dominant routing mechanism according to its mobility features. As a multi-hop clustering scheme, DDCA can better tolerate mobile nodes' movement with less re-affiliation because a CH and its member nodes require no direct connection. In DDCA, a CH does not need to have specific attributes in its

neighborhood, and hence mobile nodes do not need to be stationary during cluster formation in order to get complete and accurate information of a local area.

2.2.3.3 Summary of MAC

Table 2.8 provides the summary for the two MAC schemes respectively. Both schemes are CH-based clustering, but MOBIC keeps 1-hop clusters whereas DDCA maintains multi-hop clusters. They try to group mobile nodes with low relative mobility with respect to each other into the same cluster. MOBIC uses explicit relative speed to form clusters but DDCA adopts path availability to determine mobility manners between mobile node pairs.

Table 2.8: Summary of MAC schemes.

	With CH?	1-hop or Multi-hop	Objectives
MOBIC	Yes	1-hop	Choosing a mobile node with low relative speed to its direct neighbors as a CH to stabilize the intra-cluster links.
DDCA	Yes	Multi-hop	Choosing a mobile node with low mobility to its neighbors as a CH to dynamically adjust the size of a cluster and dominant routing mechanisms.

Table 2.9: Cost Comparison of MAC schemes.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation rounds	Communication complexity
MOBIC	Yes	Yes	No	No	$O((2\Delta + 1 + m)N)$
DDCA	Yes	No	No	No	N. A.

Table 2.9 shows the cost comparison between the two schemes. In Table 2.9 Δ indicates the average number of direct (1-hop) neighbors that a mobile node has. As a multi-hop clustering, DDCA requires its CHs to keep a larger number of members' information, and the status change of a mobile node may cause table updates of more mobile nodes resulting in increased control overhead. Re-clustering of DDCA does not invoke ripple effect since there is no hop limit between two neighboring CHs. This is beneficial for keeping the stability of cluster topology and reducing the control overhead for cluster maintenance. Both schemes do not require the frozen period for the initial cluster formation because they utilize the mobility behavior of mobile nodes to decide their relative speed or path availability. In the cluster formation phase of MOBIC, firstly each mobile node sends two consecutive messages to each of its direct neighbors to help that neighbor decide the relative speed of them. Thus, 2Δ messages are necessary for each mobile node to determine the relative speed between itself and each neighbor. Then each mobile node calculates its own aggregate local mobility and broadcasts this information to its neighbors. Since MOBIC is with an overlapping cluster structure, a mobile node may broadcast more than one message (supposing m) about its cluster-related status during the cluster formation procedure. Hence, a mobile node in MOBIC is required to send out a total of $(2\Delta + 1 + m)$ messages for the initial cluster construction.

2.2.4 Energy-efficient Clustering (EEC)

Mobile nodes in a MANET normally depend on battery-power supply during operation. Hence, the energy limitation poses a severe challenge for network performance [41, 42]. In CH-based clustering, a CH bears extra work as compared to ordinary members, and it more likely "dies" early because of excessive energy

consumption. The early energy depletion of partial mobile nodes may cause network partition and communication interruption [9]. Hence, it is important to balance the energy consumption of clustering events among mobile nodes to avoid node failure for a clustering scheme, especially when some mobile nodes bear special tasks or the network density is comparatively sparse.

2.2.4.1 ID Load Balancing Clustering (IDLBC)

In IDLBC [33], each mobile node has a variable, named *VID*, and the value of *VID* is set as its ID number at first. Initially, mobile nodes with the highest IDs in their local area win the CH role. IDLBC limits the time that a mobile node can continuously serve as a CH up to *Max_Count* time units. So, when a CH exhausts its duration budget (*Max_Count*), it resets its *VID* to 0 and becomes a non-CH node. When two CHs move into the reach range, the one with higher *VID* wins the CH role. Each ordinary member increases its *VID* value by one every time unit. When a CH resigns, an ordinary member with the largest *VID* value in the neighborhood can resume the CH function.

IDLBC tries to avoid possible node failure due to early energy depletion resulting from excessively shouldering the CH role. When a mobile node resigns its CH status because of the expiration of its duration budget, another mobile node with highest *VID* in the local area is chosen to resume the CH role. A newly chosen CH in the maintenance phase is the one that its previous total CH serving time is the shortest in its neighborhood, and hence it is the most suitable one for being a CH according to IDLBC's criteria. The CH serving time alone, however, may not be able to promise a good indication on energy consumption of a mobile node.

2.2.4.2 Wu's Algorithm

Wu [34] proposed an EEC scheme based on the DS marking algorithm [11]. Mobile nodes inside a DS consume remarkable more battery energy than those outside. This is because mobile nodes inside the DS bear extra tasks, such as routing information spread and data packet relay.

Wu [34] points out that it is necessary to minimize the energy consumption of a DS in order to increase mobile nodes' lifetime. One way is to decrease the size of a DS without impairing its function, and thus unnecessary mobile nodes can be excluded from the DS and save their energy consumed for serving as CHs. In [11], Wu gave some extension rules to eliminate unnecessary dominating nodes. In [34], energy level instead of ID or node degree is used in the extension rules to determine whether a mobile node should remain in a DS. A mobile node can be deleted from a DS when its close (open) neighbor set is covered by one (two connected) dominating neighbor(s); and at the same time it has less residual energy than the dominating neighbor(s).

The proposed scheme is more energy-aware compared with other DSC algorithms because it tries to delete mobile nodes with less residual energy from the CDS when possible. However, it still cannot balance the great difference of energy consumption between dominating nodes and non-dominating nodes because the main objective in [34] is to minimize the DS updates rather than to balance the energy consumption among all mobile nodes. Hence, dominating nodes still likely deplete their energy much faster.

2.2.4.3 Summary of EEC

Table 2.10 provides some features and objectives of the two EEC schemes. They both are CH-based clustering. The difference is that Wu’s algorithm forms 1-hop clusters, whereas IDLBC does not specify its cluster size explicitly but uses 2-hop clusters for simulations [33]. Although they both attempt to consume the energy of a MANET more wisely, they have quite different focus on the energy conservation. IDLBC believes that the CHs consume more energy, and thus it proposes a scheme to balance the CH serving time among all mobile nodes. Wu’s algorithm tries to reduce the size of a DS to avoid unnecessary energy consumption of mobile nodes due to serving as dominating nodes without breaking the direction connections between neighboring dominating nodes. Since Wu’s scheme is based on CDS algorithm, it may not be suitable for a network with high mobility.

Table 2.10: Summary of EEC Schemes.

	With CH?	1-hop or Multi-hop	Specific Objectives
IDLBC	Yes	Simulation with 2-hop	Avoiding early energy depletion of mobile nodes due to excessive CH responsibility by limiting the continuous CH serving time for a mobile node;
Wu [34]	Yes	1-hop	Avoiding early energy depletion of mobile nodes by deleting energy-poor nodes from a DS without breaking the direct connections between dominating nodes.

Table 2.11: Cost Comparison of EEC Schemes.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation round	Communication complexity
IDLBC	Yes	No	Yes	No	$O([1 + (1 + \Delta)m]N)$
Wu [34]	Yes	Yes	Yes	Two rounds	$O(2N)$

Table 2.11 shows the cost comparison for the two schemes. They both are active clustering because they demand explicit clustering-related control message. They both depend on the frozen period for the cluster formation. This is because in IDLBC and Wu's algorithm, a mobile node can become a CH only after it collects some attribute information from all its neighbors and confirms that it is the one with some specific attribute in its local area. Referring to the simulation parameters, IDLBC can be considered as a 2-hop clustering, and is more complicated than Wu's energy-efficient algorithm because clustering messages are required to propagate up to two hops. Also, IDLBC requires non-constant number of rounds to finish its cluster formation. Wu's algorithm is more time efficient since it is based on Wu's CDS scheme and can be completed in two rounds. The communication complexity of Wu's EEC is $O(2N)$ if only considering the marking process.

2.2.5 Load-balancing Clustering (LBC)

LBC algorithms believe that there is an optimum number of mobile nodes that a cluster can handle, especially in a CH-based MANET. A too-large cluster may put too heavy load on the CHs and cause them become the bottleneck of a MANET, resulting in deteriorated system throughput. A too-small cluster, however, may produce a large number of clusters and increase the length of hierarchical routes resulting in longer end-to-end packet delay and inefficient packet delivery. LBC schemes set limits on the number of mobile nodes in a cluster. When a cluster size exceeds its predefined limit, re-clustering procedures are invoked to adjust the number of mobile nodes in that cluster.

2.2.5.1 Adaptive Multi-hop Clustering (AMC)

AMC [35, 43] maintains a multi-hop cluster structure and sets upper and lower bounds (U and L) on the number of member nodes in a cluster.

In AMC, each mobile node periodically broadcasts its information, including its ID, CID and cluster-related status, to others in the same cluster. By such message exchange, each mobile node obtains the topology information of its cluster. Each CGW also exchanges information with CGW neighbors in neighboring clusters periodically and reports to its CH. Then a CH can obtain the number of mobile nodes of each neighboring cluster. When the number of members, $|C_i|$, in a cluster C_i is less than L , the merge mechanism is invoked. Cluster C_i tries to find a neighboring cluster C_j to satisfy $|C_i| + |C_j| \leq U$ and maximize the sum value. If $|C_i| + |C_j| > U$ for all neighboring clusters, it tries to find a C_j to minimize the sum value. When two clusters merge, the CH with more member nodes wins to continue the CH role. Figure 2.6(a) shows the initial structure of cluster A , B and C in a MANET. Figure 2.6(b) shows that gateways f and j of cluster C exchange the member number information with neighboring clusters. Figure 2.6(c) shows that the information is forwarded to the CH of cluster C . Figure 2.6(d) displays that cluster C merges with cluster A . The CH of cluster A becomes the CH of the newly merged cluster because it has more member nodes than cluster C .

Figure 2.6 Illustration of Cluster Merging in AMC.

A cluster C_i with $|C_i| > U$ performs the division mechanism. By choosing a suitable node $v_k \in C_i$ as a new CH for the detached cluster, C_i can be separated to two clusters roughly with the same size. Nodes v_i , the original CH of C_i , and v_k send out CID update messages to member nodes of C_i . If mobile nodes receive the update message from v_i first, normally indicating they are more close to v_i , they keep their CID. Otherwise, mobile nodes change their CID to v_k . But how to select a proper node, v_k , to serve as the CH for the detached cluster, is not addressed in AMC.

AMC does not describe how the clusters are initially constructed. Although AMC mentions that the upper and lower bounds should be decided by network size and mobility, the values for U and L are given in advance in simulations without any justification.

2.2.5.2 Degree Load Balancing Clustering (DLBC)

DLBC [33] periodically runs the clustering scheme in order to keep the number of mobile nodes in each cluster around the preset value, *ED*, which indicates the optimum number of mobile nodes in a cluster. A CH degrades to an ordinary member node if the difference between *ED* and its cluster size exceeds *Max_Delta*. This mechanism tries to make all CHs serve about the same and optimal number of member nodes.

DLBC may likely cause frequent re-clustering because the movement of mobile nodes and consequent link setup/break result in frequent change of number of mobile nodes in a cluster. In addition, how to select a CH is not addressed in DLBC if no mobile nodes can satisfy the degree difference requirement between *ED* and its current node degree in a local area. Similar to AMC, how to decide the important system parameters, *ED* and *Max_Delta*, is not discussed in DLBC.

2.2.5.3 Summary of LBC

Table 2.12: Summary of LBC Schemes.

	With CH?	1-hop or Multi-hop	Specific Objectives
AMC	Yes	Multi-hop	Balancing the load and limiting the number of mobile nodes in each cluster by merging small clusters or splitting large clusters;
DLBC	Yes	Simulation with 2-hop	Balancing the load and limiting the number of mobile nodes in each cluster by choosing mobile nodes with appropriate node degree as CHs.

Table 2.12 shows the summary of the two LBC schemes. Both are CH-based clustering schemes. AMC keeps multi-hop cluster structure, whereas DLBC does not address its cluster size explicitly but uses 2-hop cluster for simulations. Both LBC schemes manage to balance the load of each cluster by keeping the number of mobile nodes in a cluster within a preset range. DLBC tries to replace a current CH with a new one when the current CH cannot satisfy the node degree requirement. But AMC try to merge a cluster with some neighboring cluster or split a cluster into two when the size of such a cluster is too small or too large.

Table 2.13 provides the cost comparison between these two schemes. AMC does not address initially how to form the multi-hop cluster structure, so it is not convenient to evaluate its cost in terms of stationary assumption, computation round and communication complexity. Since AMC forms a cluster structure with multi-hop clusters and has no limits on the hop distance between neighboring CHs, the merge or division procedure only affects the manipulated clusters. Thus, AMC brings no ripple effect of re-clustering and maintains good cluster stability. DLBC, however, may raise ripple effect of re-clustering because the node degree of its CHs is required to be within specific range all the time. Also, DLBC requires the frozen period in the cluster formation phase for each mobile node to get the node degree information of neighbors within two hops.

Table 2.13: Cost Comparison of LBC Schemes.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation round	Communication complexity
AMC	Yes	No	N. A.	N. A.	N. A.
DLBC	Yes	Yes	Yes	No	$O([1 + (1 + \Delta)m]N)$

2.2.6 Combined-metrics-based Clustering (CMC)

CMC takes a number of parameters into consideration for cluster configuration, including node degree, residual battery energy, moving speed and so on. CMC aims at electing the most suitable CH in a local area, and does not give bias to mobile nodes with certain attributes, such as lowest ID or highest node degree. Thus, it can flexibly adjust the weighting factors for each metric to cater for different scenarios. For example, in a system where battery energy is more important, the weighting factor associated with energy capacity can be set higher [14].

2.2.6.1 On-Demand Weighted Clustering Algorithm (ODWCA)

ODWCA [14] considers four parameters in the CH election procedure. They are degree-difference D_v , sum of the distance with all neighbors P_v , average moving speed, M_v , and CH serving time T_v . Here, D_v is given by $D_v = |d_v - M|$ where d_v is the number of neighbors of a mobile node v , and M is the number of mobile nodes that a CH can handle ideally. How to select M is not addressed explicitly. Besides, it is not easy to estimate P_v in a practical environment. And T_v alone cannot guarantee a good assessment of energy consumption of mobile nodes.

The combined weight I_v , is calculated as $I_v = c_1 D_v + c_2 P_v + c_3 M_v + c_4 T_v$, where c_1 , c_2 , c_3 and c_4 are the weighting factors and $\sum_{i=1}^4 c_i = 1$. All the parameter values are normalized according to some pre-defined values. ODWCA chooses mobile nodes with minimum I_v in the local area to be CHs. All mobile nodes covered by elected CHs cannot participate in further CH selection. This procedure is repeated until each mobile node is assigned to a cluster. The cluster formation procedure can be shown as Figure 2.7. Since

it is necessary for each mobile node to obtain rather large information and to calculate its combined weight for the initial CH election, the cluster formation procedure requires a longer frozen period of motion for all mobile nodes.

Figure 2.7 Flowchart of Clustering Formation Procedure in ODWCA.

In ODWCA, when a mobile node goes into a region not covered by any CH, the CH election will be performed over the whole network again to ensure that all newly elected CHs are with local minimum I_v . This kind of re-clustering raises ripple effect and requires large information exchange. Hence, the effectiveness of maintenance mechanisms of ODWCA is with doubt.

The summary and cost of ODWCA are shown in Table 2.14 and Table 2.15 respectively. ODWCA requires non-constant number of rounds to complete its cluster formation, indicating its unbounded time complexity. By sending a message to each direct neighbor, a mobile node can help each of its neighbors to decide the distance between them (by measuring the ratio of receiving power and transmission power). Supposing Δ is the average number of direct neighbors of a mobile node, each mobile

node then needs to send up to Δ messages for the distance confirmation for all its neighbors. Then, each mobile node broadcasts at least one message to claim its own information, including D_v , P_v , M_v , and T_v , to its neighbors. As ODWCA forms an overlapping cluster structure, a mobile node may broadcast more than one message about its cluster-related status. On average each mobile node sends out m such messages. Hence, then the communication complexity of ODWCA can be specified as $O((\Delta + 1 + m)N)$.

Table 2.14: Summary of ODWCA Scheme.

	With CH?	1-hop or Multi-hop	Specific Objectives
ODWCA	Yes	1-hop	Electing the most suitable CHs in local areas by considering multiple parameters and keeping a stable cluster structure by reducing re-clustering situations.

Table 2.15: Cost of ODWCA Scheme.

	Explicit clustering message	Ripple effect of re-clustering	Frozen period for cluster formation	Constant computation round	Communication complexity
ODWCA	Yes	Yes	Yes	No	$O((\Delta + 1 + m)N)$

2.3 Conclusions

In this Chapter, we provide fundamental concepts about clustering in MANETs, including the definition of clustering and a cluster, the necessity of clustering for a large dynamic MANET, and the side effects and cost of clustering. Then we classify twelve typical clustering schemes into six categories based on their main objectives, elaborate each scheme in terms of objective, mechanism, performance and feasible application, and discuss the common and difference between schemes of the same category. With this

survey we can find that a cluster based MANET has many issues to concern, such as the cluster structure stability, the clustering control overheads, and the CH election. And clustering cost needs to be considered because it is an important metric to evaluate the performance and scalability improvement of a clustering scheme.

By the detailed study of these twelve clustering schemes, we can conclude that PC can perform best in terms of communication complexity because it completely eliminates explicit clustering-related control message but attaches clustering information in other outgoing packets for building and maintaining a cluster structure in a MANET, whereas the other eleven schemes need to exchange explicit control messages for clustering.

Ripple effect of re-clustering is important to decide the communication cost of a clustering scheme in cluster maintenance phase since the ripple effect may invoke re-building of the cluster structure over the whole network. Normally multi-hop clustering schemes, such as DDCA and AMC, do not raise ripple effect of re-clustering because their cluster structure is more flexible. But most 1-hop clustering algorithms have the ripple effect. Wu's CDS scheme requires neighboring CHs to be always directly connected, and hence the re-election of one CH may break this connection and raise ripple effect of re-clustering. MOBIC and ODWCA require the cluster structure rebuilt over the whole network when a mobile node moves to a location not covered by any existing CH.

Some schemes, including Wu's CDS algorithm, Chen's WCDS algorithm, LCC, Lin's algorithm, IDLBC, Wu's EEC algorithm and DLBC, require frozen period of motion for mobile nodes when cluster formation is in progress. The reason is that in these schemes a mobile node can declare as a CH only after it exchanges some attribute

information with its neighbors and confirms that it bears some specific attribute in its neighborhood. Only when mobile nodes are stationary, the information obtained can be guaranteed correct and accurate. But this assumption may not be applicable for a real scenario, where mobile nodes may move from the very beginning of an operation. PC and DDCA, however, do not depend on this frozen period for cluster formation because their CHs do not need to be with specific attributes in their neighborhoods and mobile nodes do not exchange specific information before CH claims.

Although each scheme discussed in this Chapter can perform well in certain scenarios, no one is guaranteed to suit for all situations. We believe that this survey can help people have a more comprehensive understanding of MANETs.

Chapter 3

Literature Survey of Routing Protocols in MANETs

3.1 Classification of MANET Routing Protocols

A typical MANET is a self-organizing and self-configuring multi-hop wireless network without any fixed infrastructure, where each mobile node functions not only as a host but also as a router [1]. Thus, the routing behavior in MANET is important because the routing decides how the mobile nodes cooperate with each other for the data communication in the network. Routing in MANETs faces extreme challenge from node mobility, large number of nodes, limited communication resources (e.g., wireless bandwidth and battery energy) [1, 3]. Hence a lot of effort for MANET research has been put on proposing various routing protocols to solve different challenges for MANETs and cater for different application requirements. Also, a considerable amount of literature has addressed research on routing in MANETs [19, 44, 45]. The routing protocols can be classified and discussed based on different metrics. Based on the routing strategy, MANET routing can be categorized as proactive (table-driven) routing [27, 46-48] and reactive (on-demand) routing [17, 44, 49-51]. Based on the network structure underlying routing protocols, MANET routing can be classified as flat routing [48-51] and hierarchical routing [15, 52-56]. As the energy limitation issue in MANETs gains more

attention, energy-aware routing protocols [8, 9, 42, 57-61] for MANETs have been one of the hottest research topics over the past several years. In addition, a large number of Quality of Service (QoS) provisioning routing protocols [62, 63] and cross-layer optimization designs [63-65] have been proposed and studied lately to satisfy some applications with time and bandwidth requirements. The civil commercialization of global positioning system (GPS) makes mobile nodes equipped with GPS devices in a MANET become possible. The location information and universal timing provided by GPS equipments help to introduce directional location-based routing [66-68] to improve the performance of a MANET.

This Chapter reviews some typical flat routing schemes, hierarchical routing schemes and energy-aware routing schemes.

3.2 Flat Routing Vs. Hierarchical Routing

In MANETs, flat routing protocols adopt a flat addressing scheme, and each mobile node participating in routing plays an equal role. In contrast, hierarchical routing usually assigns different roles to mobile nodes. And some hierarchical protocols require a hierarchical addressing system [3].

3.2.1 Flat Routing

For the routing protocols based on a flat structure, they can be further divided into two categories: table-driven routing and on-demand routing. Many table-driven routing protocols are originated from conventional distance vector (DV) routing [69-71] or link state (LS) routing [71, 72]. On-demand routing, on the other hand, is a new emerging routing philosophy designed for MANETs.

3.2.1.1 Table-driven Routing

In table-driven routing, mobile nodes exchange routing information regardless of communication requests. These table-driven routing protocols have many desirable properties, especially for applications requiring real-time communication and QoS guarantees, such as low latency route access and alternative QoS path support [3].

3.2.1.1.1 Fisheye State Routing (FSR)

FSR [46] is a simple, efficient LS type routing protocol that maintains a network topology map at each mobile node and propagates LS update. The major difference between FSR and conventional LS routing protocols is the way in which routing information is spread.

First, in FSR, a mobile node exchanges the LS information only with neighbors instead of flooding it over the network. And the LS table at a mobile node is updated with the information received from its neighbors. Second, the LS exchange in FSR is not event-driven but periodic, and this can avoid frequent LS update caused by link breaks resulting from unreliable wireless links or mobile nodes' movement. The periodic broadcasts of the LS information are performed with different frequencies for different entries depending on their hop distance to the current mobile node. Entries corresponding to destinations outside a predefined scope are disseminated with lower frequency than those corresponding to destinations within the predefined scope. Hence, a large portion of entries are suppressed from frequent LS exchange packets. FSR produces accurate distance and path information about the immediate neighborhood of a mobile node, and imprecise knowledge of the best path to a distant destination. However, this imprecision

is compensated by the fact that the route on which the packet travels becomes more and more accurate as the packet approaches its destination.

3.2.1.1.2 Destination-Sequenced Distance Vector (DSDV)

DSDV [48] is a hop-by-hop DV routing protocol requiring each mobile node to periodically broadcast routing updates. The major advantage of DSDV over traditional DV routing is that DSDV guarantees loop-free routes [44, 48].

In DSDV, each mobile node maintains a routing table listing the routing entry for all available destinations. Each entry records the number of hops and the “next hop” to the corresponding destination, as well as the sequence number of the information received regarding that destination, as originally stamped by the destination. To maintain the consistency of routing tables in a dynamically changing topology, each mobile node periodically transmits routing updates or when significant new information is available. For each update from a mobile node, the *even* sequence number for itself will be increased monotonically and piggybacked in the routing update message. In DSDV, routes with larger sequence numbers are always preferred as the basis for forwarding decisions. Of the path with the same sequence number, those with the minimum number of hops will be chosen. When a mobile node, **A**, notices that its route to a destination, **D**, is broken, it advertises the route to **D** with an infinite metric and a sequence number one greater than its sequence number for that broken route. This cause any mobile nodes, **B**, to route packets through **A** to **D** to incorporate this infinite metric route into its routing table until **B** hears a route to **D** with higher sequence number. This sequence number advertising is the key mechanism to avoid route loop in DSDV.

3.2.1.2 On-demand Routing

In on-demand routing, no routing activities and no permanent routing information are maintained at mobile nodes if there is no data communication in the network [3]. The main idea behind on-demand routing is that each mobile node tries to reduce routing overheads by only sending routing packets when a mobile node has data packets to send.

On-demand routing typically have a *route discovery* phase. Route query packets are flooded to the network by the source node in search of a path to a specific destination. And this phase completes when a route is found or all possible outgoing routes from the source are searched [3]. Another important mechanism in on-demand algorithms is *route maintenance*, which handles issues related to the break of a route. When a link along an active path is broken, the upstream mobile node of the broken link initiates a route error (RERR) message to the affected source node. When the source node receives the RERR, it tries to obtain a new route to the destination if necessary.

3.2.1.2.1 Dynamic Source Routing (DSR)

DSR utilizes *source routing* that a source indicates in a data packet's header the sequence of intermediate mobile nodes on the routing path [17]. The use of source routing allows packet routing to be trivially loop free, avoids the need for maintaining up-to-date routing information in the intermediate mobile nodes, and allows mobile nodes that are forwarding or overhearing packets to record the routing information for their own future usage [3, 17, 44]. A storage space, named route cache (RC), is maintained at a mobile node to record source routes that it has learned or overheard. RC is a major mechanism in DSR to help reduce the route discovery cost.

In DSR, when a source node **S** has some data packets to send to a destination node **D**, it first checks its RC. If a route to **D** is found in its RC, it simply uses the route for data delivery. Otherwise, it will invoke the route discovery procedure. To perform a route discovery, **S** broadcasts a RREQ packet that is flooded through the network and is answered by a route reply (RREP) packet from either **D** or some mobile node that has a route to **D** recorded in its RC. Hence, it can be seen that the route discovery needs the cooperation between the source node, the destination node and many intermediate nodes, and the complete procedure can be illustrated by Figure 3.1.

Figure 3.1 Illustration of Route Discovery in DSR.

When route maintenance indicates a source route to **D** is broken, **S** is notified with a RERR packet from the upstream node of the broken link. **S** can then use any other route to **D** already in its RC or can re-invoke the route discovery in order to find a new route.

3.2.1.2.2 Ad-Hoc On-Demand Distance-Vector (AODV) Routing

AODV [51] is essentially the combination of DSR and DSDV. It borrows the basic on-demand mechanisms of route discovery and route maintenance from DSR, whereas the use of hop-by-hop routing, sequence numbers, and periodic Hello beacons from DSDV [3, 44, 51].

In AODV, when a source node **S** needs a route to some destination **D**, it first checks its DV routing table. If an entry corresponding to **D** is found, it simply forwards its data packets to the next hop as indicated in the routing table. Otherwise it broadcasts a RREQ to its neighbors, including the last known sequence number for that destination. Each mobile node that forwards the RREQ creates a *reverse route* for itself back to **S** in the DV routing table. The RREQ is flooded through the network until it reaches an intermediate mobile node **IM** that has a route to **D** recorded in its DV routing table. Then **IM** generates a RREP that contains the number of hops to reach **D** and the sequence number for **D** most recently seen by **IM**. Each mobile node that participates in forwarding this RREP back to **S**, creates a *forward route* to **D** in its DV routing table. Hence, the state created in each mobile node along the path from **S** to **D** is a hop-by-hop state; that is, each mobile node remembers only the next hop and not the entire route. In order to maintain routes, AODV requires that each mobile node periodically transmit a Hello message. Failure to receive several consecutive Hello messages from a neighbor indicates that the link to that neighbor is broken. When a link goes down, any upstream node that has recently forwarded packets to a destination using that link is notified via RERRs containing an infinite metric for that destination, indicating a route to that destination becomes unavailable. Hence, **S** will rediscover a route to **D** by initiating a RREQ message if necessary.

3.2.2 Hierarchical Routing

In MANETs, as the network size increases, the flat routing schemes likely become infeasible because of link and processing overhead [1, 3, 4]. One way to solve this scalability problem is hierarchical routing. Wireless hierarchical routing is based on the idea of organizing mobile nodes in groups and assigning mobile nodes with different functionalities inside and outside a group [3, 4]. The most popular way to build hierarchy structure is to group mobile nodes geographically near into explicit clusters based on specific rules. An alternative way is to have implicit hierarchy. In this way, each mobile node has a local scope, and normally different routing strategies are utilized inside and outside the scope.

3.2.2.1 Clusterhead-Gateway Switch Routing (CGSR)

CGSR [54] is a typical cluster based hierarchical routing scheme. In CGSR, LCC [13] is utilized as the underlying clustering scheme to partition the network into 1-hop CH-based clusters. Routing in CGSR is based on a DV routing scheme. Each mobile node maintains a cluster member table and a DV routing table. The cluster member table records the CH for each mobile node in the network and is broadcast periodically. The cluster member table of a mobile node is updated when it receives the cluster member table broadcast from others. The routing table maintains only one entry for each cluster recording the path to its corresponding CH.

To route a data packet, the source node first looks for the cluster of its corresponding destination node from its cluster member table. Then it consults its routing table to find the next hop for that destination cluster. When the destination CH receives the data packet, it forwards the packet to the corresponding destination, which is one of

its members. The path utilized for routing data packets is always with the format of “CH-CGW-CH-CGW...”. Thus, it may introduce longer routes for packet delivery and over-utilize CHs as data relays. CGSR can greatly reduce the routing table size as compared to traditional DV routing algorithms because only one entry is necessary for all mobile nodes in the same cluster. Hence, the packet size for periodic routing table broadcast is reduced accordingly. However, building and maintaining the cluster structure in a mobile scenario brings additional overheads for CGSR. Especially, the 1-hop cluster structure based on LCC in CGSR is likely highly-overlapping in a network with moderate to high node density and thus involves more mobile nodes as CHs, and that is not favor of simplifying the network topology.

3.2.2.2 Cluster Based Routing Protocol (CBRP)

CBRP is another popular cluster based routing scheme designed for MANETs. In CBRP, initially mobile nodes are grouped into 1-hop clusters based on LIC [29]. The cluster maintenance in CBRP is event-driven and similar to LCC. In CBRP, each mobile node periodically broadcasts “Hello” message to its neighbors, and thus, a mobile node can grasp the topology information within two hops and maintain such information in its NT. Each mobile node also maintains a cluster adjacency table (CAT), recording its neighboring CHs and corresponding CGWs to reach each neighboring CH.

Routing in CBRP is based on source routing and consists of two phases: route discovery and route maintenance. When a source node attempts to obtain a route to its intended destination, it broadcasts a RREQ packet. When an intermediate CH node receives such a RREQ for the first time, it attaches its own ID in a space called traveled cluster list (TCL) in the packet header and then broadcasts the RREQ. When an

intermediate CGW node receives the RREQ, it unicasts the RREQ to the corresponding CH as indicated in the packet header. Thus, when the RREQ arrives at the destination, it will include the sequence of CHs that the RREQ has traveled through. Then, a RREP message will be returned to the source using the TCL copied from the received RREQ. With the information from TCL, the RREP packet knows the necessary sequence of CHs to travel in order to reach the source. RREP utilizes the IP loose source routing² [73] to travel between neighboring clusters because a CH always knows how to reach its neighboring CHs based on the information from its NT and CAT. Finally, RREP will discover a route for subsequent data delivery.

In route maintenance, CBRP can locally adjust routes to save undeliverable packets due to link breakage along an active path based on its NT and CAT. When a route with broken links is successfully adjusted, a gratuitous RREP with the adjusted route will be returned to the source node. Thus, additional RREQs for discovering new routes can be avoided.

3.2.2.3 Zone Routing Protocol (ZRP)

ZRP [53] is a hybrid routing protocol that combines table-driven and on-demand routing strategies. In ZRP, each mobile node has a predefined zone centered at itself in terms of number of hops. For mobile nodes inside the zone, it utilizes proactive routing schemes to maintain the routing information. For mobile nodes outside the zone, it adopts on-demand routing mechanisms for discovering and maintaining the routes.

ZRP consists of three components. Within the zone, proactive intra-zone routing protocol (IARP) is adopted. IARP can be any DV or LS routing depending on the

² The IP loose source routing means that no hop-by-hop route is provided at the packet header for the corresponding packet to travel between two mobile nodes.

implementation. Outside the zone, reactive inter-zone routing protocol (IERP) is deployed. IERP utilizes RREQ/RREP packets to discover routes in a way similar to typical on-demand routing mechanisms. IARP always provides a route to mobile nodes within a mobile node's predefined zone from its routing table. When the intended destination is outside the zone area, a RREQ is broadcast via the mobile nodes on the border of the zone. Such a RREQ broadcast mechanism is called Bordercast Resolution Protocol (BRP) [53]. Routing query packets are only broadcast from one mobile node's border nodes to other border nodes until one mobile node finds out that the destination is within its zone. And the route discovery procedure in ZRP can be illustrated as in Figure 3.2.

Figure 3.2 Illustration of Route Discovery in ZRP.

The hybrid mechanisms in ZRP limit the proactive overhead to only the zone size, and the on-demand routing search overhead to only selected border mobile nodes. However, potential inefficiency may occur when the flooding of RREQ packets spreads through the whole network [3].

3.3 Energy-Aware Routing

Routing is one of the key issues in MANETs due to its highly dynamic and distributed nature. Energy-aware routing may be one of the most important design criteria because mobile nodes in a typical MANET are supplied with batteries with limited capacity. Power depletion of a mobile node not only affects the node itself but also its function as a data relay for others and thus degrades the overall network performance [10, 74]. Hence, many research efforts have been devoted to developing energy-aware routing protocols for energy-limited MANETs. Energy-aware routing indicates that energy-related metrics are taken into consideration in routing and route selection so that the network can operate longer. An energy-aware routing protocol can be a flat routing scheme or a hierarchical routing scheme based on whether a hierarchical structure is adopted for the routing.

Energy-aware routing schemes can be broadly categorized based on when the energy-related mechanisms are performed. A mobile node consumes its battery energy not only when it actively sends or receives packets but also when it stays idle listening to the wireless medium [8, 74]. Therefore, energy-aware routing protocols introduce the energy-aware mechanisms for a mobile node either when it actively transmits/receives packets or during its inactive period. Accordingly, energy-aware routing protocols can be

categorized as active communication energy consumption optimization and inactive energy consumption optimization [74].

Protocols that belong to the former category can be further divided into two types. The first type is called transmission power control approach [74]. It reduces the active communication energy consumption of mobile nodes by adjusting each mobile node's transmit power just enough to reach the receiving node by satisfying signal-to-noise ratio (SNR). The objective is to obtain the optimal route that minimizes the total transmission energy required to deliver data packets between a source-destination pair [42, 59, 74]. The second type is called load distribution approach and its main goal is to avoid over-utilizing mobile nodes serving as data relays [9, 57, 74]. Thus, the overall network performance for such an energy-limited MANET can be improved by avoiding the network partition and communication interruption caused by the early "death" of partial mobile nodes due to energy depletion [9, 10].

For protocols that belong to the latter category, each mobile node can save its inactivity energy by switching the operation mode of its radio into sleep/power-down mode or simply turning it off when there is no data traffic. This leads to considerable energy savings, especially when the network is with low data traffic load [8, 74]. However, it requires well-designed routing protocols to provide data delivery guarantee because partial mobile nodes turning into sleep mode may impair route diversity, route discovery and packet delivery. Table 3.1 shows the energy-aware routing protocols discussed in this Chapter.

Table 3.1: Category of Energy-Aware Routing Protocols.

Approach		Protocols	Objectives
Optimizing Active Communication Energy Consumption	Transmission Power Control	1) Minimum Total Transmission Power Routing (MTPR) [9] 2) Online Max-Min Routing (OMM) [59]	Minimizing the total transmission energy consumption along a route
	Load Distribution	1) Conditional Max-Min Battery Capacity Routing (CMMBCR) [9] 2) Localized Energy Aware Routing (LEAR) [61]	Distributing relay load to energy-rich nodes to avoid early energy depletion of partial mobile nodes
Optimizing Inactivity Energy Consumption	Sleep/Power-down Mode	1) SPAN [8] 2) Geographic Adaptive Fidelity Routing (GAF) [58]	Minimizing power consumption of mobile nodes during inactivity

3.3.1 Transmission Power Control Protocols

Energy-aware routing protocols based on transmission power control try to find the optimal route that minimizes the total transmission power between a source-destination pair because more power is consumed during the packet transmission than the reception or idle period. Transmission to a distant device at higher power may consume a disproportionate amount of power as compared to transmission to a mobile node in closer neighborhood [42]. Hence, transmission power control routing protocols likely involve more intermediate nodes to achieve less total transmission power consumption for a communication pair. However, lower transmission power may make the topology sparse, resulting in network partitioning and high end-to-end delay due to a large hop count [74].

3.3.1.1 Minimum Total Transmission Power Routing (MTPR)

In wireless communication, for a successful transmission, the received signal strength at a mobile node n_j should be higher than a specified threshold in order to satisfy

SNR requirement. And the relationship between the transmitted signal strength P_t and the received signal strength P_r based on the free space propagation model [75] is as follows,

$$\frac{P_r}{P_t} \propto \frac{1}{d^n} \quad (3.1)$$

where d is the distance between the transmitter and the receiver and n is normally ranged from 2 to 4. Hence, when the required minimum received power is achieved, reducing the distance between a pair of nodes, n_i and n_j , can reduce the required transmitted signal strength at node n_i .

To obtain the route with minimum total power between a communication pair, the transmission power between nodes n_i and n_j can be used as a metric [76] because more power is consumed for packet transmission than packet reception or during idle periods [9]. Then, the total transmission power P_l for route l is shown as

$$P_l = \sum_{i=0}^{L-1} P(n_i, n_{i+1}) \quad (3.2)$$

where $P(n_i, n_{i+1})$ is the transmission power from n_i to n_{i+1} for all mobile nodes $n_i \in \text{route}$, and n_0 and n_D are the source and destination nodes, respectively. Then the desired route based on minimum total transmission power can be obtained from

$$P_k = \min_{l \in A} P_l \quad (3.3)$$

where A is the set including all possible routes.

However, this algorithm likely selects routes with more hops [9], which may involve more mobile nodes as data relays and introduce longer end-to-end packet delivery delay. Hence, the route obtained solely based on minimum transmission power consumption metric is not desirable, and the metric also considering the power

consumption for receiving data is introduced in [9, 77] in order to select routes with less number of hops.

3.3.1.2 Online Max-Min Routing (OMM)

The OMM protocol [59] minimizes the power consumption for a data communication pair. Given all link costs in terms of transmission power consumption of all possible routes between a source-destination pair, OMM protocol can find the optimal path by using *Dijkstra's algorithm* [69], and this path consumes the minimal transmission power, P_{min} , and is named min-power path. However, OMM also maintains multiple near-optimal min-power paths with a near optimal value zP_{min} , where $z \geq 1$. Among them, OMM selects a path, in which the minimum residual energy node is with higher energy level after transmission as compared to the minimum residual energy node in other near optimal min-power paths, as the active path for communication between this source-destination pair. This selected path is called max-min path, and will be used for data delivery for a period of time until the source tries to find another replaced max-min path by repeating the procedure as addressed above. This metric is helpful in preventing the occurrence of overloaded mobile nodes [59, 74].

Figure 3.3 shows an example of the algorithm for a given pair (S, D). In Figure 3.3(a), the path $S \rightarrow B \rightarrow D$ is the min-power path because it consumes the least transmission power with $P_{min} = 18$. If we set $z=2$, alternative paths $S \rightarrow A \rightarrow D$ (path cost=22) and $S \rightarrow C \rightarrow D$ (path cost=31) can also be considered since their path cost are within the tolerance range of $zP_{min} = 36$. In order to obtain the max-min path, the mobile node with the minimal residual power in each path is compared. Refer to Figure 3.3, nodes A, B and C are compared. After this transmission, the residual energy of nodes A,

B and C will drop to 13, 2 and 9 respectively. Therefore, the max-min path is $S \rightarrow A \rightarrow D$ as shown in Figure 3.3(b). The parameter z measures the tradeoff between the max-min path and the min-power path. Hence, the proper selection of z is important in determining the energy consumption optimization [74]. Also, how to adaptively adjust the value for z can be referred to [59].

Figure 3.3 Min-Power Path and Max-Min Path in OMM.

3.3.2 Load Distribution Protocols

The objective of the load distribution approach is to avoid using mobile nodes with low residual energy level as relays if possible. Thus, the network partition and communication interruption due to the energy depletion of partial mobile nodes can be avoided. Generally, this can be achieved by routing packets through energy-rich mobile nodes. Protocols based on this approach do not necessarily provide the minimum energy consumption route. But they prevent certain mobile nodes from being overloaded and draining their energy too fast. Thus, network operation time can be prolonged [10, 74].

3.3.2.1 Conditional Max-Min Battery Capacity Routing (CMMBCR)

CMMBCR [9] manages to maximize the lifetime of each mobile node and use the battery energy of all mobile nodes fairly as well. CMMBCR uses a threshold to decide whether a mobile node is with enough battery to serve as relays for others. When all mobile nodes along some possible routes between a source-destination pair have enough battery capacity, the route with minimum total transmission power consumption is chosen as the active route. Since the selected path requires less total transmission power to deliver packets, the relay loads for most mobile nodes will be reduced and their lifetime can be extended [9]. However, if all possible routes contain mobile nodes with battery capacity lower than the threshold, the route, in which the minimum energy mobile node is with the highest residual battery as compared to corresponding mobile nodes in other routes, is chosen for packet delivery.

By adjusting the threshold, the network partition time or the average lifetime of mobile nodes can be maximized, and thus caters for different applications.

3.3.2.2 Localized Energy Aware Routing (LEAR)

LEAR [61] is originated from DSR but modifies DSR's route discovery procedure in order to choose mobile nodes with enough battery capacity as data relays, and hence to avoid the early "death" of partial mobile nodes due to energy depletion. In LEAR, a mobile node decides whether to process a RREQ depending on its residual battery capacity E_r . If E_r is higher than the threshold Th_r , the mobile node forwards the RREQ packet. Otherwise, it drops the RREQ. Hence, when the RREQ arrives at the destination, it contains a route with all intermediate nodes with satisfying energy levels.

As E_r for mobile nodes decreases with time, the value of Th_r should be adjusted adaptively to identify energy-rich mobile nodes and energy-poor ones dynamically [61, 74]. If a source node does not receive any RREP within a specified time for its out-going RREQ message, the source will send a duplicated RREQ with a different sequence number. When an intermediate node receives the duplicated RREQ, it adjusts its Th_r to allow forwarding to continue.

However, if an intermediate node utilizes some route recorded in its RC to reply to a RREQ without evaluating the battery capacity of mobile nodes along this route, it implies that the returned route may include energy-poor nodes. To solve this problem, a new routing packet, named route_cache (RCHE) is introduced in LEAR. As shown in Figure 3.4, when node **B**, upon receiving a RREQ from the source node **S**, has a recorded route to the destination **D** in its RC, it stops broadcasting the RREQ but initiates a RCHE message to **D**. As long as intermediate nodes along the route from **B** to **D** have battery energy above Th_r , the RCHE can reach **D** successfully. Hence, **D** can reply a route composing of energy-rich intermediate mobile nodes to **S**. Otherwise, the RCHE will be dropped by some intermediate node, whose battery energy is lower than Th_r . The RCHE message does not add significant routing traffic because it is delivered in unicast mode. With this mechanism, LEAR can effectively make use of RCs at mobile nodes for route discovery and routing traffic reduction; while still discover energy-rich paths.

Figure 3.4 Illustration of RCHE Message Utilization in LEAR.

3.3.3 Sleep/Power-down Mode Protocols

Based on statistics, a mobile node in a network is often with idle status [8, 57], which means it does not transmit or receive any useful information. Also, the radio transceiver often consumes significant amount of power of a mobile computing device [7, 8]. It is desirable to put the radio into power save/sleep mode or simply turn it off to save energy since most radio hardware supports low power states [8]. Sleep/power-down mode protocols manage to put partial mobile nodes into power saving mode at a time and periodically rotate this portion among all mobile nodes when possible without affecting the network performance much. Hence, the lifetime of individual mobile nodes as well as the network lifetime can be extended. This requires the cooperation between the power saving mechanism and other functionalities of a mobile node. In a multi-hop MANET, the coordination of power saving with routing is especially important considering mobile nodes may depend on each other for packet forwarding [74].

3.3.3.1 Span Protocol

Span [8] dynamically maintains a DS for a MANET. A mobile node decides to become a coordinator (a mobile node in DS) if two of its direct neighbors cannot reach each other directly or via one or two coordinators. This rule does not guarantee the minimum DS size but provide robust connectivity in the network because each mobile node is covered in the radio range by at least one coordinator [8, 74]. Then, other mobile nodes except those coordinators can put their radio into power saving mode. A non-coordinator node also needs to wake up periodically and check whether it should join the DS based on the addressed rule. In addition, the rotation of the coordinator role among mobile nodes is important to avoid the early energy depletion of partial mobile nodes. So, a coordinator will mark itself as a tentative coordinator after serving a period of time. If it finds out that every pair of neighbors can reach each other via one or two other neighbors, even those neighbors are not coordinators, it can resign its coordinator role to some neighbor. Otherwise, it remains as a coordinator.

Span can be implemented with many routing protocols, however, geographic forwarding (GF) [8] is utilized due to its simplicity. Span's coordinator election algorithm requires a mobile node to advertise its status (coordinator, tentative coordinator or non-coordinator), coordinator(s) and neighbor list in its neighborhood periodically. To reduce this overhead, the Span Hello information is piggybacked on the periodic broadcast updates required by GF. Thus, each mobile node can obtain its own neighbor list, coordinator list and such information for its 1-hop neighbors. GF forwards a packet using a greedy algorithm [58]. When a coordinator receives a data packet for a mobile node not within its radio coverage, it forwards the packet to a neighboring coordinator, who is the closest to the destination. If no such coordinators, it then

forwards the data packet to a non-coordinator, who is closer to the destination compared to itself. Otherwise, it drops the data packet. As the coordinators can form a connected backbone for data forwarding and those non-coordinator nodes can be safely turned into power sleep mode without affecting the packet delivery.

3.3.3.2 Geographic Adaptive Fidelity Protocol (GAF)

In GAF [58], each mobile node utilizes the location information provided by GPS to associate itself with a virtual grid so that the entire network area is divided into several square grids as shown in Figure 3.5. A mobile node should be able to reach other mobile nodes in a neighboring grid for inter-grid communication; hence the relationship between the grid size r and the transmission range of a mobile node R should satisfy:

$$r^2 + (2r)^2 \leq R^2 .$$

Figure 3.5 Virtual Grid Structure in GAF.

In GAF, mobile node with the highest residual battery capacity within each grid becomes the master of the grid. Hence, other mobile nodes in the same grid can be released from the inter-grid routing and packet forwarding, and they can be safely put to power saving mode without affecting the routing efficiency. In order to participate in the

master election in GAF, a mobile node may experience the three possible states as shown in Figure 3.6 [58]: *sleeping*, *discovering* and *active*. A slave node of a grid periodically becomes awake to make sure that there is a master node in its grid to stay awake and route packets. Normally a master node continues serving its grid for a period of time T_a . Then it changes its status to *discovery* to give other mobile nodes in the same grid a chance to become the grid master. A slave node wakes up every T_s seconds and changes to discovery status. Then, it tries to become a master by sending out discovery message including its grid ID and its residual energy. A mobile node successfully wins the master role if it does not hear any other discovery message for a predefined duration T_d . If more than one mobile node is in the discovery status, the one with the highest energy level becomes a master.

Figure 3.6 Illustration of Master Election in GAF.

3.4 Conclusions

Dynamic routing is an important behavior for a multi-hop MANET because it decides how a mobile node functions as a relay and how a packet is delivered in the

network. Hence, dynamic routing can affect the overall performance of such a data communication network. Since scalability and energy limitation are among the most challenging issues for MANETs, mechanisms concerned about how to make a MANET scale well and how to consume mobile nodes' energy wisely should be integrated into routing schemes to enhance the network performance.

Providing hierarchical structure and maintaining a routing backbone are considered as the promising techniques for solving the scalability issue of MANETs without additional hardware cost. Hence, several hierarchical routing schemes have been reviewed in this Chapter. CGSR and CBRP are cluster based routing schemes. They both maintain a 1-hop cluster structure with comparatively low clustering cost because their re-clustering is event-driven and invoked in limited situations. CGSR uses proactive DV mechanism for inter-cluster routing whereas CBRP utilizes on-demand source-routing mechanism for inter-cluster routing. ZRP defines a routing zone for each mobile node, and proactive and on-demand mechanisms are used respectively for routing to a destination node inside and outside of a predefined zone. The RREQ/RREP route discovery mechanism is utilized for the on-demand routing component, and a mechanism named BRP is introduced to effectively reduce the RREQ flooding traffic.

Energy metrics are implemented in routing mechanisms in order to consume the energy of each mobile node in a MANET more wisely and fairly. Different energy-aware routing schemes may focus on different aspects [78]. Transmission power control protocols try to minimize the transmission power consumption for delivering a packet from its source to its corresponding destination because transmission usually consumes more power as compared to receiving or listening/idle. Load distribution schemes manage to use mobile nodes with higher energy level as relays and thus to avoid the

early death of partial mobile nodes because of energy depletion when possible. Hence, the occurrence of network partition and communication interruption due to lack of living mobile nodes can be avoided or delayed. Sleep/power down mode protocols try to put mobile nodes into the power saving mode when possible because mobile nodes consume considerably less power in this mode. This can be achieved by turning off partial mobile nodes while remaining others active to keep enough connectivity and route diversity for the network. By cooperation, mobile nodes can switch between active mode and power saving mode so that the energy consumption for different mobile nodes can be kept more fairly.

Chapter 4

An Efficient Clustering Scheme for Large and Dense MANETs

4.1 Introduction

As elaborated in Chapter 2, a MANET with flat structure cannot support scalability well, and hierarchical structure is essential for achieving basic performance guarantee in a large dynamic MANET [3-5, 19]. The cluster-control structure, as a typical hierarchy, is often used to solve the scalability issue in large dynamic MANETs. In addition, cluster structure brings great convenience in intra-cluster and inter-cluster transmission management [23], and consequently maximizes the network throughput.

In general, clustering schemes can be divided into CH-based clustering schemes and non-CH-based clustering schemes. In the CH-based clustering, protocols can be further categorized to 1-hop clustering and multi-hop clustering based on the cluster diameter. In this Chapter, we focus on 1-hop CH-based clustering schemes. In 1-hop clustering, the hop distance between a CH and any of its members is 1. And the maximum hop distance between two member nodes in the same cluster is 2. In multi-hop clustering, a CH is not necessarily directly connected with all its members. CHs can reach its farther member nodes with the help of intermediate member nodes. Usually, the

cluster maintenance is more complicated and difficult in multi-hop clustering, especially under high mobility environment [31].

Clustering can be normally separated into two phases, cluster formation and cluster maintenance. Cluster formation refers to initially how to build a cluster structure when a MANET begins to function. Cluster maintenance is about how to dynamically update the cluster structure according to the underlying network topology change during the operation.

As discussed in Chapter 2, a lot of CH-based clustering schemes require the elected CHs are with specific attributes in the cluster formation phase. For example, in LIC [29], a mobile node can claim as a CH only if it has the lowest ID compared with all direction neighbors with unspecified clustering status. In HCC [16], however, a mobile node claims as a CH only when it finds out that it has higher node degree than all direct neighbors, who have not determined their clustering status yet. Mobile nodes in those schemes need to exchange corresponding information in their neighborhood to elect appropriate CHs so that they are assumed frozen [16, 29, 30] to guarantee that the exchanged information is accurate and that the elected CHs satisfy the pre-defined properties. However, this kind of assumption may not be applicable in an actual scenario, where mobile nodes may move randomly all the time [4, 38].

In 1-hop clustering schemes, the distance between CHs of neighboring clusters is normally 2-hop [5, 13, 22, 23]. Limiting the number of hops between CHs has potential to reduce inter-cluster communication delays. But in a network with moderate to high node density, this likely forms a large number of highly overlapping clusters [12, 37]. A highly overlapping cluster structure makes CH density increase and most member nodes are covered by more than two CHs. Thus, a single node's movement likely affects more

clusters. Even though a cluster structure helps build a routing backbone [4, 12, 25, 79], a highly overlapping cluster structure needs more CHs to cover mobile nodes in a MANET and more clustergateways (CGWs) for inter-cluster connection, and this is not favorable for simplifying a network structure.

How to update a cluster structure to respond to the underlying network topology change is important because it affects the performance of a clustering scheme in a dynamic environment. Frequent membership update and especially frequent CH change adversely affect the cluster stability, which is important for performance guarantee of protocols residing on the cluster structure, such as dynamic routing and channel assignment. Hence, frequent cluster structure update is expensive because it introduces extra clustering-related control overheads, invalidation of existing routes, and even ripple effect of re-clustering, which causes cluster structure re-buildt over the whole network [4]. Thus, to maintain a stable cluster structure, especially to minimize CH change should be a major target for a clustering scheme.

Clustering schemes normally require mobile nodes to exchange explicit clustering-related control messages for cluster structure formation and maintenance. Thus, clustering behavior competes for the limited radio band resource with other network events, such as dynamic routing and data delivery [4, 36]. Hence, clustering overheads, referred to the number of clustering-related messages sent from mobile nodes during the network operation time, is a key issue to validate the effectiveness and enhancement of a cluster structure and should be quantitatively taken into consideration.

This Chapter proposes a 1-hop CH-based clustering algorithm, named ECS, to solve the common problems mentioned above for typical 1-hop clusters, such as frozen period requirement for cluster formation, high cluster overlapping, frequent cluster

structure change, and large clustering overheads. ECS can be extended to a k -hop ($k > 1$) clustering scheme by relaxing the requirement that members of a cluster must be directly (1-hop) connected with their CH. The modification to become a k -hop for ECS is minimal on the clustering mechanisms.

In ECS, the frozen period for initial cluster structure construction is not necessary because a mobile node can claim to be a CH without exchanging information with its neighbors as long as it is not covered by any CH. A newly introduced node status named *CG*, plus a maintenance mechanism called *cluster deletion* in ECS helps maintain a moderate overlapping cluster structure for a large and dense MANET by eliminating unnecessary small cluster, reducing the number of clusters and reducing the impact of mobile nodes' movement on cluster structure. ECS provides a stable cluster structure for MANETs because CH change in ECS is event-driven and invoked in just few situations. The clustering overheads can be kept comparatively low because ECS eliminates the ripple effect of re-clustering and limits the CH change frequency.

The performance of ECS is compared with two other typical 1-hop clustering schemes, RCC [5] and a modified version of HCC [16], named HCCM, in terms of average number of cluster, average cluster size, average CH lifetime, average clustering overheads and so on. The original HCC is based on the node degree for cluster formation and maintenance. Initially each mobile node obtains its node degree information by collecting the periodic Hello message from its neighbors and then includes this information in its following Hello message. Thus, the mobile nodes with the highest node degree in their neighborhoods claim as CHs. During the cluster maintenance phase, each CH periodically compares its own node degree with that of each neighbor by consulting its information table. If a neighbor with higher node degree is found, the CH then resigns

its CH role to that neighbor. Hence, the clustering structure changes accordingly. The algorithm of HCCM will be described in Section III. In RCC, a mobile node not involved in any cluster is with *unspecified* status and any mobile node with unspecified status can claim as a CH. But before a mobile node broadcasts a CH claim message, if it receives such message from some other mobile node in its neighborhood, it aborts its own CH claim and joins that neighbor's cluster as a CM. The cluster structure is updated accordingly in RCC when the underlying network topology is changing. The CH related change in RCC is only invoked in several situations: 1) When two CHs move into the reach range of each other, one of them gives up the CH role. 2) When a mobile node becomes disconnected with its CH(s), it claims as a new CH in order to cover itself in some cluster.

The remainder of this Chapter is organized as follows. In Section II, ECS is explained in detail. Section III describes the simulation models and parameters. Section IV provides the simulation results and analysis for ECS together with RCC and HCCM. Finally, a brief conclusion about ECS is drawn in Section V.

4.2 Efficient Clustering Scheme (ECS)

In this Section, some concepts used in ECS are defined first followed by the description of the clustering mechanisms.

4.2.1 Concepts Clarification

Node Status: an item to describe the clustering-related status of a mobile node. Node Status of a mobile node can be set to CH, CM, CGW, CG, standalone and unspecified.

CH: one kind of Node Status. A node with CH status normally serves as a local coordinator for its cluster, in charge of intra-cluster transmission arrangement and inter-cluster routing, and its ID is usually utilized to identify the corresponding cluster.

CGW: one kind of Node Status. A node with CGW status is a member node with inter-cluster links, so that it can connect to neighboring clusters and forward information between clusters.

CM: one kind of Node Status, which indicates that the corresponding mobile node is a member node of some cluster without any inter-cluster link.

CG: one kind of Node Status, which indicates that the corresponding mobile node is out of the reach range of any existing CH, but is directly connected with some member node(s) of some cluster(s).

Standalone: one kind of Node Status, which implies that the corresponding mobile node is partition from the network and cannot communicate with any other mobile nodes in the network.

Unspecified: a Node Status used only for the initial cluster formation to indicate that a mobile node does not have a clustering-related status yet.

According to the clustering status definitions as addressed above, mobile nodes can be classified based on their different status. Hence, in a MANET running ECS, a mobile node can be a node with CH status, CM status, CGW status, CG status, standalone status or unspecified status. And it may be called a CH, a CM, a CGW, a CG, a standalone node or an unspecified node accordingly for simplicity.

Since a node with CG status in ECS is out of the reach range of any existing CH, but such a CG needs to be attached to some cluster. In order to help the CG node be attached and recognized by some cluster and its corresponding CH, some member node

(a CM or a CGW) of the cluster, which can directly connect with both the CH and the CG, is necessary to bridge between the CH and the CG. And such a member node is considered as the Access Point (AP) of the corresponding CG.

4.2.2 Cluster Formation

The cluster formation mechanism in ECS is based on *random claim* mechanism [5] to avoid frozen period requirement.

At the very beginning of cluster formation, the Node Status of all mobile nodes is set to be unspecified. Any mobile node with unspecified status can claim to be a CH by sending out a *CH claim* message. But a mobile node backs off a random period of time before it broadcasts out its CH claim. If a mobile node hears CH claim message from any other neighbor during the backoff period, it cancels its own claim, joins that neighbor's cluster as a CM and indicates such change to its neighbors in its next Hello message. Otherwise, it claims itself as a CH in the neighborhood, and its ID will serve as the CID to identify the corresponding cluster. By collecting the messages from neighbors, a CH can build its information table. Initially this table includes all members' ID, CID(s) and clustering-related status. Information table in a mobile node is a storage space for recording all its neighbors' clustering-related information. Details about the information table are addressed later. If a CH does not get any message from other mobile nodes to indicate their joining as member nodes within a predefined period of time since it sends out its CH claim, it changes its status to standalone. If a member node hears CH claim from more than one CH or finds out that it has a neighbor with a different CID, indicating it has some inter-cluster link. Hence, in the cluster formation process, no mobile node will become a CG. Figure 4.1 gives an illustration of the initially

constructed cluster structure for a MANET based on ECS cluster formation mechanisms. As shown in Figure 4.1, mobile nodes n_3, n_{11}, n_{18} and etc. serve as CHs, mobile nodes n_1, n_4, n_{10} and etc. are CMs, mobile nodes n_2, n_6, n_{13}, n_{20} and so on are CGWs, and mobile node n_{19} is a standalone node.

Figure 4.1 Illustration of Cluster Structure after Cluster Formation.

4.2.3 Cluster Maintenance

Cluster maintenance discusses issues related with how to update the cluster structure when the underlying network topology changes. And the cluster structure update includes ordinary membership update and cluster change. Ordinary membership update refers to how a non-CH mobile node changes its attaching cluster, whereas cluster change indicates changes related with CHs, such as the resign of an existing CH or the claim of a new CH.

4.2.3.1 Membership Update

The ordinary membership updates include (1) how a CM changes its status to CGW and vice versa; (2) how a member node changes its CID because of changing

attaching cluster(s); (3) how a member node changes from CM or CGW to CG and vice versa; and (4) how a standalone mobile node changes to a CM or a CG. The mechanisms corresponding to the first two situations are the same as those in other typical 1-hop clustering schemes [5, 16, 29, 30] so the descriptions are skipped here. The following addresses mechanisms for (3) and (4) due to underlying network topology change.

- When a member node (CM or CGW) of some cluster(s) moves and disconnects with all its CH(s), if this mobile node finds that it still connects with some member node(s) of some cluster(s) by consulting its information table, it chooses one of them as its AP and sets its status to be CG. Such status change will be indicated in the next Hello message of this mobile node and hence all its neighbors can update their information table accordingly.
- When a CG node hears CH claim from some mobile node or periodic Hello from some existing CH, it joins that cluster as a member node and changes its status to CM. Such change will be indicated in the next Hello message of this mobile node and hence all its neighbors can notice its cluster status change. So does the standalone node.
- When a standalone mobile node hears Hello message from some existing CM/CGW, it then joins the corresponding cluster as a CG and indicates such change in the neighborhood in its next Hello.

4.2.3.2 CH Change

CH change is event-driven in ECS and invoked in the following three cases:

- When two CHs moves into the reach range of each other, one will resign its CH role by broadcasting *CH resign* message and change its own status to CM. In

ECS, it is the one with fewer member nodes with CM status will resign. For the two meeting CHs **A** and **B**, if **A** is the first one to realize this meeting, it sends a unicast *CH meeting* message to **B**, including the number of member nodes with CM status in its table. If **B** finds out that the number of member nodes with CM status in its table is less, it simply broadcasts a CH resign message to give up its CH role. If **B** finds that it has more member nodes with CM status, it replies back a unicast CH meeting message also including its number of member nodes with CM status. Then **A** terminates its CH role by broadcasting CH resign message. The reason to choose the CH with fewer member nodes with CM status to resign in the CH meeting situation is because when an existing CH gives up its CH role, the impact to its member nodes with CM status is larger compared to that for CGW nodes. This is because those CGW nodes only need to delete the corresponding entry about the resigned CH from their information table and change their status if necessary. But for those CM nodes, they need to actively seek involvement into some cluster as CM/CG or build a new cluster to cover itself.

- When a non-CH node becomes uncovered by any CH and finds out that all neighbors are either CGs or with standalone status, it claims itself as a CH and the claim procedure is the same as that introduced in the cluster formation part. This mechanism promises that the distance between the newly claimed CH and its neighboring CHs is 3-hop, and thus moderates the cluster density for a large and dense MANET.
- By consulting its information table, if a CH node finds out that all its member nodes are with CGW status and are covered by at least two other CHs, it invokes

cluster deletion mechanism to give up its CH role by broadcasting a cluster resign message in its neighborhood and sets its status to CG by choosing one of the neighbors as its AP randomly. This mechanism reduces the number of clusters and the cluster overlapping. Also, this kind of CH resign does not affect neighboring clusters so that the cluster structure can be kept well.

Figure 4.2 is a snapshot of cluster structure in the cluster maintenance phase. Hence, we can see that there are four kinds of mobile nodes (except those standalone mobile nodes) in this MANET considering their clustering status, including CH, CM, CGW and CG.

Figure 4.2 Illustration of Cluster Structure in Cluster Maintenance.

4.2.3.3 Clustering Messages

Clustering messages used in ECS can be categorized into two types. The first type is periodic Hello message and the second type is event-triggered clustering message related with CH change.

For the first type of clustering message, each mobile node periodically broadcasts Hello message in its local area to indicate its availability in the neighborhood during the

whole operation time. This message includes its ID, status, CID(s) and AP ID. For a CH, its CID is the same as its ID. Only mobile nodes with CG status will have AP ID information in its Hello and the CID(s) for a CG are actually the CID(s) of its corresponding AP. A CG node may be able to connect to more than one neighbor with CM/CGW status, and in ECS each CG chooses at most two of them as its APs at a time.

The second type of clustering message is related with CH change, i.e. when a mobile node changes from a non-CH to a CH or when a CH resigns its CH role, it sends out specific clustering message. For ordinary membership change/update, i.e. when a mobile node changes its status from CM/CGW to CG and vice versa or when a member node changes its attaching cluster(s) or when a CG node changes its attaching AP node(s), it does not send out explicit clustering message but indicates the change in the Hello message. This is because the membership update of a mobile node usually does not affect other mobile nodes' clustering-related status and hence neighboring nodes do not need to be notified by specific clustering messages.

The attributes and functions of every clustering message used in ECS are summarized in Table 4.1 and Table 4.2 for clarification.

Table 4.1: Summary of Clustering Messages for ECS.

Message Type	Message Name	Periodic or Event-driven	Broadcast or Unicast
Hello	Hello	Periodic	Broadcast
Clustering Message	CH Claim	Event-driven	Broadcast
	CH Meeting	Event-driven	Unicast
	CH Resign	Event-driven	Broadcast

Table 4.2: Functions and Information of Clustering Messages for ECS.

Message Name	Functions	Information Included
Hello	To help neighbors refresh information about a mobile node;	Node ID, node status, CID(s), and AP ID(s);
CH Claim	To notify neighbors that a mobile node becomes a CH;	Node ID and node status,;
CH Meeting	To help corresponding CH realize the CH meeting and to send information between two meeting CHs;	Node ID and the number of its member nodes with CM status;
CH Resign	To notify neighbors that a CH changes to some other status;	Node ID, node status, CID, and AP ID(s);

4.2.3.4 Information Table

Information table is maintained at each mobile node to record its direct neighbors' information. When a mobile node receives a clustering message, any message in Table 4.1, from some other mobile node, it first checks whether the corresponding entry for that mobile node exists in its information table. If not, it builds a new entry for that mobile node. Otherwise, it updates the existing entry depending on the received message. Once a mobile node receives a message, it sets *entryUpdateTime* for the corresponding entry to current time no matter any information about this entry has changed or not. Also, a mobile node periodically scans its table and deletes obsolete entry for some neighbor when it finds out that the time elapsed is longer than predefined *entryValidPeriod* since last *entryUpdateTime*.

Table 4.3 shows a typical information table for mobile node n_8 under the cluster structure illustrated in Figure 4.2 as an example.

Table 4.3: Information Table for Mobile Node n_8 .

Node ID	Status	CID(s)	AP ID	entryUpdateTime (S)
8	CM	5	N/A	N/A
5	CH	5	N/A	100.0
7	CM	5	N/A	99.8
9	CM	5	N/A	99.6
20	CGW	5, 2	N/A	100.2
12	CG	2	18	99.3

The information table at a CH node may be slightly different since it records not only its direct neighbors' information but also the information for CG nodes, which attach to its cluster.

4.2.4 Summary of the Newly Proposed Mechanisms in ECS

ECS introduces a new clustering status for mobile nodes, named CG. When a member node loses its connection with its CH(s), it can still join some cluster as a CG as long as there is some neighbor to bridge it with some existing CH. This mechanism can avoid a mobile node to form any unnecessary clusters in order to just cover itself.

In ECS, when a CH node finds out that all its member nodes are covered by at least two other CHs, it can resigns its CH role and join some neighboring cluster as a CG. This is the cluster deletion mechanism in ECS. In a MANET with moderate to high node density, the formed cluster structure is normally highly overlapped. This makes a single mobile node's movement affect many clusters. Also, a highly overlapping cluster structure makes more mobile nodes unnecessarily bear the CH responsibility. The cluster deletion mechanism can help reduce the cluster overlapping in a MANET with high or moderate density without affecting the cluster structure much.

ECS promises that for a newly formed cluster, its CH is at least 3-hop away from any existing CHs. Hence, this is another mechanism to help moderate the cluster density for a cluster based large and dense MANET.

The new mechanisms proposed in ECS help to build a stable cluster structure and generate fewer clustering overheads by reducing cluster structure change, especially CH change, and avoid bearing more mobile nodes as CH by eliminating some unnecessary cluster and reducing the cluster overlapping. These mechanisms are effective to help ECS achieve its objectives and this can be proved by simulation results.

ECS can be easily extended to a k -hop ($k > 1$) clustering scheme just by relaxing the requirement that members of a cluster must be directly (1-hop) connected with their CH. In the k -hop ECS, the maximum distance between a member node and its CH is k -hops. The CG can still be introduced into a k -hop situation to help avoid formation of some unnecessary clusters. A CG is a mobile node that cannot reach any existing CH by k -hops but can still be involved into some cluster by the bridge function of some AP. In the k -hop environment, the cluster deletion mechanism can be implemented as in the 1-hop situation. When a CH finds out that all its member nodes are CGWs, which means that all of them can be reached by some other CH within k -hops, this CH can give up its CH role and join a neighboring cluster as a CG. Hence, all the newly proposed mechanisms in ECS can be implemented in a k -hop situation without much complex modification.

4.3 Simulation Models and Performance Metrics

4.3.1 Simulation Models

NS-2 simulator with the CMU wireless extension [80] is used in this Chapter for simulating the performance of all three schemes. NS-2 can simulate the physical, MAC and data link layer of a multi-hop wireless network. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs is utilized as the MAC layer [80, 81]. Lucent's WaveLAN is used as the radio model, which is a shared-media radio with a nominal bit rate of 2Mbps and a nominal transmission range of 250m. With the use of a NS-2 simulator, we can correctly model the effects of contention for the media and the distance between mobile nodes in determining whether a transmitted packet will be successfully received [81]. In the simulations, Random waypoint mobility model [82] is adopted for simulating the moving behaviors of all mobile nodes in such a MANET. Random waypoint mobility model involves some pause time between consecutive movements of a mobile node [82]. In Random waypoint model, initially a mobile node pauses at a location for a certain period of time. When this period of time expires, the mobile node begins to move directly to a random destination inside the simulated area with a specific speed. The moving speed is uniformly randomly chosen from (0.0, MaxSpeed]. Once the mobile node reaches the destination, it pauses again for the pre-defined time period before starting to move again. The movement behavior is repeated for all mobile nodes during the whole simulation process.

Table 4.4 shows default values for related parameters used in the simulations for ECS, RCC and HCCM.

Table 4.4: Default Simulation Parameters.

Parameters	Values
Simulation Time	600s
Simulation Area	2000m × 2000m
Transmission Range for Mobile Nodes	250m
Transmission Data Rate	2Mbps
Mobility Model	Random Waypoint
Pause Time for Mobile Nodes	50.0s
Max. Speed for Mobile Nodes, v_{max}	10.0m/s
Number of Mobile Nodes, N	500
Buffer Size for Queuing Outgoing Packets	50
Hello Message Interval, h_s	1.0s
Information Table Scan Interval	0.2s
<i>entryValidPeriod</i> for Information Table Entries	2.2s
Clustering Events Backoff Time	0.1s

In Table 4.4, Buffer size for queuing outgoing packets refers to the maximum number of packets can be buffered in the interface queue before being sent out. Information Table Scan Interval decides how often a mobile node scans its table to delete obsolete entries. Clustering Events Backoff Time refers to the backoff time for sending clustering messages except Hello. Clustering Events Backoff Time is introduced for avoiding concurrent clustering messages in a neighborhood because such messages may influence each other. The successful transmission of some clustering message from some mobile node may cancel the upcoming clustering message or invoke new clustering message at other mobile nodes. HCCM has no CH meeting message because its CH change is only decided by the node degree difference between an existing CH and its neighbors.

4.3.2 Performance Metrics

ECS is compared with RCC and HCCM in terms of average number of clusters N_c , average cluster size S_c , average CH lifetime T_c , average membership lifetime T_m and average clustering control overheads O_c .

By comparing N_c , we can check whether ECS maintains a cluster structure with less number of clusters compared with HCCM and RCC and achieves its goal in reducing cluster overlapping.

S_c can help to check whether a CH is overloaded in ECS by involving more member nodes into a cluster. And S_c refers to the number of mobile nodes covered by a CH and is defined as follows:

For HCCM and RCC,

$$S_c = \frac{a + b + \sum_{i=1}^c n_i}{a} \quad (4.1)$$

where a and b are the number of CHs and CMs in the network respectively. c refers to the number of CGWs, and n_i indicates the number of CHs that a CGW node, i , is covered by.

For ECS,

$$S_c = \frac{a + b + \sum_{i=1}^c n_i + \sum_{j=1}^d m_j}{a} \quad (4.2)$$

where d is the number of CG nodes in the network and m_j indicates the number of APs that a CG is connected with.

T_c is denoted as the average time period for a mobile node taking the role of CH continuously. And T_c is an important metric to judge the stability of a clustering scheme.

T_m is another metric for judging the stability of a clustering scheme. In HCCM and RCC, T_m is denoted as the average time period for a member node attaching to the same CH. In ECS, T_m indicates the average time period for a member node attaching to the same CH and a CG node attaching to the same AP.

O_c includes Hello message and clustering message due to the CH change sent from all mobile nodes during the operation. O_c is a necessary criterion to validate a clustering scheme. To maintain a perfect cluster structure with large O_c will consume much network resource and degrade the overall network performance.

4.3.3 Modifications of HCC

It is well known that cluster topology changes more frequently in typical HCC compared with other clustering schemes, which are based on event-driven CH changes. Thus, HCC has short CH lifetime [16, 30] and large clustering overheads. Hence, we modify HCC in some ways to make it perform better and make the comparison among the three schemes more meaningful. And the modified HCC is named HCCM in this Chapter. The difference of HCCM from HCC comes from how a CH gives up its CH role to some neighbor. A CH resigns only if it finds a neighbor satisfying the following two conditions:

- The corresponding neighbor's node degree is at least 3 higher than itself. This mechanism indicates that a CH in HCCM can somewhat tolerate neighbors with higher node degree and promise less CH change, especially in a large and dense MANET as simulated in this Chapter.
- The corresponding neighbor is a new neighbor judged from its information table, which means that the difference between the last update time of that neighbor and

the first record time of that neighbor is less than a predefined threshold, called *stablePeriod*. Thus, this mechanism requires that the information table in HCCM has some additional information for each entry, named *entryRecordTime*, which is the time that the corresponding entry is first recorded in the table. In this Chapter, the value of *stablePeriod* is set the same as that of *entryValidPeriod*. This can promise that node degree alone cannot invoke CH change and the cluster structure change always responds to the underlying network topology change.

If a CH finds out that several neighbors can satisfy the previous two conditions by consulting its information table, it chooses the one with the highest node degree among them to replace its CH function.

4.4 Simulation Results and Discussions

In this simulation, we study the performance of ECS with different v_{max} for mobile nodes, different N in a network or different Hello message interval, h_b . The simulation result of each point on the plotted figures is the average over 10 random simulations following the specific mobility and traffic parameters.

4.4.1 Effects of Node Mobility

Fig. 4.3 – Fig. 4.9 show the simulation results in terms of T_c , T_m , N_c , S_c , O_c-H and O_c for HCCM, RCC and ECS with different v_{max} of mobile nodes.

Figure 4.3 shows the comparison of T_c among HCCM, RCC and ECS with different v_{max} . The number, N , of mobile nodes is 500 in this simulation. All the three schemes' performance drops with v_{max} . For RCC and ECS, CHs more likely meet each other in a high mobility scenario, resulting in shorter CH lifetime. For HCCM, a high

mobility environment makes the node degree of a mobile node changes frequently and hence likely invokes more CH changes. The CH lifetime for RCC and ECS is about 240% and 290% of that of HCCM. This is because CH changes in RCC and ECS are invoked in few limited situations, so that cluster stability can be greatly improved. CH lifetime of ECS is about 20% longer than that of RCC. This is because the major CH changes of the two schemes come from CH meeting and the chance of CH meeting in ECS is smaller than that in RCC due to a less overlapping cluster structure and a lower CH density, which is proved by later simulation results. CH lifetime of original HCC is also shown in Figure 4.3 and it is found that the performance of HCC is quite poor in such a large and dense MANET, resulting from frequent CH change. Hence, we do not include HCC in later simulation studies.

Figure 4.3 T_c vs. v_{max} with $N = 500$.

Figure 4.4 T_m vs. v_{max} with $N = 500$.

Figure 4.4 shows the comparison of T_m among HCCM, RCC and ECS with different v_{max} . Similarly, the performance for all three schemes drops with v_{max} . T_m of all the three schemes is shorter than their corresponding T_c as shown in Figure 4.3. This is because a mobile node’s membership lifetime is affected by (1) its connecting or disconnecting with its CH due to mobility and (2) the change of its and surrounding CH in its neighborhood. T_m of ECS is about 100% and 10% longer than that of HCCM and RCC. The membership lifetime improvement of ECS over the other two schemes is smaller compared with CH lifetime improvement. This is because the membership update for a mobile comes from two aspects. The first one is due to the mobility, which makes a member node leaving or joining a cluster from time to time. The membership updates due to the mobility is almost the same for the three schemes. The second type of update is the CH change in the neighborhood of a mobile node. The membership lifetime

improvement of ECS over the other two mainly comes from the second type of updates because ECS has less CH change.

Figure 4.5(a) shows N_c with different v_{max} for the three schemes. HCCM produces much more clusters compared to RCC and ECS. This is because no CH resigns even when two CHs meet with each other in HCCM. And it implies that the distance between neighboring CHs can be less than 2-hop during operation. Hence, more clusters exist to cover all mobile nodes. Since the minimum hop distance between neighboring CHs in RCC and ECS is always kept at 2, their corresponding cluster number is smaller. ECS produces lesser number of clusters compared with RCC. This is because the introduction of CG and cluster deletion mechanism in ECS helps eliminate some unnecessary clusters. In addition, the CH claim mechanism during cluster maintenance in ECS can promise the newly claimed CH is 3-hop away from neighboring CHs. Thus, the cluster density in ECS can be kept at a comparatively low level. N_c of RCC and ECS slightly increases with v_{max} . This is because by adopting random waypoint model, mobile nodes more likely gather at the centre part of the simulation area in a low mobility environment than in a high mobility scenario [82] and thus, fewer clusters are needed to cover all mobile nodes in a low mobility scenario. Figure 4.5(a) shows that HCCM forms highly overlapped clusters. Figure 4.5(b) shows the comparison of S_c among the three schemes. HCCM normally chooses the mobile node with highest node degree in a local area to be a CH, so it has the largest S_c among the three. S_c of ECS is about 10% higher than that of RCC because ECS forms less number of clusters mainly by eliminating small clusters. It can be seen that the speed of mobile nodes does not affect N_c and S_c much.

Figure 4.5 N_c and S_c vs. v_{max} with $N = 500$.

In order to explain that N_c for HCCM is about twice that of ECS but S_c for them is about the same as shown in Figure 4.5, a more detailed comparison between HCCM and ECS about the mobile nodes' clustering status distribution is given in Figure 4.6 and Figure 4.7. In Figure 4.6 and Figure 4.7, the clustering status of a mobile node indicates that it is a CH or a member node. In order to have insight comparison between the maintained cluster structure of HCCM and ECS, a member node is further classified by the number of its covering CHs, as a member node covered by one CH, a member node covered by two CHs, a member node covered by three CHs, and a member node covered by four or more CHs.

Figure 4.6 Clustering Status Distribution of HCCM vs. Simulation Time with $v_{max} = 10\text{m/s}$ and $N = 500$.

Figure 4.7 Clustering Status Distribution of ECS vs. Simulation Time with $v_{max} = 10\text{m/s}$ and $N = 500$.

By observing Figure 4.6 and Figure 4.7, we can see that the clustering status distribution for both schemes is generally stable as simulation goes by, i.e., the

percentage of mobile nodes for each clustering status group does not vary much with simulation time. According to Figure 4.7, ECS has about 34%, 42%, 11.5% and 0.5% of cluster member nodes covered by one, two, three, and four or more CHs respectively. However, as shown in Figure 4.6, HCCM has about 7%, 17%, 18% and 43% of cluster member nodes covered by one, two, three, and four or more CHs respectively. This indicates that HCCM forms a much more overlapping cluster structure as compared to ECS, and many CHs may be directly connected with each other. Refer to (4.1), for a member node with one covering CH, indicating it is attached to only one CH, it is counted one time in the calculation of S_c , whereas for a member node with four covering CHs, it is counted four times in the calculation of S_c as it is attached to four CHs. Hence, the statistics on Figure 4.6 and Figure 4.7 can probably help explain why HCCM has more clusters than ECS but they two have similar cluster size.

Figure 4.8 O_c-H vs. v_{max} with $N = 500$.

Figure 4.8 shows the comparison of clustering messages due to the CH change among the three schemes. Since this overhead is the total number of clustering overheads deducted by the periodic Hello messages, it is called O_c-H . Figure 4.8(a) is the comparison between ECS and RCC, whereas Figure 4.8(b) is the comparison between ECS and HCCM. The CH change overhead of ECS is always lower than that of RCC for about 15% to 30% depending on v_{max} . This is due to the less overlapping cluster structure and the lower CH density in ECS, resulting from the use of CG and cluster deletion mechanism. These mechanisms in ECS make its cluster structure better tolerate underlying network topology change, especially under a high mobility environment. Figure 4.8(b) shows that HCCM produces 10 times more of CH change overheads compared with RCC and ECS. This indicates that a clustering maintenance mechanism that requires CHs to bear specific attributes in their local neighborhoods, such as HCCM requiring its CHs are always with the highest node degree, may cause frequent cluster structure change, especially in a high mobility environment.

Figure 4.9 shows the comparison of O_c among the three schemes. O_c consists of Hello messages and clustering messages related with CH change. Since the Hello frequency for all three schemes is the same, the Hello overheads are almost the same, and count a large portion among O_c for all the three schemes because a comparatively high frequency is adopted for the Hello. Then, the difference of O_c among three schemes mainly comes from O_c-H . HCCM produces much larger O_c-H due to frequent CH change resulting in generating a high number of CH change messages as shown in Figure 4.8(b). The clustering overheads of RCC and ECS are quite close because Hello messages count for more than 95% of the total clustering overheads in the two schemes, and then the clustering overheads difference from the CH change messages becomes insignificant.

Figure 4.9 O_c vs. v_{max} with $N = 500$.

4.4.2 Effects of Network Size

Fig. 4. 10 – Fig. 4.14 show the performance of HCCM, RCC and ECS in terms of T_c , T_m , N_c , S_c , O_c-H and O_c with different number of mobile nodes, N , in the network.

Figure 4.10 shows the change of T_c with different N at $v_{max}=10\text{m/s}$. As N increases, T_c for all 3 schemes decreases. With the same simulated area, larger N means higher node density. For a CH in HCCM, its CH status can be affected by any neighboring nodes. Hence, a CH is more likely affected in a higher node density environment resulting in more frequent CH changes. Thus, the CH lifetime drops with N in HCCM. For RCC and ECS, a higher node density scenario makes a higher cluster density as well. In other words, a CH is surrounded by more neighboring CHs resulting in higher CH meeting

chance and shorter CH lifetime. Similarly, ECS outperforms RCC and HCCM as explained in Figure 4.3.

Figure 4.10 T_c vs. N with $v_{max} = 10\text{m/s}$.

Figure 4.11 shows the change of T_m with different N at $v_{max} = 10\text{m/s}$. It can be seen that T_m decreases with N for the three schemes. This is because the membership lifetime of a mobile node is affected when a CH change takes place in its local neighborhood. As N increases, the CH change occurs more often to cause more membership change. ECS can outperform RCC and HCCM by about 10% and 100% respectively and the reason can be referred to the explanation for Figure 4.4. As seen from Figure 4.10 and Figure 4.11, T_c and T_m of HCCM drop greatly with a large N because the chance that a CH has a new neighbor with higher node degree is higher in a dense network.

Figure 4.11 T_m vs. N with $v_{max} = 10\text{m/s}$.

Figure 4.12 N_c and S_c vs. N with $v_{max} = 10\text{m/s}$.

Figure 4.12(a) and Figure 4.12(b) show N_c and S_c for the three schemes respectively with different N . As N increases, N_c increases for all three schemes. This is because a higher node density increases the clustering overlapping, and hence more clusters are needed for covering all mobile nodes. A higher node density makes the number of mobile nodes covered by a CH increase as well, and thus S_c increases. The other observations are similar to Figure 4.5.

Figure 4.13(a) and Figure 4.13(b) show the performance comparison for CH change overheads among the three schemes with different N . O_c-H increases with N for all three schemes. Since the increase of N makes T_c for the three schemes drop as shown in Figure 4.10, more frequent CH change occurs. Similarly, ECS outperforms RCC and HCCM as explained in Figure 4.8.

Figure 4.13 O_c-H vs. N with $v_{max} = 10\text{m/s}$.

Figure 4.14 O_c vs. N with $v_{max} = 10\text{m/s}$.

Figure 4.14 shows the comparison of O_c with N among the three schemes. O_c is almost proportionally increased with N for the three schemes because periodic Hello message counts a considerably large portion among O_c , and the Hello is proportionally increased with N as a fixed Hello frequency is adopted for the three schemes. Comparing to the number of Hello messages, the clustering overheads due to CH changes for RCC and ECS become ignorable and count about 3% among O_c , and hence O_c of the two is almost the same. The difference of O_c between HCCM and RCC/ECS mainly comes from the overheads due to the CH change as shown in Figure 4.13.

4.4.3 Effects of Hello Frequency

According to the simulation results shown in Section 4.4.1 and 4.4.2, the performance of RCC and ECS is more outstanding and more comparable to each other,

as compared to HCCM, and hence it is more interesting to study how RCC and ECS will perform as Hello message interval, h_B , varies. Fig. 4.15 – Fig. 4.17 show the performance of RCC and ECS in terms of T_c , T_m and O_c-H with different h_B . In order to study how h_B affects the performance of RCC and ECS in different mobility scenarios, the simulation results of RCC and ECS vs. h_B with two v_{max} , 5m/s and 15m/s, are presented. And the two different v_{max} , 5m/s and 15m/s, can represent a low-to-medium mobility scenario and a medium-to-high mobility scenario respectively.

Since the Hello messages are used to maintain the information table at each mobile node and the cluster structure is built and maintained by the Hello messages and information tables, the Hello message interval cannot be longer than *entryValidPeriod*, 2.2s, for entries in information tables, otherwise the cluster structure cannot be maintained. Thus, h_B is varied in the range of [0.5s, 1.0s, 1.5s, 2.0s] for simulations in this Section.

Figure 4.15 T_c Vs. h_B with $N=500$.

Figure 4.16 T_m Vs. h_B with $N=500$.

It can be seen from Figure 4.15 and Figure 4.16 that as h_B increases, T_c and T_m of both RCC and ECS decreases. As h_B increases from 1.0s to 1.5s, T_c and T_m for both RCC and ECS drops dramatically, especially for $v_{max}=5m/s$. This is probably because *entryValidPeriod* is set to 2.2s and corresponds to about 4 times and 2 times of h_B when h_B is 0.5s and 1.0s respectively. However, when h_B is set to 1.5s or 2.0s, the *entryValidPeriod* is less than 2 times of h_B . Hence, a mobile node will treat a neighbor entry invalid and delete it from its information table as long as one Hello message is lost from the corresponding neighbor, no matter the actual link with the neighbor is lost or not. However, message lost from neighbors is highly possible due to transmission collision as the underlying MAC scheme runs independently on each mobile node and there is no central transmission schedule. This frequent neighbor entry change, due to small ratio between *entryValidPeriod* and h_B , may make mobile nodes illusively experience the frequent change of network topology and invoke unnecessary re-

clustering. The drop for low mobility scenario is more obvious because the re-clustering due to movements of mobile nodes and actual link breakage between neighboring nodes occurs less frequently in such a low mobility environment. Hence, the re-clustering due to mobile nodes' illusive experience of topology change may affect more on T_c and T_m as compared to a highly mobile environment. When the ratio between *entryValidPeriod* and h_B is larger than 2, mobile nodes can in some extend bear Hello message lost from neighbors due to transmission collision, and then the information table of a mobile node can more correctly reflect its actual local network topology because the chance for failing to receive two or more consecutive Hello messages from a neighbor should be much lower. With simulation results from Figure 4.15 and Figure 4.16, it should be able to conclude that a proper ratio between *entryValidPeriod* and h_B is important for a clustering scheme to function correctly, and a ratio larger than 2 should be good enough to maintain the clustering performance.

Figure 4.17 O_c-H vs. h_B with $N=500$.

It can be seen from Figure 4.17 that O_c-H for both RCC and ECS is increased with h_B for $v_{max}=5\text{m/s}$ and $v_{max}=15\text{m/s}$ as well. When a comparatively large h_B is chosen, mobile nodes may illusively think that the network topology changes fast due to the reason explained for Figure 4.15 and Figure 4.16, and hence the clustering mechanisms will invoke more clustering messages to adjust the cluster structure in order to adapt to the frequently changed underlying network topology. Consequently, as the clustering mechanisms trigger more messages related with re-clustering, T_c and T_m , as shown in Figure 4.15 and Figure 4.16 drops.

As seen from Figure 4.15 to Figure 4.17, a small h_B can more correctly update the information tables of mobile nodes, reduce the chance of unnecessary re-clustering due to inaccurate or incorrect information from information tables, and then stabilize the cluster structure. However, a small h_B may trigger more periodic Hello messages. Table 4.5 shows the total number of periodic Hello messages and re-clustering messages for RCC and ECS with $v_{max} = 5\text{m/s}$ and 15m/s . It can be seen from Table 4.5 that O_c increases with the decrease of h_B . In conclusion, small h_B increases the cluster stability at the expense of large O_c .

Table 4.5 O_c vs. h_B with $N=500$ for RCC and ECS.

h_B	0.5s	1.0s	1.5s	2.0s
RCC, 5m/s	546431	286777	195327	148275
ECS, 5m/s	546111	286450	195074	146233
RCC, 15m/s	547401	287865	196893	149932
ECS, 15m/s	546612	287053	195795	148866

Figure 4.18 Clustering Status Distribution with $v_{max} = 10\text{m/s}$ and $N = 500$.

Figure 4.18 shows the distribution of mobile nodes with different clustering status for the three schemes. About 68% and 77% of mobile nodes in RCC and HCCM are member nodes covered by two or more CHs while only 57% of mobile nodes in ECS are involved in more than one cluster. It proves that ECS forms a lesser overlapping cluster structure. About 5% of mobile nodes in ECS are CGs. Although mobile nodes with CG status are only a small portion in ECS, it effectively helps reduce the number of clusters and reduce the cluster overlapping to form a stable cluster.

4.5 Conclusions

A clustering scheme, named ECS, for large and dense MANETs had been proposed and studied in this Chapter. By using a random CH claim mechanism, the frozen period requirement for cluster formation is eliminated in ECS. With the introduction of a new clustering status, named CG, the formation of small or unnecessary clusters can be avoided. The implementation of cluster deletion mechanism, together

with the introduction of CG, helps reduce cluster overlapping in ECS. In addition, ECS effectively improves the cluster stability by prolonging the CH lifetime with moderate clustering overheads. Simulation results show that ECS outperforms RCC and HCCM in terms of CH lifetime, clustering overheads and cluster number. The results show that ECS can perform well and outperform RCC and HCCM in a large mobile network environment. In conclusion, ECS successfully fulfills its targets at providing a more stable and less overlapping cluster structure with moderate clustering cost for a large and dense MANET. In addition, by evaluating the performance of RCC and ECS with different Hello frequencies, we are able to conclude that a proper ratio between *entryValidPeriod* of information table entries and Hello message interval is important for mobile nodes to maintain valid information tables, and for the clustering scheme to function properly.

Chapter 5

Implementation and Performance Evaluation of CBRP Based on ECS in MANETs

5.1 Introduction

As addressed in Chapter 2, 3 and 4, MANETs based on flat routing schemes, such as FSR [46], AODV [51] and DSR [17], cannot perform well when the network size increases, especially in face of node mobility as well, due to link and processing overhead [3-5, 19]. One way to solve this scalability problem is hierarchical routing. A typical way to build hierarchy is to group mobile nodes geographically near to each other into explicit clusters and assign different functionalities to mobile nodes inside and outside a cluster [3, 4]. In a cluster structure, mobile nodes may have different cluster-related status or function, such as CH, CM and CGW. A mobile node selected as a CH serves as the local coordinator for its cluster, and its ID is usually utilized to identify the corresponding cluster. A CGW is a non-CH node that either belongs to two or more different clusters or can directly connect to some non-CH node residing in a different cluster. A CM is a non-CH node of a cluster without any inter-cluster links [4]. Routing

based on this kind of cluster structure is considered as cluster based routing. With the cluster structure, the processing and spreading of routing information are restricted to partial mobile nodes in the network based on their cluster status. Mobile nodes that participate in routing in such cluster based routing are called routing backbone nodes, and normally CHs and CGWs serve as the routing backbone nodes [3, 4, 25, 26]. In other words, the cluster structure can help reduce the routing space (referring to the number of routing backbone nodes) and the routing overheads without affecting the routing efficiency [58]. Hence, cluster based routing should be able to provide better scalability solutions for MANETs compared to flat routing schemes.

A cluster based routing scheme consists of two major parts: the clustering algorithm and the routing algorithm. The clustering scheme discusses how to form and maintain a cluster structure in a dynamic MANET. The routing scheme discusses how to discover and maintain routes on the top of the cluster structure.

As the scalability becomes one of the most challenging issues for the real implementation of MANETs, a lot of researches put efforts on proposing clustering protocols and cluster based routing protocols [5, 13, 15, 23, 34, 36, 54] to provide efficient solutions for medium to large size MANETs. Two typical cluster based routing schemes, CGSR [54] and CBRP [15] are briefly described and discussed in Chapter 3. By comparing CGSR and CBRP, we found that CBRP is more promising due to its several attractive features: 1) A mobile node in CBRP only needs to maintain cluster-related information for its 1-hop and 2-hop neighbors. Thus, the information table size of CBRP is much smaller than that of CGSR. 2) The inter-cluster routing is based on on-demand mechanisms, which avoids the periodic routing information exchange between mobile nodes, and thus makes the routing information processed and stored by routing

backbone nodes less, compared to protocols utilizing proactive mechanisms for inter-cluster routing, such as CGSR. 3) IP loose source routing is utilized as the inter-cluster routing mechanism when a RREP travels back from the destination to the source, and the route discovered by the RREP will serve as the active route for subsequent data delivery. This can help reduce the length of discovered routes and release CHs from serving as data relays when possible. 4) Local repair mechanism based on the neighbor information at each mobile node can locally salvage a forwarded packet and repair a broken route. This mechanism functions well in preventing packet drop and discovering new routes to replace the stale ones with low routing cost.

In a cluster based routing scheme, the clustering algorithm should provide a stable cluster structure because frequent cluster change, especially frequent re-clustering, may generate large clustering overheads and adversely affect its upper layer routing, which highly depend on the cluster stability to function well [4, 13]. Also, a clustering algorithm should respond fast to the underlying network topology change because its upper layer routing protocols depend on its correctness and accuracy for routing and packet delivery. In addition, the routing algorithm should make full use of its underlying cluster structure to facilitate its function and the information contained in the routing events should be utilized to maintain the cluster structure. In other words, the clustering algorithm and the routing algorithm in a cluster based routing scheme should be mutual beneficial to each other and enhance the overall network performance together.

Hence, in this Chapter, we first implement the routing mechanisms of CBRP on the proposed clustering scheme in Chapter 4, named ECS [6]. Such a cluster based routing scheme is named ECBRP in this Chapter. ECS can provide a more stable cluster structure as compared to the clustering mechanisms used in CBRP by reducing the re-

clustering occurrence. Also, ECS provides a less overlapping cluster structure by introducing a new cluster status for mobile node, called CG, and a cluster deletion mechanism. Hence, ECS is more efficient in simplifying the network structure. In our implementation, some routing mechanisms in CBRP are modified in order to fully utilize the underlying cluster structure from ECS. A new information table update mechanism by utilizing routing events and data forwarding events to enhance the clustering and routing performance is introduced. Then, the complete routing performance of ECBRP in terms of packet delivery ratio, normalized routing overhead, end-to-end transmission delay and so on is compared with that of CBRP and DSR. By evaluating the performance of ECBRP and CBRP, we can study how a cluster based routing protocol and its corresponding performance is affected by its underlying cluster structure, including cluster stability and cluster overlapping. By comparing the performance between DSR and the two cluster based routing schemes, we can study how a cluster structure brings scalability and routing efficiency for a MANET as the network traffic load, node mobility or network size increases.

The remainder of this Chapter is organized as follows. Section II shows the detailed implementation of ECBRP. Section III describes the simulation models and parameters. In Section IV, the simulation results are presented and analyzed. Finally, a brief conclusion is drawn in Section V.

5.2 Implementation of ECBRP

In this Chapter, we have implemented ECBRP in NS-2 [80]. This Section presents a detailed description of the implementation for some routing-related mechanisms in ECBRP, including neighbor information discovery and maintenance,

route discovery and route maintenance. Even though the routing mechanisms of ECBRP are based on CBRP, some mechanisms are modified in order to make full use of the underlying cluster structure maintained by ECS. In addition, a new information table update mechanism is introduced in ECBRP to enhance the clustering and routing performance.

5.2.1 Neighbor Information Discovery and Maintenance

Figure 5.1Cluster Based MANET Topology.

Each mobile node periodically sends out Hello message to its 1-hop neighbors. The Hello message includes its own ID, node degree, cluster status (CH, CM, CGW, CG, or unspecified), CID(s) as well as each 1-hop neighbor's ID, node degree and cluster status. By the Hello exchange, each mobile node can obtain the complete information for all neighbors within two hops. Based on received Hello messages, a mobile node then can build its 1-hop NT and 2-hop NT. The 1-hop NT and 2-hop NT at node n_1 corresponding to Figure 5.1 are shown in Table 5.1 and Table 5.2 respectively. Other than the periodic Hello messages, mobile nodes will also send out triggered Hello messages, including CH claim and CH resign [6] to indicate re-clustering and cluster

structure change. These triggered Hello messages and ordinary periodic Hello messages are all treated as Hello in this Chapter.

Table 5.1: 1-hop NT of Node n_1 .

Node ID	Node Degree	Node Status	CID	EntryUpdateTime(s)
1	4	CM	3	N.A.
2	3	CGW	3, 4	10.5
3	5	CH	3	11.0
8	1	CG	N.A.	10.1
13	3	CM	3	11.8

Table 5.2: 2-hop NT of Node n_1 .

Node ID	Node Degree	Node Status	Next Hop	EntryUpdateTime(s)
2	3	CGW	3	11.0
3	5	CH	2	10.5
4	4	CH	2	10.5
5	4	CGW	3	11.0
6	4	CGW	3	11.0
6	4	CGW	13	11.8
13	3	CM	3	11.0

A mobile node knows its own CHs and its neighboring CHs based on the information listed in its 1-hop NT and 2-hop NT respectively. However, for a CH, its neighboring CHs could be 2 or 3 hops away. As shown in Figure 5.1, for CH n_4 , the distance between itself and its two neighboring CHs, n_3 and n_{10} , is 2-hop and 3-hop respectively. Based on the information included in the Hello message, a CH has no ways to know its 3-hop neighboring CH because the Hello can only provide information for neighbors within two hops. Hence, a CH has to rely on its member nodes' neighboring cluster information to discover its 3-hop neighboring CHs. In ECBRP, a CGW will

include its neighboring cluster information in the outgoing Hello messages, so that when its corresponding CHs receive the Hello, they can know their 3-hop neighboring CHs. For example, in Fig. 1, n_5 will indicate n_{10} as its neighboring CH in its outgoing Hello. When n_3 receives the Hello, it first checks its 2-hop NT and finds out that n_{10} is a 2-hop neighbor. So it ignores this information. However, when n_4 receives the Hello from n_5 and checks its 2-hop NT, it understands that n_{10} is not a 2-hop neighbor but a 3-hop neighboring CH, and it can reach n_{10} through n_5 . Hence, it records the information for n_{10} in its CAT. Here, CAT is a storage space for CHs to record the information for their 3-hop neighboring CHs. Based on the cluster structure illustrated in Figure 5.1, CAT at CH n_4 is shown as in Table 5.3.

Table 5.3: CAT of Node n_4 .

Cluster ID (CID)	Gateway ID (GID)	EntryUpdateTime(s)
10	5	11.2
10	9	10.6

A mobile node scans its 1-hop NT, 2-hop NT and CAT (if any) every *NTScanInterval* and deletes entries with elapsed time longer than a predefined *entryValidPeriod* since the last *entryUpdateTime*. The recorded information at these information tables can help dynamically maintain the cluster structure and routes for a MANET.

5.2.2 Route Discovery

Route discovery refers to mechanisms how a mobile node **S** wishing to send a data packet to a destination node **D** obtains a route to **D**. Like DSR and CBRP, routing in

ECBRP is based on source routing. Similar to many on-demand routing protocols or routing schemes with the on-demand component, the way **S** discovers a route to **D** in ECBRP is achieved by flooding. However, because of the underlying cluster structure, the flooding traffic and the number of mobile nodes involved in the route discovery in ECBRP can be greatly reduced [15].

When **S** attempts to send a data packet to **D**, it first checks its 1-hop NT. If **D** is found in the 1-hop NT, **S** sends the data packet directly. If **D** is not found in its 1-hop NT, **S** then checks its 2-hop NT. If **D** is found in the 2-hop NT and can be reached through more than one 1-hop neighbor nodes, it chooses the one with the most recent *EntryUpdateTime* as the intermediate node and sends the data packet along this source route. Refer to Figure 5.1, if **S**, n_1 , needs to send data packets to **D**, n_6 , after consulting Table 5.2, it will use n_{13} instead of n_3 as the relay. This is because the information for n_{13} is more recently updated and thus is guaranteed to be more correct. If **D** is not found in its 2-hop NT either, **S** then consults its RC. If a route to **D** is found in the RC, **S** simply uses it to send the data packet. Here, RC is a storage space in each mobile node for recording discovered routes to other mobile nodes [17].

5.2.2.1 Route Request

A RREQ packet is initiated to discover a route by **S**, if **S** cannot find any route to **D** in its NTs or RC. The RREQ header format is shown as Table 5.4. Before **S** broadcasts the RREQ, it always fills the neighboring CID (NCID) entries with its host CHs and neighboring CHs (from its information tables, such as NTs and CAT). The gateway ID (GID) entries refer to either the host CHs itself or the CGWs to reach neighboring CHs listed as the NCID entries [15]. Hence, GID and NCID always appear in pair in the

packet header of RREQ. *Num1* indicates how many such CH/CGW pairs listed in the RREQ header. RREQ identifier is a flag in the packet header to indicate that such a packet is a RREQ. RREQ sequence is a unique request ID determined by *S* to help identify the corresponding RREQ. More information about RREQ identifier and RREQ sequence can be referred to [17].

Table 5.4: Partial RREQ Header Format.

RREQ identifier	<i>Num1</i>	<i>Num2</i>	RREQ sequence
Source ID			
Destination ID			
Gateway ID (GID) [1]			
Neighboring CID (NCID) [1]			
...			
Gateway ID (GID) [<i>Num1</i>]			
Neighboring CID (NCID) [<i>Num1</i>]			
Traversed CID (TCID) [1]			
...			
Traversed CID (TCID) [<i>Num2</i>]			

When an intermediate node **IM** receives the RREQ and finds out that **D** is a 1-hop neighbor, **IM** then forwards (unicast) the RREQ to **D**. Otherwise, **IM** will decide how to process the RREQ based on its cluster status and the information displayed in the RREQ packet header as follows:

- If **IM** is a CM, a CG or with unspecified status, **IM** simply drops the RREQ. A CM is a mobile node without any inter-cluster links, and thus has no idea about how to forward a packet to neighboring clusters. Hence, it is not involved in the inter-clustering routing. An inter-cluster route that connects two neighboring CHs

through a CG is at least 4 hops. However to connect two neighboring CHs through CGWs is 2 or 3 hops. Since involving CGs for inter-cluster routing is not efficient, CGs are not supposed to handle RREQs in ECBRP. Since the cluster based routing utilizes the cluster structure for upper layer routing, an unspecified mobile node is not supposed to participate in routing because it does not involve in any cluster yet and thus has no clear picture of the underlying cluster structure.

- If **IM** is a CGW, it checks whether it is listed as a GID[n] entry in the RREQ packet header. If no, it simply drops the packet. Otherwise, it unicasts the RREQ to the corresponding NCID[n] as recorded in the header.
- If **IM** is a CH, and finds out that this RREQ is a copy of a previously processed one, it drops the RREQ packet. Otherwise, it appends its ID in the TCID sequence in the packet header and increases *Num2* by 1. Here, the TCID sequence in a routing packet's header shows the sequence of clusters that the packet has traveled or is about to travel [15]. *Num2* indicated the number of TCIDs listed in the packet header. If **D** is found to be a 2-hop neighbor, **IM** unicasts the RREQ to **D** based on its 2-hop NT. Otherwise, for each neighboring cluster (from its 2-hop NT and CAT), which is not listed in NCIDs and TCIDs, **IM** records the CH of each neighboring cluster and the corresponding CGW to reach that cluster to NCID/GID entry pair in the packet header, and then broadcasts the RREQ. Thus, *Num1* is increased by 1 accordingly. If no such neighboring cluster is found, it drops the RREQ.

For a cluster based routing, only partial mobile nodes, normally CHs and CGWs, are involved in routing by constructing the routing backbone. With the introduction of CG and the cluster deletion mechanism [6], the clustering scheme in ECBRP can form a

less overlapping cluster structure and have a higher portion of mobile nodes with CG or CM status, compared to other typical 1-hop clustering schemes, such as LCC, which is implemented as the underlying cluster algorithm in CBRP. Hence, the routing space in ECBRP can be further reduced and the routing traffic can be reduced accordingly.

5.2.2.2 Route Reply

Table 5.5: Partial RREP Header Format.

RREP identifier	Num1	Num2	RREQ sequence
Source ID			
Destination ID			
Traversing CID (TCID) [1]			
...			
Traversing CID (TCID) [Num1]			
Calculated Route ID (CRID) [1]			
...			
Calculated Route ID (CRID) [Num2]			

When **D** receives the RREQ, it returns a RREP to **S** as a reply, with the format shown in Table V. **D** copies the list of TCIDs from the received RREQ packet into RREP. And the reverse of this TCID list gives the complete information about the sequence of CHs that this outgoing RREP should travel in order to reach **S**. Since each CH knows how to reach its neighboring CHs by consulting its 2-hop NT and CAT, RREP will be routed to **S** eventually using IP loose source routing. The calculated route ID (CRID) list in Table 5.5 is a sequence of addresses of the hop-by-hop source route calculated by the traversing CHs (the TCID list) when they forward RREP back to **S**. **S** will use the route listed in CRID for subsequent data delivery. The RREP procedure is described as follows:

Initially, **D** will put its own ID in CRID[1] before sending out this RREP and set *Num2* to 1. When an **IM** receives a RREP, it will proceed as follows:

- When an **IM**, with CH status, receives the RREP packet, it checks whether its ID matches with TCID[*Num1*]. If not, the **IM** simply drops the RREP. Otherwise, it decrements *Num1* by 1 and delete itself from the TCID list. **IM** knows that the current TCID[*Num1*] is the next CH that should receive this RREP. Then, **IM** tries to find a CGW node **X** that can reach TCID[*Num1*] and CRID[*Num2*] directly based on its NTs. If **IM** finds **X** successfully, it sends the RREP to **X**. Otherwise, **IM** increments *Num2* by 1 and records its own ID into CRID[*Num2*]. The reason to find **X** is that the **IM** node, who is a CH, can be released from serving as a hop in the route for actual data delivery when possible. Thus, **X** will be used in the route for actual data delivery.
- When an **IM**, with CGW status, receives the RREP, it first checks its 1-hop NT and 2-hop NT to see whether it can reach TCID[*Num1*]. If not, it drops the packet. Otherwise, it increases *Num2* by 1 and records its own ID into CRID[*Num2*]. If TCID[*Num1*] is found to be a 1-hop neighbor, it sends the RREP to TCID[*Num1*] directly. If TCID[*Num1*] is a 2-hop neighbor and can be reached by a CGW node **X**, it sends the RREP to **X**.
- When **S** receives the RREP, it will copy the CRID list from the packet header to its RC, and use this CRID list as the actual route for data delivery.

For example, as shown in Figure 5.2, n_7 and n_{11} are **S** and **D** of a communication pair respectively. When n_{11} receives a RREQ packet originated from n_7 , indicating this RREQ has traveled through CHs n_4 and n_{10} , n_{11} will return a RREP to n_7 by traversing the CHs listed in the received RREQ with reverse order. Then, the reversed loose source

route for the RREP packet to travel from n_{11} to n_7 is $[n_{11}, n_{10}, n_4, n_7]$, where n_{10} and n_4 are listed as TCIDs in the RREP's header. As the RREP travels back, the strict source route, which will be utilized for the data forwarding, is computed based on the mechanisms addressed above and recorded in the CRID list in RREP's header. Then n_7 can obtain the actual source route, $[n_{11}, n_{10}, n_{12}, n_9, n_7]$, where $(n_{11}, n_{10}, n_{12}, n_9)$ is stored as the CRID list in RREP's header, once it receives the RREP.

Figure 5.2 Illustration of the RREP mechanism.

While forwarding the RREP, traversed CHs can calculate an optimized hop-by-hop route. And this mechanism makes ECBRP find shorter routes in terms of hops and release CHs from serving as data relays when possible, as compared to other cluster based routing protocols, such as CGSR. Hence, the chances that CHs become network bottlenecks due to traffic overloading in ECBRP should be reduced.

5.2.3 Route Maintenance

Route maintenance discusses issues about how to repair a broken route and how to make a route more optimal when the underlying network topology changes. The basic

route maintenance mechanisms in ECBRP follow DSR. We also implement the local route repair mechanism based on CBRP. In addition, we introduce an information table update mechanism to achieve information update with low cost and thus to enhance the routing performance.

5.2.3.1 Basic Route Maintenance Mechanisms

The basic route maintenance mechanisms are described as follows:

- A RERR message is generated at the upstream node of a broken link along an active route and sent back to **S** if the local salvage (addressed later) is not successful. Hence **S** can use an alternative route, if any, recorded in its RC or discover a new route for subsequent data delivery.
- When **S** decides to re-discover a route to its **D** upon receiving a RERR, it piggybacks the broken link information in the outgoing RREQ. Thus, stale routing information in the RCs of mobile nodes around **S** will not generate any RREPs that contain the same broken link.
- When a mobile node **IM** overhears a packet carrying a source route, it examines the route's unused portion. If **IM** is not the intended next hop but is listed in the later unused portion of the packet's source route, which implies that the intermediate nodes before itself along the source route are unnecessary [17]. Then **IM** returns a *gratuitous* RREP back to **S**, with the shortened route recorded in the packet header.

5.2.3.2 Packet Salvage Using Local Route Repair

In ECBRP, when a forwarding mobile node detects a broken link along an active route, it will manage to salvage the data packet by utilizing the local route repair mechanism as follows:

- 1) It first checks its 2-hop NT to find out whether it has any 1-hop neighbor, other than the downstream node of the broken link, which is directly connected to the hop after next along the source route. If more than one 1-hop neighbor can satisfy this requirement, it chooses the one with the most recent *EntryUpdateTime*. For example, as show in Figure 5.3(a), when n_7 delivers packets to n_6 using route $[n_7, n_4, n_5, n_6]$ and finds out that its next hop n_4 become unreachable, it uses n_9 to replace n_4 because n_9 can directly connect to n_5 .
- 2) If no such 1-hop neighbor is found, it consults its 2-hop NT to check whether the downstream node of the broken link can be reached through another mobile node. If more than one such mobile node is found, it uses the one with the most recent *EntryUpdateTime*. For example, as shown in Figure 5.3(b), when n_8 delivers packets to n_5 using route $[n_8, n_1, n_3, n_5]$ and the link between n_3 and n_5 is broken, it uses a new link $n_3-n_6-n_5$ to replace the stale link n_3-n_5 so that the source route can be locally adjusted.
- 3) If the local salvage is successful, a *LS* flag is set in the data packet header, indicating this packet is saved by the local salvage mechanism and the source route has been locally repaired. When **D** receives such a data packet with *LS* flag, it sends a gratuitous RREP to the corresponding **S** by setting a *G* flag and piggybacking the repaired route and the broken link of the original route in the RREP header. As **S** receives the RREP, it can delete source routes containing the

broken link in its RC, copy the repaired route into its RC, and use the repaired route for data delivery if necessary.

Figure 5.3 Illustration of the Local Route Repair Mechanism.

As addressed above, this local repair mechanism is inherited from CBRP. Since the second type of local route repair will increase the hop length of the source route and increase the relay traffic load for mobile nodes as compared to the first one, ECBRP treats the two mechanisms differently. If a route with broken link is repaired by mechanism 1), the *LS* flag in the salvaged data packet header will be set to 1, and no RERR message is generated. Hence, when **D** receives a data packet with *LS*=1, it returns a gratuitous RREP to **S**. When **S** receives such a RREP, it can use the repaired route for packet delivery. However, if a route is locally repaired by mechanism 2), the *LS* flag in the salvaged packet is set to 2, and a RERR is sent back to **S**. When **D** receives a packet with *LS*=2, it does not return a RREP to **S**. When **S** receives the RERR message, it can use alternative routes in its RC, if any, or actively re-discover a route to **D**. With this modification, it can be guaranteed that the route length is always an important metric for selecting active routes for packet delivery.

5.2.3.3 Information Table Updates Using Routing Packets and Data

Packets

The correctness and accuracy of information tables are quite important because they can reflect the real-time underlying network topology, which is the basis for clustering, routing and data delivery. Besides the explicit Hello messages, ECBRP utilizes data packets and routing packets for updating information tables at each mobile node so that the information tables can be updated more frequently as a byproduct when a mobile node receives packets from others. There are two solutions to utilize the data packets or routing packets to update the information tables:

- In ECBRP, a mobile node attaches its cluster-related information, including its cluster status, node degree and CIDs in an outgoing data packet or routing packet, and this information may travel up to two hops. Hence, the information about this mobile node in its neighbors' information tables can be updated when its neighbors receive or overhear such a packet. This update can be achieved at low cost because a mobile node adds only several bytes in the header of its outgoing packets to include the corresponding information.
- When an **IM** detects the link breakage between itself and its downstream mobile node along an active path, it deletes the entry corresponding to the downstream node from its 1-hop NT, and all entries with the downstream node as the "next hop" from its 2-hop NT. When a RERR message, indicating a broken link along an active path, travels back to **S** of the path, each mobile node, receiving or overhearing the message, will check whether its 2-hop NT includes an entry, with the upstream node of the broken link indicated in the RERR as the "node ID" and

the downstream node as the “next hop” or vice versa. If true, it will delete the corresponding entries from its 2-hop NT. Hence, the link break information can help to update information tables at a mobile node without adding any additional information to the RERR packet.

ECBRP makes full use of the information included in all kinds of packets to update the recorded information at a mobile node, so that the information tables can be maintained with more accuracy to better reflect the network and cluster topology change. Accordingly, the upper layer routing can be better served because of the information about neighbors and clusters at a mobile node is more accurate.

5.3 Simulation Models and Performance Metrics

5.3.1 Simulation Models

The NS-2 simulator, with wireless extension [80] is used for simulating the performance of four routing schemes: ECBRP, CBRP, DSR and AODV. NS-2 can simulate the physical, MAC and data link layer of a multi-hop wireless network. The DCF of IEEE 802.11 for wireless LANs is utilized as the MAC layer [80, 81]. The random waypoint model [82] is adopted for simulating movement behaviors of all mobile nodes in the MANET. Table 5.6 shows the default parametric values used in the simulations.

Table 5.6: Default Simulation Parameters.

Parameters	Values
Simulation Time	600s
Number of Mobile Nodes, N	50
Simulation Area, A_S	1000m × 1000m

Transmission Range for Mobile Nodes	250m
Transmission Data Rate	2Mbps
Mobility Model	Random Waypoint
Pause Time for Mobile Nodes	50.0s
Max. Speed for Mobile Nodes, v_{max}	10.0m/s
Traffic Pairs, n_t	15
Data Generation Rate at Source Nodes	4pkts/s
Packet Size of Data Packets Generated at Upper layer, L_p	512Bytes
Buffer Size for Queuing Outgoing Packets	50
Hello Message Interval, h_B	2.0s (for ECBRP and CBRP)
<i>NTScanInterval</i>	0.2s (for ECBRP and CBRP)
<i>entryValidPeriod</i>	4.0s (for ECBRP and CBRP)

Simulation results presented in the next Section are based on different mobility speeds v_{max} , different network traffic loads in terms of number of traffic pairs, n_t , and different network size in terms of number of mobile nodes, N . Data generation rate at source nodes refers to the number of data packets generated at each source node per second. L_p is the size of data packets generated at the upper application layer, and when such a data packet passed to the IP layer (network routing layer) for delivery, the necessary IP header, such as the IP address of source and destination, and the specified source routes, will be added to the packet And when the corresponding encapsulated IP packet is delivered to the MAC layer, the necessary MAC header will be furthered added to the packet, such as the MAC address of the current node and the next hop. Hence, when the data packet with size of L_p is transmitted on the wireless medium, its size is larger than L_p , and such header adding issue is considered and handled in NS-2. *NTScanInterval* decides how often a mobile node scans its information tables to delete obsolete entries in ECBRP and CBRP. *entryValidPeriod* shows the maximum time that

an information entry can reside in the information tables of a mobile node without being updated in ECBRP and CBRP. *Hello Message Interval* h_b reflects how often a mobile node sends out periodic Hello messages in ECBRP and CBRP.

5.3.2 Performance Metrics

The performance of ECBRP is compared with CBRP and DSR by the following performance metrics: i) data packet delivery ratio, r_p ; ii) normalized routing overheads, NO_v , and iii) end-to-end transmission delay, d_t .

r_p is defined as the total number of data packets received by all destination nodes over the total number of data packets sent by all source nodes in the network. NO_v refers to the total number of non-data packets transmitted at the IP layer over the total data packets received during the simulation. Each transmission of a non-data IP layer packet from one hop to another hop is counted as one packet [81]. The non-data packets in DSR include all kinds of routing packets, such as RREQ and RREP, whereas the non-data packet in ECBRP and CBRP includes the routing packets plus the Hello message. d_t calculates the average time from a data packet is generated at the source node till this data packet is received at the destination node.

5.4 Simulation Results and Discussions

In this Section, the performance of ECBRP, CBRP and DSR based on different simulation parameters combination is studied. The simulation result of each point on the plotted figures is the average over 10 random simulations following the specific mobility and traffic parameters. We also study the performance of AODV [51] in Section 5.4.1 in order to have more comparisons with ECBRP and more thorough judgment about

ECBRP. In addition, we show the performance of ECBRP without the mechanisms addressed in Section 5.2.3.3 in some of the simulations, and this version of ECBRP is called ECBRP-S. The main difference between ECBRP-S and CBRP is that ECBRP-S introduces a new cluster status, CG, and cluster deletion mechanism in its underlying clustering scheme. As a result, ECBRP-S can maintain a more stable and a less overlapping cluster structure and consequently involves fewer mobile nodes in the routing backbone as compared to CBRP. By comparing the performance between ECBRP-S and CBRP, we can study the impact of a cluster based routing scheme making full use of its underlying cluster structure on the network performance. By comparing the performance of ECBRP and ECBRP-S, we can study the performance improvement from routing mechanisms introduced in Section 5.2.3.3.

5.4.1 Effect of Traffic Load

Figure 5.4-Figure 5.7 show the simulation results in terms of r_p , NO_v and d_t for ECBRP, ECBRP-S, CBRP, DSR and AODV with different n_t .

5.4.1.1 Packet Delivery Ratio

Figure 5.4 shows that r_p drops for all the 5 schemes with n_t . ECBRP and ECBRP-S perform better than CBRP because they generate a routing backbone with smaller size due to ECS, which effectively reduces the routing overheads without impairing the routing efficiency. The routing overheads reduction in ECBRP and ECBRP-S can be seen from Figure 5.5 and Figure 5.6. With fewer routing overheads in ECBRP and ECBRP-S, mobile nodes are more lightly overloaded, especially in a network with heavy traffic load, as compared to those in the network with CBRP. Hence, fewer packets are

dropped due to queue size full of mobile nodes for ECBRP or ECBRP-S as shown in Table 5.7. Consequently, more packets can be delivered for ECBRP and ECBRP-S.

Figure 5.4 r_p vs. n_t with $N=50$ and $v_{max}=10m/s$.

Table 5.7: Number of Dropped Packets at IP Level Due to Queue Size Full of Mobile Nodes.

	20 traffic pairs	25 traffic pairs
CBRP	2731	8994
ECBRP	380	5117
ECBRP-S	509	5645

ECBRP can outperform ECBRP-S because ECBRP utilizes data packets and routing packets for updating the information tables of mobile nodes. Thus, the neighbor information and cluster information recorded at mobile nodes become more correct and accurate in ECBRP. Hence, ECBRP can rely on more reliable information for routing and consequently has a higher packet delivery ratio as compared to ECBRP-S. It can be seen that the performance improvement of the three cluster based routing schemes over

DSR increases with n_t . DSR aggressively utilizes route caching mechanism and greatly depends on the information in RCs for route discovery and route repair. The extend of caching is too large to benefit the performance with increased network load [83]. Stale routes are often chosen from the RCs because there is no time expiry mechanism for managing the routes recorded in RCs [83]. Therefore, large routing overheads are generated to repair the stale routes or re-discover new routes. Consequently, the network becomes highly congested and packets are likely dropped due to no valid routes. Hence, r_p of DSR is adversely affected with large n_t . However, ECBRP, ECBRP-S and CBRP mainly depend on information tables of mobile nodes for routing and such information is more reliable due to the periodic update of information tables. Thus, the three cluster based routing schemes can have better r_p , especially for high data traffic load. In addition, AODV performs well as n_t increases. Referred to Section 3.2.1.2.2 of Chapter 3, AODV has some kind of timeout mechanisms for recorded routes in the routing tables of mobile nodes, and this mechanism is effective in guaranteeing the validity of cached routes. Hence, packets can be more guaranteed for delivery by utilizing the recorded routes in AODV, especially in the network with heavy data traffic load.

5.4.1.2 Routing Overheads

Figure 5.5 shows NO_v excluding the Hello messages with n_t for ECBRP, ECBRP-S, CBRP, DSR and AODV, and we denote it as NO_v-H . As the traffic load increases, mobile nodes become overloaded and wireless channels are over-utilized, and thus packets are likely dropped due to queue full at mobile nodes or transmission collision in space. Hence, the delivery of a data packet may require more routing packets sent, and NO_v-H increases with n_t for all the five schemes. AODV generates the largest NO_v-H as

compared to the other four schemes. This is probably because AODV does not cache overheard routes as DSR, CBRP and ECBRP, and hence less likely finds a route in the routing table of a mobile node for packet delivery. Thus, AODV needs to generate more RREQs to actively discover routes. Also, AODV treats RERRs as broadcast messages while the other compared schemes treat RERRs as unicast messages, and hence AODV likely has more routing packets in terms of RERRs [83]. Surprisingly, the large NO_v-H of AODV does not deteriorate its performance of r_p greatly as n_t increases. This is because probably up to 90% of routing overheads of AODV are broadcast packets [83]. When a unicast packet delivered from the routing layer to the MAC layer, it always requires the RTS-CTS-packet-Acknowledgement procedure for delivering such a unicast packet. In case of transmission failure, it can repeat up to 7 times of RTS retransmission and 4 times of packet retransmission to guarantee the packet delivery. On the contrary, when a broadcast packet delivered to the MAC layer, it will be directly sent out simply by piggybacking the necessary MAC header in the packet. Thus, broadcast packets invoke much fewer MAC overheads as compared to unicast packets, and the MAC overheads reflect the actual network overheads [83]. Consequently, the large NO_v-H of AODV does not greatly drag down its r_p due to its broadcast nature. As n_t increases, DSR generates significantly larger NO_v-H as compared to the three cluster based routing schemes. This is mainly because stale routes are often chosen from RCs in DSR for packet delivery, and hence large routing overheads are invoked to repair the stale route or re-discover new routes to replace the stale ones. ECBRP and ECBRP-S can outperform CBRP because ECS providing a less overlapping cluster structure with fewer mobile nodes serve as CHs and CGWs. Hence the routing space in ECBRP and ECBRP-S is smaller than that of CBRP, and their routing overheads are reduced accordingly.

Figure 5.5: NO_v-H vs. n_t with $N=50$ and $v_{max}=10m/s$.

Figure 5.6 shows NO_v with n_t for the five schemes. When $n_t=10$, the routing overheads generated due to route discovery and maintenance are comparatively low for all the schemes, and the Hello messages of the three cluster based routing schemes count a comparatively larger portion in NO_v . Hence, CBRP, ECBRP and ECBRP-S show higher NO_v than DSR and AODV. As n_t increases, routing overheads invoked for route discovery and maintenance increase dramatically for all the schemes, and the Hello portion in NO_v becomes relatively small for the three cluster based routing schemes. As shown in Figure 5.5, since CBRP, ECBRP and ECBRP-S can generate less NO_v-H , their NO_v increases more slowly with n_t as compared to DSR and AODV. It can be seen that NO_v of ECBRP and ECBRP-S is lower than that of CBRP because they three generate similar number of Hello messages, but NO_v-H of ECBRP and ECBRP-S is lower

Figure 5.6: NO_v vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$.

5.4.1.3 End-to-End Delay

Figure 5.7 shows that d_t increases with n_t for the five schemes. As the traffic load increases, the wireless channels become more crowded, and packets are likely accumulated at mobile nodes before they can be sent out. Hence, the packet queuing delay at mobile nodes increases, resulting in longer d_t . d_t of DSR increases faster with n_t as compared to that of the other four schemes. This is because when n_t is high, the network with DSR is more congested due to the significant increase in routing overheads, especially the increase of unicast routing packets, as shown in Figure 5.6, and hence adversely affects d_t of DSR. Various buffering and queuing delays and time to gain access to the radio medium in a single congested node are often very large compared to the same delays in other nodes in a multi-hop route [83]. In AODV the destination node

only replies to the first arrived RREQ, indicating AODV favors the least congested route instead of shortest route. However, DSR replies to all received RREQs and this makes it difficult to decide the least congested route for subsequent route discovery [83]. CBRP and ECBRP use a routing backbone composed with a small set of mobile nodes for route discovery and data delivery, and hence these mobile nodes likely become congested due to over utilization. Hence, in a scenario with heavy traffic load, compared to the other four schemes, AODV has higher chance to choose less congested routes for data delivery and shows better performance in terms of d_t consequently. ECBRP and ECBRP-S generate lower d_t than CBRP because they can generate fewer routing overheads mainly resulting from the less overlapping cluster structure maintained by ECS.

Figure 5.7 d_t vs. n_t with $N=50$ and $v_{max}=10\text{m/s}$.

5.4.2 Effect of Node Mobility

Figure 5.8-Figure 5.10 show the performance of ECBRP, CBRP and DSR with different v_{max} . ECBRP-S is not included for comparison because it has been demonstrated in Section 5.4.1 that ECBRP can outperform ECBRP-S in various routing metrics.

5.4.2.1 Packet Delivery Ratio

Figure 5.8 shows that r_p decreases with v_{max} for ECBRP, CBRP and DSR. This is because communication paths are more likely to be broken in a highly mobile network and thus adversely affects the data delivery. It can be seen that ECBRP can tolerate the node mobility better than CBRP because r_p drops only about 3% for ECBRP while over 6% for CBRP when v_{max} increases from 5m/s to 20m/s. This is due to two reasons: 1) ECBRP maintains a less overlapping and more stable cluster structure because of ECS. 2) The information recorded at each mobile node is more correct and accurate in ECBRP as compared to that of CBRP because ECBRP utilizes all types of packets to update information tables of mobile nodes. Benefiting from the more stable underlying cluster structure and the more reliable cluster and neighbor information, ECBRP can thus deliver more data packets. Table 5.8 shows the number of triggered Hello messages for CBRP and ECBRP. Triggered Hello is related with CH change, which adversely affects the cluster topology and cluster stability. It can be seen that ECBRP generates fewer triggered Hello messages than CBRP. And that is a good indication that ECBRP can provide better cluster stability.

Figure 5.8 r_p vs. v_{max} with $n_t=15$ and $N=50$.

Table 5.8: Number of Triggered Hello Messages.

	10m/s	20m/s
CBRP	264	473
ECBRP	95	153

CBRP performs better than DSR when v_{max} is low to moderate because a comparatively low v_{max} does not have much impact on r_p for all the three schemes. In such a scenario, CBRP is demonstrated to be a more efficient routing scheme as shown in Figure 5.4. However, when v_{max} becomes comparatively high, the cluster structure of CBRP changes frequently and adversely affects its upper layer routing and packet delivery. Thus, r_p drops significantly for CBRP at $v_{max}=20\text{m/s}$.

5.4.2.2 Routing Overheads

Figure 5.9 NO_v vs. v_{max} with $n_t=15$ and $N=50$.

Figure 5.9 shows that NO_v for ECBRP, CBRP and DSR increases with v_{max} . This is because with more routes broken in a highly mobile environment, the routing schemes need to initiate more routing packets to establish and maintain valid routes. DSR generates lower NO_v than ECBRP and CBRP. It is because when $n_t=15$, indicating the network traffic load is moderate, Hello messages count a significant part in the total routing overheads of ECBRP and CBRP. DSR, without triggering any Hello messages, then can have lower NO_v than the other two schemes. ECBRP always has lower NO_v than CBRP. By maintaining a routing backbone with smaller size because of ECS, ECBRP can generate fewer routing overheads. Also, ECS maintains a more stable cluster structure, and hence the routing overheads due to route invalidation resulting from the underlying cluster structure change or information inaccuracy are smaller for ECBRP.

5.4.2.3 End-to-End Delay

Figure 5.10 d_t vs. v_{max} with $n_t=15$ and $N=50$.

As explained for Figure 5.7, d_t is highly related with network traffic load. As v_{max} increases, the total network load increases due to larger routing overheads as shown in Figure 5.9, and hence d_t increases with v_{max} for all the three schemes. Since DSR generates the lowest NO_t at all speeds, it has the lowest d_t . Similarly, CBRP generates the highest NO_t , and thus it has the highest d_t .

5.4.3 Effect of Network Size

Figure 5.11-Figure 5.13 show the performance of ECBRP, CBRP and DSR with different network size in terms of N , while the network density in terms of N/A_s is kept constant.

5.4.3.1 Packet Delivery Ratio

Figure 5.11 r_p vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$.

Figure 5.11 shows that r_p decreases with N for ECBRP, CBRP and DSR. As N increases, a routing packet will be handled by more mobile nodes and thus brings significantly larger routing overheads for the network. Also, to keep N/A_s constant, A_s increases with N , and hence the average route length in terms of hops for a communication pair increases. This results in more relay loads for mobile nodes and generates heavier data traffic load for the network. Besides, longer routes likely increase the packet dropping probability. All these explain that r_p drops with N . r_p of DSR drops much faster with N . As we know, the routing overheads are highly related with the routing space. The routing space of DSR increases with N whereas that of ECBRP and CBRP increases with the number of CHs and CGWs, which is smaller than N . Thus, the network with DSR is much worse congested as N increases. Besides, the increase in

route length between a communication pair brings more difficulty in route discovery and route maintenance. ECBRP and CBRP mainly depend on information tables for routing and show more routing efficiency than DSR because DSR highly depends on RCs for routing as explained in Figure 5.4.

5.4.3.2 Routing Overheads

Figure 5.12 shows that NO_v increases with N for ECBRP, CBRP and DSR. NO_v of DSR increases much faster than that of ECBRP and CBRP as explained for Figure 5.11. It can be seen that ECBRP can effectively reduce the routing overheads as compared to CBRP because it generates a smaller routing space and brings more routing efficiency.

Figure 5.12 NO_v vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$.

5.4.3.3 End-to-End Delay

Figure 5.13 shows that d_t increases with N for the three schemes. d_t of DSR increases much faster because its routing overheads increase dramatically with N , resulting in a network with much worse congestion as compared to ECBRP and CBRP. The results in this part show that the cluster based routing can provide better scalability in terms of network size. Besides, ECBRP can provide satisfying performances as N increases. Hence, it can be concluded that ECBRP has better scalability than CBRP and DSR.

Figure 5.13 d_t vs. N with $n_t=15$ and $v_{max}=10\text{m/s}$.

5.4.4 Effect of Hello Frequency on ECBRP

In this section, the performance of ECBRP with different Hello frequencies is presented and discussed. And the Hello frequency is represented by the Hello Message

Interval h_B as shown in Table 5.6, and different h_B of 1s, 2s or 3s is utilized for simulations. Hence, the corresponding Hello frequency is represented as 1/1s, 1/2s and 1/3s. As h_B is varied to study the effect of Hello frequency, *entryValidPeriod* as shown in Table 5.6 is adjusted accordingly to keep the ratio of *entryValidPeriod* and h_B to be constant, with the value of 2, always.

5.4.4.1 Number of RERR Messages

Figure 5.14 shows the number of RERRs generated with v_{max} for ECBRP running with different Hello frequencies when $n_t = 15$. As described in Section 5.2.3, the RERR message is generated when a unicast message cannot be delivered to its next hop as indicated in the source route and no other replacing node can be found to locally repair the invalid route. It can be found that the number of RERR messages is increased with v_{max} . This is because when the network becomes more mobile, the active routes are likely broken due to node mobility, and hence the RERRs generated to report route invalidation becomes more. In addition, it can be seen that the higher the Hello frequency, the less number of RERRs, and ECBRP with Hello frequency of 1/1s generates fewest RERRs. In ECBRP, when the Hello messages are sent with higher frequency, the NTs maintained at mobile nodes are with higher accuracy. Correspondingly, the routes discovered or locally repaired by the NTs are more correct and valid, and hence the RERRs generated due to route invalidity are fewer.

Figure 5.14 Number of RERRs vs. v_{max} with $n_t=15$.

Figure 5.15 shows the number of RERRs generated with v_{max} for ECBRP with different Hello frequencies when $n_t=20$. It can be seen that ECBRP with Hello frequency of 1/1s generates fewer RERR messages as compared to the other two for all simulated mobility. And the reason can be referred to the explanation for Figure 5.14. However, by comparing Figure 5.14 and Figure 5.15, it can be found that the number of RERRs is much larger for $n_t=20$. This is because with more communication pairs in the network, more active routes need to be maintained, and hence RERRs due to route invalidity are more likely to be generated.

Figure 5.15 Number of RERRs vs. v_{max} with $n_t=20$.

5.4.4.2 Packet Delivery Ratio

Figure 5.16 shows that r_p drops with v_{max} for ECBRP when $n_t=15$ and the explanation can be referred to Figure 5.8. It can be seen that as v_{max} increases from 5m/s to 20m/s, r_p with Hello frequency of 1/3s drops the fastest, followed by r_p with Hello frequency of 1/2s, and r_p with Hello frequency of 1/1s drop most slowly, although the difference is not with much significance. This is probably because as the network becomes more mobile, the route invalidity due to node mobility becomes more frequent, and hence the NT accuracy becomes more important since ECBRP often depends on NTs for route discovery and repair. As NTs are mainly maintained by Hello messages, and the more frequent Hello messages are exchanged between neighboring nodes, the more accurate of NTs maintained at mobile nodes. Hence, ECBRP with Hello frequency of 1/1s performs the best when the network mobility is high.

Figure 5.16 r_p vs. v_{max} with $n_t=15$.

Figure 5.17 r_p vs. v_{max} with $n_t=20$.

Figure 5.17 shows r_p with v_{max} for ECBRP with different Hello frequencies when $n_t=20$. It can be seen that when v_{max} is comparatively low, corresponding to 5m/s and 10m/s, ECBRP with Hello frequency of 1/1s performs the worst. $n_t=20$ indicates the network is somewhat overloaded with the data traffic, and when the Hello message is sent with comparatively high frequency of 1/1s as compared to the Hello frequency of 1/2s and 1/3s, the network is more overloaded due to these “Hello” routing overheads, and hence r_p is adversely affected. However, as v_{max} increases to 15m/s and 20m/s, ECBRP with Hello frequency of 1/3s drops faster than the other two. This is probably because when v_{max} is comparatively high, the accuracy of NTs is more likely affected by the Hello frequency because the network topology is changing fast as compared to small v_{max} . Hence, when Hello is sent with low frequency, correspondingly to 1/3s, the correctness and accuracy of NTs is greatly affected. As explained before, the performance of ECBRP highly depends on the NT accuracy, and hence r_p of ECBRP with Hello frequency of 1/3s drops the fastest when the network is highly mobile.

5.4.4.3 End-to-End Delay

Figure 5.18 shows that d_t performs the best with Hello frequency = 1/1s when $n_t=15$. This is probably because the higher frequency of Hello resulting in the more accuracy of NTs at mobile nodes, and hence packets can be delivered to the destinations with fewer errors. And the delivery errors normally require packet retransmission or delivery route re-discovery, and hence adversely affect d_t .

Figure 5.18 d_t vs. v_{max} with $n_t=15$.

Figure 5.19 shows the performance of d_t for ECBRP with different Hello frequency when $n_t=20$. Different from Figure 5.18, ECBRP with Hello frequency of 1/2s has the best performance of d_t in Figure 5.19. Referring to the discussion for Figure 5.7, d_t is highly related with network load, especially when the network is overloaded, and $n_t=20$ indicates the network is somewhat overloaded by the data traffic. When ECBRP utilizes a moderate Hello frequency of 1/2s, it can generate fewer Hello messages as compared to that with Hello frequency of 1/1s, and produce fewer routing packets for route maintenance as compared to that with Hello frequency of 1/3s. Consequently, this moderate Hello frequency piggybacks the fewest routing overheads into the network, and hence achieves performance of d_t in such an overloaded network.

Figure 5.19 d_t vs. v_{max} with $n_t=20$.

5.5 Conclusions

This Chapter deploys main routing mechanisms in CBRP on ECS to implement a cluster based routing scheme. Some routing mechanisms in CBRP are modified in order to fully utilize the underlying cluster structure maintained by ECS. Also, a new information table update mechanism is proposed by utilizing routing events and data forwarding events to enhance the clustering and routing performance. Such a cluster based routing scheme is named ECBRP in this Chapter. Due to the more stable and less overlapping cluster structure provided by ECS and the newly proposed table update mechanism, ECBRP generates a routing backbone with smaller size and providing better routing efficiency. As a result, ECBRP can outperform CBRP and DSR in terms of

packet delivery ratio, normalized routing overhead and end-to-end transmission delay in various simulated scenarios.

By comparing the performance between DSR and the two cluster based routing schemes, ECBRP and CBRP, we can conclude that a cluster structure bring scalability and routing efficiency for a MANET as the network traffic load or network size increases. By partially comparing the performance of DSR and AODV, we believe that some timeout mechanism for recorded routes to guarantee the freshness of RCs/routing tables is important for route validity and packet delivery. Also, unicast packets more likely congest the network and deteriorate the throughput performance as compared to broadcast packets because they invoke much more MAC overheads in the network. By evaluating the performance of ECBRP and CBRP, we can conclude that the performance of a cluster based routing scheme highly depends on its underlying cluster structure. A more stable cluster structure brings efficiency in route discovery and maintenance whereas a less overlapping cluster structure brings efficiency in routing overheads reduction. Hence, ECBRP can outperform CBRP as the network traffic load, node mobility or network size increases. The performance of ECBRP highly depends on its underlying cluster structure, and the cluster structure is mainly maintained by the periodic Hello mechanism. Hence, the performance of ECBRP with different Hello frequencies is studied, and from the simulation results, it can be concluded that an appropriate and moderate Hello frequency is important to achieve good performance, especially when the network is overloaded with data traffic.

Chapter 6

HEAR: Hybrid Energy-Aware Routing for Energy-Limited MANETs

6.1 Introduction

Mobile nodes in a MANET are normally based on battery energy supply and cannot be recharged during operation. Thus, energy limitation poses a severe challenge for the performance of a MANET. Dynamic routing is a basic but important event in MANETs so that energy metrics should be taken into consideration for guaranteeing the routing performance. As addressed in Chapter 3, a lot of researches have been done to improve the energy efficiency of MANET routing. The first category introduces the sleep mode to minimize the idle state energy consumption of mobile nodes [8, 58]. The second category utilizes multiple power levels to search a path that consumes minimum transmission energy between a source-destination pair [42, 59, 60, 84]. The third category reasonably makes use of each mobile node as data relay based on their energy levels to avoid network disconnection and service interruption due to the early energy depletion of partial mobile nodes, and thus to guarantee the routing-related performance

in an energy-limited MANET [9, 10, 18, 57, 61]. Our proposed HEAR algorithm belongs to the last category.

Network throughput is an important metric to evaluate the performance of a data communication network. For an energy-limited network, such as a MANET, its throughput is greatly related to the lifetime of each mobile node. The early “death” of some mobile nodes may cause severe problems, such as network partitioning and communication interruption [9]. Hence, it is mandatory to prolong the network operation time in order to maximize the network throughput [9, 85]. The natural way to realize this objective is to choose mobile nodes with higher energy level to constitute the active paths for relaying data between source-destination pairs if possible. Thus, mobile nodes with poor energy level can use their energy in more critical cases, such as transmitting and receiving data packets for themselves, or serving as data relays for a specific communication pair that builds the only route through them.

Previous researches add energy-related mechanisms into some on-demand routing protocols, and develop energy-aware routing protocols. Table-driven routing protocols are considered unfeasible for an energy-limited MANET because each mobile node needs to regularly exchange its routing table with neighbors or broadcast its routing-related information from time to time. This causes large amounts of unnecessary routing overheads and makes mobile nodes drain their battery energy fast. Hybrid routing schemes have not been considered for developing an energy-aware routing protocol in previous researches either because of its table-driven routing component.

Several energy-aware routing schemes, such as min-max battery cost routing (MMBCR) [10], CMMBCR [9] and minimum drain rate (MDR) [57], utilize the route discovery mechanism in the basic on-demand MANET routing to exclude energy-poor

nodes from constituting the communication routes between source-destination pairs. Before an intermediate node forwards a RREQ, it attaches not only its ID but also its current energy level information to the RREQ. Thus, the destination can determine the energy level of each path included in these RREQs and choose the best path according to the given energy metrics. In such schemes, however, a destination node cannot return a RREP message back immediately until it collects all the possible RREQs in order to decide the best energy route among them. Unfortunately it is difficult for a destination to estimate how long it would need to wait before selecting the best route without global information, such as network density and system traffic load [61]. A short waiting period may make some better paths in terms of energy not be considered. But a long waiting period may greatly affect the route discovery time. Some energy-aware routing schemes, such as local energy-aware routing (LEAR) [61] sets an energy threshold for each mobile node to decide whether it should participate in the RREQ forwarding procedure. Only when the residual energy level of a mobile node is higher than the preset threshold, the node is allowed to forward RREQs for other nodes. Otherwise, it simply drops them. This mechanism excludes some energy-poor mobile nodes from serving as data relays between any communication pair, which may be indispensable for building a valid route between a source-destination pair. Since the energy level for all mobile nodes drops with simulation time, the threshold should be adjusted accordingly. However, the energy drain rate for mobile nodes is highly dependant on the system traffic load, movement pattern of mobile nodes and network density. Without such information, the dynamic adjustment of the threshold is not an easy task, and hence it is difficult to select the optimal energy-rich path accordingly.

Time delay on-demand routing (TDOR) is an on-demand routing scheme with energy awareness developed from DSR [17]. TDOR introduces a time delay mechanism in the route discovery procedure to help select an energy-rich path. In TDOR, when an intermediate node receives a RREQ for the first time, it will not immediately forward but hold this RREQ for some time. And the holding time is inversely proportional to its residual battery capacity. This indicates that mobile nodes with more battery capacity will forward the RREQ earlier. Hence, the first arrived RREQ at the destination should contain an energy-rich path [18] so that the destination node can return a RREP back to the source node immediately when it receives the first RREQ.

In this Chapter, we propose a hybrid routing scheme with energy awareness, namely HEAR, to improve the performance of an energy-limited MANET in network throughput and lifetime. In HEAR, each mobile node adaptively exchanges energy-based Hello messages with its direct neighbors, and maintains a NT that records the information about neighbors within two hops all the time. By using its NT, a mobile node always knows the optimal route to a destination within two hops. For a destination node beyond two hops, HEAR introduces a time delay mechanism in the RREQ forwarding in order to search a route with good energy level. In addition, LRM is introduced to help replace energy-poor nodes along an active path with other energy-rich nodes by consulting their NTs. Hence, an active route can be locally adjusted from time to time to maintain a good energy level. The performance of HEAR is compared with DSR, TDOR and HYB scheme in terms of network throughput and network lifetime. HYB is essentially HEAR but without HEAR's energy-related mechanisms. Simulation results show that HEAR can outperform the other three schemes in such an energy-limited MANET.

6.2 Hybrid Energy Aware Routing (HEAR)

HEAR is hybrid routing scheme. Each mobile node consistently maintains the routes to any other mobile nodes within its 2-hop neighborhood by its NT and this is regarded as the proactive part of HEAR. For destination nodes beyond two hops, a source node may need to actively seek a new route on-demand or use a known route recorded in its RC to reach its corresponding destination nodes. The on-demand component of HEAR inherits most features from DSR. Unless otherwise stated, HEAR implements its on-demand component as specified in the IETF draft for DSR [17]. The new features of HEAR are introduced in detail in this Section.

1) Energy-aware Adaptive Hello

Each mobile node in HEAR adaptively sends out Hello message to its 1-hop neighbors. The Hello includes its own ID and energy level, as well as all those of its 1-hop neighbors. In order to reduce the effect of Hello message draining mobile nodes' energy, each mobile node dynamically adjusts its Hello frequency based on its residual energy level. Their relationship is defined by the following equation:

$$h = \begin{cases} h_B & \text{if } E \geq E_{Thre_} \\ h_M \times \frac{E_{Thre_} - E}{E_{Thre_}} & \text{if } E < E_{Thre_} \end{cases} \quad (6.1)$$

where h is the Hello interval for a mobile node at a given time. h_B and h_M refer to the basic Hello interval and the maximum allowed Hello interval respectively. E is the residual energy level of a mobile node, and $E_{Thre_}$ is the preset energy level threshold.

By limiting the Hello frequency for energy-poor nodes, the energy consumed by broadcasting Hello message for such mobile nodes can be reduced. Since an energy-poor node broadcasts Hello message less frequently, fewer mobile nodes can keep an entry for it in their NTs. Consequently, such an energy-poor node is less often used as a data relay and its energy drain rate due to serving as a data relay can be reduced.

2) 2-hop NT Maintenance

A mobile node can maintain a 2-hop NT by exchanging Hello message with its 1-hop neighbors. In addition, a mobile node also attach its energy level in other outgoing packets, including data packets and routing packets and this information may travel up to two hops. Hence, the information about this mobile node in its neighbors' NT can also be updated when its neighbors receive or overhear such packets. This update can be achieved at low cost because a mobile node adds only several bytes in the header of its outgoing packets to include the corresponding information. By such local information exchange, each mobile node not only can obtain and update information about other mobile nodes but also the network topology within two hops. Also, a mobile node scans its NT every *NTScanInterval* and deletes entries with elapsed time longer than a predefined *entryValidPeriod* since the last *entryUpdateTime*.

The recorded information in each entry of a mobile node's NT includes a neighbor's ID, the hop distance to the neighbor (1-hop or 2-hop), the next hop node for reaching the 2-hop neighbor, the residual energy level of the neighbor and the most recent *entryUpdateTime*. An example MANET is shown in Figure 6.1. And the NT at node n_1 corresponding to Figure 6.1 is shown in Table 6.1.

The NT of node n_1 only contains information about other mobile nodes within the dotted circle because the NT of a mobile node only records information for its 1-hop and 2-hop neighbors.

Figure 6.1 MANET Topology.

Table 6.1: NT of Node n_1 .

Node ID	Hop Distance	Next Hop	Energy Level (J)	entryUpdateTime (S)
2	1	NA	37.5	84.2
3	1	NA	33.0	83.6
3	2	2	32.7	84.2
4	2	2	39.0	84.2
2	2	3	37.9	83.6
4	2	3	39.2	83.6
5	2	3	36.8	83.6
6	2	3	42.0	83.6

3) Route Searching in NT and RC

When a source node **S** originates a data packet for delivery, it first checks its NT. If the destination node **D** is found to be a 1-hop neighbor in its NT, it sends the data packets directly. If **D** is found to be a 2-hop neighbor and can be reached through more than one 1-hop neighbor, **S** chooses the one with the highest energy

level as the intermediate node to reach **D**. If **S** cannot find the corresponding **D** in its NT, i.e. **D** is more than 2-hop away, it consults its RC. If a route is found, **S** simply uses it for sending the data packet. Otherwise, **S** initiates a RREQ to discover a route to **D**. Here, RC is a storage space for recording discovered routes to other mobile nodes.

In HEAR, RC is composed of two parts: primary route cache (PRC) and secondary route cache (SRC). Only routes obtained from RREPs or route energy messages (RENRs) that are destined to itself, can be cached in a mobile node's PRC. Here, RENR is a newly created routing message in HEAR and its details will be explained in 5). Routes obtained by other means, such as eavesdropping or received from RREQs or RERRs, can only be recorded in SRC. When **S** searches in its RC for a route to some **D**, it would first check its PRC. If a route is found there, **S** would not consult its SRC. If multiple routes are found in its PRC, **S** chooses the most recent one for data delivery. If no route to **D** is found in its PRC, **S** then searches its SRC. If multiple routes are found in its SRC, **S** chooses the one with minimum number of hops for data delivery. In HEAR, the routes cached in the SRC cannot be promoted to the PRC even if they are used as active routes.

If **S** uses a route cached in its SRC for delivering data packets to some **D**, i.e. it has no route to the corresponding **D** recorded in its PRC, it will immediately initiate a route discovery procedure with time delay mechanism for searching a new route to **D** as explained in 4). Allowing **S** to use routes cached in its SRC can avoid a data packet from being dropped due to no route, although the utilized route from SRC may not be composed of energy-rich nodes. Subsequently if **S**

discovers a new route that considers the energy metrics, it will be cached in the PRC of **S**, and replace the currently used route for data delivery.

4) Route discovery with time delay mechanism

If a route to some **D** is not found in the NT or RC of **S**, **S** initiates a RREQ to discover a route to **D**. And a time delay mechanism [18] is introduced in the RREQ forwarding of the route discovery procedure in HEAR. When an intermediate mobile node **IM** receives such a RREQ not destined to itself for the first time, it does not use routes cached in its RC for replying. This is because the cached routes may no longer be guaranteed with good energy level. However, this **IM** will append its own ID information to the packet header and rebroadcasts this RREQ after a certain holding time, t_{hold} , which is dependant on its residual energy level, E , as follows:

$$t_{hold} = \begin{cases} 0 & \text{if } E \geq E_{Thre_} \\ t_{Max} \times \frac{E_{Thre_} - E}{E_{Thre_}} & \text{if } E < E_{Thre_} \end{cases} \quad (6.2)$$

where t_{Max} is the maximum time period that an **IM** can hold a RREQ and $E_{Thre_}$ is the predefined energy level for comparison purpose.

Thus, the first arrived RREQ at **D** should contain a route with good energy level for delivering data for this source-destination pair [18]. Upon receiving, **D** immediately returns a RREP with the discovered route, without waiting for other RREQs that may arrive later. Hence, **D** only needs to reply to the first arrived RREQ in HEAR.

5) Local Replacement Mechanism (LRM)

S will periodically check the energy level of **IMs** along an active path recorded in its PRC in order to replace energy-poor **IMs**. To do so, some new fields, *checkFlag*, *minEnergyNodeID* and *minEnergy*, are added to the data packet header. Initially, S turns on *checkFlag* and sets *minEnergy* to infinity in the data packet header during this energy checking process. When an **IM** along this path receives such a data packet, it compares its own residual energy level with *minEnergy* in the header. If its energy level is lower than *minEnergy*, it replaces *minEnergy* with its own energy level and attaches its own ID in *minEnergyNodeID*, and then forwards the packet to its next hop. Otherwise, it simply forwards the data packet. All subsequent downstream nodes that receive this special data packet will process it in the same manner.

Upon receiving, **D** can identify the **IM** with minimum energy level along the path and returns a RENR message immediately back to S. A field named *RENRF* is added to the packet header and set to be True to indicate that such a routing packet is a RENR. Fields *minEnergyNodeID* and *minEnergy* are added to the RENR packet header and their values are copied from the corresponding part of the just received data packet. Another new flag, *replaceSucc*, is added to the RENR packet header as well and set to False initially. RENR is forwarded to S by inverting the source route recorded in the received data packet header.

When an **IM** receives such a RENR, it checks whether its next hop is the one with minimum energy level along the path by comparing the ID of its next hop with *minEnergyNodeID*. If yes, it consults its NT to find out whether it has a higher energy 1-hop neighbor that is also directly connected to the downstream node of its current next hop along the active path. If more than one eligible 1-hop

neighbor is available, it chooses the one with the highest energy level to replace its current next hop along the active path, and sets the *replaceSucc* bit to True. If no such 1-hop neighbor is found, the **IM** simply forwards the RENR to **S**. When the RENR message arrives at **S**, it first checks the *replaceSucc* bit. If *replaceSucc*=True, **S** then knows that the node replacement has been successful. It then copies the reversed route of received RENR to its PRC and deletes the previously used route with the replaced **IM** from its PRC. **S** will use this newly recorded route for the delivery of subsequent data packets. Hence, the **IM** with minimum energy can avoid its energy drain from serving as a data relay for this communication pair. If *replaceSucc*=False, which means that the replacement has failed, **S** would continue to use the existing route for data forwarding. Figure 6.2 shows how an RENR helps replace the **IM** with minimum energy level along an active path. n_1 utilizes a route $n_1-n_3-n_4-n_9$, recorded in its PRC for delivering data to n_9 . Periodically, n_1 tries to determine the minimum energy **IM** along this path by adding the information fields, *minEnergyNodeID* and *minEnergy*, into the header of its data packets to n_9 . Upon receiving such a data packet, n_9 initiates a RENR destined to n_1 by using the reversed route $n_9-n_4-n_3-n_1$. However, n_9 finds that its next hop n_4 , is the one with minimum energy along this path. This leads n_9 to consult its NT and selects a higher energy 1-hop neighbor n_5 , which is also connected to n_3 to replace n_4 in the reversed path. Then *replaceSucc* bit in the RENR is set to True. When n_1 receives this RENR, it copies the new route $n_1-n_3-n_5-n_9$ into its PRC and deletes the previously used route $n_1-n_3-n_4-n_9$ from its PRC.

Figure 6.2 Illustration of LRM.

6) Source-Initiated Route Refreshing

Although LRM is helpful, it can release only one mobile node from data relaying at one time. However, it is possible that several mobile nodes along an active path become energy-poor with time and need to be replaced with other nodes with higher energy level. This issue is addressed by having each mobile node periodically check the utilization time of routes in its PRC. If **S** finds that the utilization time for a route in its PRC is longer than a predefined *maxUsePeriod*, it initiates a RREQ to rediscover a route to the corresponding **D**. Thus, HEAR can avoid some energy-poor mobile nodes from being over utilized and **S** can set up a new route with good energy level to **D**.

6.3 Simulation Models and Performance Metrics

6.3.1 Simulation Models

The NS-2 simulator with wireless extension [80] is used for simulating the performance of four routing schemes: HEAR, HYB, TDOR and DSR. The random waypoint model [82] is adopted for simulating movement behaviors of all mobile nodes in the MANET. Table 6.2 shows the default parametric values used in the simulations.

Table 6.2: Default Simulation Parameters.

Parameters	Values
Simulation Time	1000s
Number of Mobile Nodes, N	50
Simulation Area	1000m $\times$ 1000m
Transmission Range for Mobile Nodes	250m
Transmission Data Rate	1Mbps
Mobility Model	Random Waypoint
Pause Time for Mobile Nodes	0.0s
Max. Speed for Mobile Nodes, v_{max}	10m/s
Traffic Pairs, n_t	10
Data Generation Rate at Source Nodes, r_s	4pkts/s
Packet Size of Data Packets Generated at Upper layer	512Bytes
Buffer Size for Queuing Outgoing Packets	50
Initial Battery Capacity of Mobile Nodes	600J
Transmission Power Consumption	1.4w
Receiving/Overhearing Power Consumption	1.0w
Idle State Power Consumption	0.43w
Energy Threshold, $E_{thre_}$	500J (in HEAR only)
Basic Hello Message Interval, h_B	1.0s (in HYB and HEAR)
Max Hello Message Interval, h_M	5.0 (in HEAR only)
$NTScanInterval$	0.2s (in HYB and HEAR)

<i>entryValidPeriod</i> in NT	3.3s (in HYB and HEAR)
Max Holding Time of a RREQ, t_{MAX}	0.3s (in TDOR and HEAR)
<i>maxUsePeriod</i>	10.0s (in TDOR and HEAR)

Simulation results presented for the four schemes are based on different mobility speeds v_{max} , and different network traffic loads in terms of number of traffic pairs n_t , and data rate generated at source nodes r_s . *Max Holding Time of a RREQ* refers to the longest time that an intermediate node can hold a RREQ before rebroadcasting it in HEAR and TDOR. *NT Scan Interval* decides how often a mobile node scans its NT to delete obsolete entries in HEAR and HYB. *entryValidPeriod* in NT shows the maximum time that an information entry of a neighbor node can reside in the NT without being updated in both HEAR and HYB. While a mobile node in HYB always sends a Hello message on every *Basic Hello Message Interval*, a mobile node in HEAR dynamically adjusts its Hello interval between *Basic Hello Message Interval* and *Max Hello Message Interval* based on its residual energy level. *Energy Threshold*, along with the residual energy level E of a mobile node, is used in HEAR to help the mobile node to decide its next Hello interval and RREQ forwarding delay.

6.3.2 Performance Metrics

HEAR is compared with HYB, TDOR and DSR in terms of the following performance metrics: i) average lifetime of mobile nodes T_L ; ii) number of mobile nodes “alive” during simulation n_L ; iii) average expiration time of communication pairs T_{exp} ; iv) the total number of data packets received by destination nodes N_p ; v) data packet delivery ratio r_p ; and vi) routing overheads O_v . The first three metrics evaluate the network

lifetime performance. N_p and r_p evaluate the network throughput performance. Finally, O_v reflects the routing efficiency performance for each scheme.

r_p is defined as the total number of data packets received by all destination nodes over the total number of data packets sent by all source nodes. O_v refers to the total number of non-data packets transmitted at the IP layer during the simulation. Each transmission of a non-data packet from one hop to another hop is counted as one routing packet [81].

6.4 Simulation Results and Discussions

In this Section, we study the performance of each routing scheme with different v_{max} for mobile nodes, different n_t in a network, and different r_s for data sources. And the simulation result of each point on the plotted figures is the average over 10 random simulations following the specific mobility and traffic parameters.

6.4.1 Network Lifetime Performance

This sub Section evaluates the network lifetime performance in terms of T_L , n_L , and T_{exp} based on different v_{max} or r_s .

Figure 6.3 shows that T_L in all the four schemes drops with v_{max} . It is observed that HEAR and TDOR can outperform HYB and DSR. This is because HEAR and TDOR avoid using energy-poor mobile nodes to serve as data relays so that the corresponding network can operate longer and the average lifetime of mobile nodes is prolonged. Although TDOR introduces the time delay mechanism in selecting routes with good energy level, the routes in use are only refreshed periodically to maintain their energy level. HEAR, on the other hand, can adjust an active route based on mobile

nodes’ NT information by the introduction of LRM so that a route in use can be refreshed at low cost more frequently to maintain good energy level. Hence, HEAR is more actively energy-aware than TDOR and can better prolong the network lifetime.

Figure 6.3 T_L vs. v_{max} with $n_t = 15$ and $r_s = 4$.

Figure 6.4 shows n_L during the simulation for the four schemes based on two different r_s (4pkts/s and 8pkts/s). Only the results from time of 650s onwards are presented because a significant portion of mobile nodes is found to have consumed up their battery energy during this time window for all the four schemes. As r_s increases from 4pkts/s to 8pkts/s, n_L in a network at a specific simulation time decreases for all the four schemes. This is because mobile nodes send, receive and forward more packets with heavier traffic load and hence drain their battery energy at a faster rate. From this figure, we could also see that HEAR has consistently the highest number of mobile nodes “alive” in the network as compared to TDOR, DSR and HYB between 650s and 750s. This

is because HEAR is able to more successfully avoid the early “death” of mobile nodes in an energy-limited network with the implementation of its energy metrics. After 800s, n_L in the network is approaching to zero for all the four schemes.

Figure 6.4 n_L vs. Simulation Time with $v_{max} = 10\text{m/s}$ and $n_t = 15$.

Table 6.3: Average Communication Expiration Time (T_{exp}).

	10m/s, 4pkts/s	15m/s, 4pkts/s	10m/s, 8pkts/s
DSR	687.4	686.0	646.1
TDOR	690.1	687.8	647.9
HYB	685.8	684.2	649.2
HEAR	697.6	696.1	658.3

Table 6.3 presents T_{exp} with different v_{max} and r_s . The expiration time of a communication pair is the time that the last data packet received by the corresponding destination node of this pair. A communication pair is said to be expired when its

corresponding source or destination is depleted of energy or when this pair is unable to find a valid path between them for data delivery. The result shows that HEAR can prolong T_{exp} because it is able to monitor and replace energy-poor nodes on active routes with energy-rich nodes by its LRM from time to time. Consequently, HEAR is able to deliver more data packets in an energy-limited MANET as compared with the other three schemes as shown in next Section.

6.4.2 Network Throughput Performance

This sub Section evaluates the network throughput performance in terms of r_p and N_p based on different v_{max} , r_s or n_t .

Figure 6.5 shows r_p with different v_{max} . As v_{max} increases from 5m/s to 15m/s, the performance for all the four schemes drops. This is because communication paths are more likely to break in a highly mobile network and thus affects the data delivery. However, surprisingly, in a completely static scenario ($v_{max} = 0$), the performance is worst for all the four schemes. This is because there is less diversity of route choices in a static network and some nodes are excessively utilized as data relays. Hence, these nodes drain their energy much faster, and their early “death” could cause problems such as network partition and thus affects the data delivery [9]. HEAR outperforms the other three schemes for any v_{max} . The performance improvement of HEAR over the other three schemes increases with v_{max} and the improvement over DSR is about 15% at 15m/s. Since mobile nodes connect and disconnect with each other more often in a highly mobile network, the information entries in a mobile node’s NT change more often in HEAR. Consequently, the LRM in HEAR can perform better in a higher speed MANET, resulting

in the network lifetime and throughput improvement of HEAR over the other schemes thus increases with v_{max} .

Figure 6.5 r_p vs. v_{max} with $n_t = 15$ and $r_s = 4$.

Figure 6.6 shows N_p with different v_{max} for the four schemes. Similarly, N_p drops as v_{max} increases from 5m/s to 15m/s, but increases as v_{max} increases from 0m/s to 5m/s for all the four schemes. In addition, HEAR outperforms the other three schemes because its energy awareness enables the network operate longer and hence more packets can be delivered. HEAR can deliver 20% more data packets as compared to DSR at 15m/s. By comparing Figure 6.5 and Figure 6.6, it is observed that the average performance gain by HEAR over other remaining schemes is larger in Figure 6.6. This is because HEAR prolongs the network operation time and enables communication pairs to live longer. Thus, it not only delivers more data packets, but also sends more data packets than the other schemes. This makes the improvement of HEAR in terms of r_p be less significant than N_p .

Figure 6.6 N_p vs. v_{max} with $n_t = 15$ and $r_s = 4$.

Figure 6.7 shows that r_p decreases with r_s for all the four schemes because as r_s increases the network becomes more congested with traffic and hence more packets could be dropped due to buffer or interface queue full. Thus, routes between communication pairs may need to be re-built more frequently in order to have data delivered. Hybrid schemes, such as HYB and HEAR, are easier to find routes using the information in NTs and thus fewer packets are dropped due to no route. Consequently, HYB and HEAR have a better packet delivery performance as compared to DSR and TDOR. HEAR outperforms DSR for almost 30% at $r_s=8\text{pkts/s}$ because of its proactive routing component and active energy-aware mechanisms.

Figure 6.7 r_p vs. r_s with $v_{max} = 10\text{m/s}$ and $n_t = 15$.

Figure 6.8 N_p vs. r_s with $v_{max} = 10\text{m/s}$ and $n_t = 15$.

Figure 6.8 shows N_p vs. r_s . The performance among the four schemes is found to be comparable in a network with light data traffic load. However, as r_s increases, HEAR can outperform the other three schemes and the improvement over DSR is more than 35% at $r_s=8\text{pkts/s}$. This is because a hybrid routing scheme can find routes more easily in a moderately or heavily-loaded network as explained for Figure 6.7. Furthermore, the energy-aware mechanisms in HEAR can prolong the network operation time so that more data packets can be transmitted and received.

Figure 6.9 and Figure 6.10 show r_p and N_p with different n_t . Expectedly, the performance of all schemes drops as n_t increases due to higher traffic. Compared with Figure 6.7 and Figure 6.8, we note that n_t has a greater impact on the deterioration of packet delivery performance than r_s . This is because with more traffic pairs, more routes need to be established and maintained. Therefore, more data packets could be dropped due to no valid routes, especially for pure on-demand routing schemes such as DSR and TDOR. HYB and HEAR still outperform DSR and TDOR for large n_t due to the same reason as explained for Figure 6.7.

Figure 6.9 r_p vs. n_t with $v_{max} = 10\text{m/s}$ and $r_s = 4$.

Figure 6.10 N_p vs. n_t with $v_{max} = 10\text{m/s}$ and $r_s = 4$.

6.4.3 Routing Overheads Performance

This sub Section evaluates O_v of the four schemes based on different v_{max} , or n_t .

Figure 6.11 O_v vs. v_{max} with $n_t = 15$ and $r_s = 4$.

Figure 6.11 shows that O_v increases with v_{max} for all the four schemes. This is because with more routes broken in a highly mobile environment, the routing schemes need to initiate more routing packets to establish and maintain valid routes. Comparatively, DSR and TDOR generates smaller O_v than HYB and HEAR because the latter has the proactive routing component. Between TDOR and DSR, the former has larger O_v because it periodically re-discovers energy-rich paths for communication pairs even if the current paths are not broken. Also, HEAR generates larger O_v than HYB due to its energy mechanisms such as LRM and route refreshing. HEAR outperforms the other three schemes in terms of network throughput and lifetime because the performance

benefits due to its actively energy-aware routing mechanisms in HEAR have outweighed the cost of higher routing overheads.

Figure 6.12 shows O_v increases with n_t for the four schemes. O_v increases faster for DSR and TDOR than for HYB and HEAR. This is because in a network with heavy traffic load, hybrid routing schemes such as HYB and HEAR, can build a route more easily with the information from NT and generate less routing traffic due to network-wide route discovery. Energy mechanisms bring additional routing traffic so that HEAR and TDOR generate larger O_v than HYB and DSR respectively.

Figure 6.12 O_v vs. n_t with $v_{max} = 10\text{m/s}$ and $r_s = 4$.

It can be seen from Figure 6.11 and Figure 6.12 that HEAR generates higher routing overheads than the other three schemes. As we know, when a network is overloaded, larger overheads will deteriorate its performance in terms of throughput and packet delivery ratio. However, the actual network overheads are decided by the MAC-

layer overheads instead of routing overheads in the network layer. Figure 6.13 shows the different routing overheads distribution, in terms of Hello, RREQ, RREP and RERR, of different schemes with $v_{max} = 10\text{m/s}$, $n_t = 15$ and $r_s = 4$. For the four types of routing overheads in the network layer, Hello and RREQ belong to broadcast messages, whereas RREP and RERR are unicast messages. Compared to broadcast message, delivering a unicast message is more complicated when IEEE 802.11 MAC is deployed as the underlying MAC scheme because it requires RTS-CTS-message-Acknowledgement procedure and up to seven times of RTS retransmission and four times of message retransmission to guarantee packet delivery. Such a procedure indicates that delivering a unicast message in the network layer invokes additional MAC-layer overheads. In other words, unicast messages (RREP and RERR) incur more MAC-layer overheads compared with broadcast messages (Hello and RREQ), and thus overload the network more likely.

In Figure 6.13, it can be seen that the routing overheads of HYB and HEAR have the Hello messages, whereas DSR and TDOR do not have. In HEAR, the RREQ messages occupies a much larger portion, and the RREP and RERR messages dwell a smaller portion among the total routing overheads. In order to find energy-rich paths during each route discovery procedure, HEAR does not allow intermediate nodes to utilize their RCs to return RREPs to a source node, and only allow the destination to reply to the first arrived RREQ. Also, HEAR can utilize the NT maintained at each mobile node to locally repair broken routes. All these above mentioned mechanisms can greatly reduce RREPs and RERRs generated in HEAR. Even though HEAR generates more routing overhead, it has much lower number of unicast messages (RREP and RERR), which introduce much more overheads in the MAC layer. This probably can

explain why the larger routing overheads of HEAR generated at the network layer does not affect its network throughput performance.

Figure 6.13 Distribution of Routing Overheads with $v_{max} = 10\text{m/s}$, $n_t = 15$ and $r_s = 4$.

6.5 Conclusions

HEAR has been proposed and studied in this Chapter. In HEAR, each node maintains a 2-hop NT, and connects to its corresponding 2-hop destination via a direct neighbor with the highest energy level. A mobile node also adaptively adjusts its Hello interval based on its energy level and attaches its energy information on any outgoing packets to reduce the number of explicit Hello sent. For destinations beyond 2-hop, a time delay mechanism is introduced in the RREQ forwarding process so that energy-rich paths can be found during route discovery. Furthermore, HEAR can adjust the active paths from time to time with low cost by utilizing the LRM. HEAR also re-discovers a path based on the energy metrics for a communication pair if it determines that the current route has been excessively used. Consequently, HEAR involves mobile nodes with higher energy level as intermediate nodes in the new routes. Simulation results show that HEAR

performs well in an energy-limited MANET, especially with high mobility and high traffic load as compared to DSR, TDOR and HYB. In conclusion, HEAR successfully achieves its aims of providing better system performance in terms of network throughput and network lifetime.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

MANETs face a lot of challenging issues for its actual implementation and application, such as scalability, energy limitation, wireless data rate limitation, security threats, and protocol standardization. In this Thesis, we focus on providing solutions for issues related with scalability and energy limitation.

This Thesis can be summarized as follows:

- 1) First we gave a comprehensive survey on twelve proposed clustering schemes in MANET research. We gave the fundamental knowledge for MANET clustering, showed the pros and cons of cluster based MANETs, categorized those proposed clustering schemes based on their objectives, and discussed their mechanisms, cost and feasible applications in detail. With this survey researchers would find that a cluster based MANET has many issues to concern, such as how to maintain the cluster structure to respond the underlying network topology change or how to reduce the clustering cost due to cluster formation and maintenance. This is because the clustering scheme and its maintained cluster structure may affect the applications residing on it greatly. Clustering cost needs to be considered always because it is an important metric to evaluate the performance and scalability improvement of a clustering scheme.

- 2) Then, we proposed a 1-hop CH-based clustering scheme, named ECS, for a large and dense MANET. ECS adopts random claim mechanism for the CH election in the cluster formation phase, and thus does not require the frozen period for initial cluster construction. With the introduction of CG and a cluster deletion mechanism, ECS can effectively eliminate small and unnecessary clusters. Hence, ECS effectively reduces the cluster overlapping in a MANET with moderate to high node density, which is helpful in simplifying the network structure. In addition, ECS can reduce the cluster re-affiliation rate, limit the re-clustering situations and eliminate the ripple effect of re-clustering to provide a stable cluster structure for upper layer applications. Simulation results show that ECS can perform well and outperform RCC and HCCM in a large mobile network environment in terms of CH lifetime, membership lifetime, maintained cluster number and clustering overheads. In conclusion, ECS successfully fulfills its targets at providing a less overlapping and more stable cluster structure with moderate clustering cost for a large and dense MANET.
- 3) Since a cluster structure is to serve for its upper layer applications, especially a routing protocol residing on it. It is necessary to evaluate the performance of its upper layer routing protocol in order to judge the efficiency of a clustering scheme. Also, the upper layer routing protocol should be able to make full use of the cluster structure to enhance its performance. CBRP is considered as a promising cluster based routing technique and has been enrolled as an Internet draft. Also, lots of research papers related with CBRP have been published. Hence, following the proposed ECS, we implement the routing mechanisms in CBRP on ECS because ECS can provide a more stable and less overlapping

cluster structure than the clustering mechanisms used in CBRP. Such a cluster based routing scheme is named ECBRP in this Thesis. Some routing mechanisms in CBRP are modified in order to make good use of the underlying cluster structure of ECS. Besides, an information table update mechanism by utilizing routing events and data forwarding events to enhance the clustering and routing performance are introduced. The performance of ECBRP in terms of packet delivery ratio, normalized routing overhead, end-to-end transmission delay and so on is compared with that of CBRP and DSR. By evaluating the performance of ECBRP and CBRP, we can know that a more stable and less overlapping cluster structure is more efficient in simplifying the network structure and hence reduces the routing overheads without impairing the routing efficiency for a MANET with moderate to high node density. By comparing the performance between DSR and the two cluster based routing schemes, we can know that a cluster structure can greatly reduce the processing overheads by simplifying the network structure and thus to make the routing more efficient as its network traffic load, node mobility or network size increases. Since ECBRP highly depends on the underlying cluster structure and NTs at mobile nodes for routing, and the cluster structure and NTs are mainly maintained by Hello messages exchange between neighboring nodes. The performance of ECBRP with different Hello frequencies is also studied and discussed. It can be concluded from the simulation results that an appropriate and moderate Hello frequency is important to achieve good performance, especially when the network is overloaded. With the simulation results, we concluded that ECBRP is a promising cluster based routing scheme by

combining routing mechanisms from CBRP and cluster mechanisms from ECS, and modifying them accordingly so that they can benefit from each other.

- 4) Energy limitation poses a severe challenge for the performance of a MANET because mobile nodes in a MANET are normally based on battery energy supply and cannot be recharged during operation. Excessively serving as data relays for other mobile nodes will make a mobile node overloaded and drain its battery energy quickly, and consequently deplete its battery energy early. It is important for a routing protocol to properly choose data relays with enough residual energy for avoiding possible network partition and guaranteeing performance for an energy-limited MANET. In this Thesis, we proposed HEAR for an energy-limited MANET. In HEAR, we integrate some energy aware mechanisms into routing, such as exchanging energy-based adaptive Hello between neighbors to maintain neighbor information of a local area, introducing time delay mechanism in the RREQ forwarding based on an intermediate node's residual battery energy, and replacing an energy-poor node along an active path with a mobile node with more residual energy based on the neighbor information. Hence, HEAR can outperform other routing protocols, such as HYB, TDOR and DSR in terms of lifetime of mobile nodes, communication pair lifetime, packet delivery ratio and total data packet received. In sum, HEAR successfully achieves its objectives at improving network lifetime and throughput in an energy-limited MANET.

7.2 Recommendations for Future Work

Since scalability and energy limitation are two of the major challenges for the real application of MANETs, our future work for MANETs will still focus on these two areas.

We suggest the following topics for our future work:

- 1) ECS can be modified to cater for some more specific application and situation. For example, in a military MANET, both vehicles and soldiers may be considered as mobile nodes. However, vehicle nodes may have longer transmission range, more sufficient power supply and higher processing capability. Hence, if we apply ECS in such a scenario, the vehicle nodes should be given higher priority in CH election. In other words, in a MANET composed of heterogeneous mobile nodes, more attributes of a mobile node should be considered in the clustering procedures. In addition, ECS can be extended to k -hop situation with $k > 1$ so that its performance can be compared with other multi-hop clustering schemes. And thus, we can have a more comprehensive judgment about the performance of ECS.
- 2) Cluster based routing seems to be a promising technique for solving the scalability problem of MANETs by building the routing backbone with partial mobile nodes based on their cluster status. However, mobile nodes involved in the routing backbone may often serve as data relays, and hence may drain their battery energy much faster due to excessively bearing a router's function. This may cause the early death of partial mobile nodes due to energy depletion, and increase the chances of network partition and communication interruption due to lack of mobile nodes in the network. Hence, energy-related metrics should be considered when designing a clustering scheme and a cluster based routing scheme so that the energy dissipation rate can be more balanced among all mobile nodes. Actually some research work has been proposed to balance the energy consumption of different mobile nodes in a clustering scheme by rotating the CH role periodically among mobile nodes or choosing mobile nodes with higher

residual battery capacity to serve as CHs when possible [33, 34, 78]. Therefore, the energy consumption difference due to serving different cluster-related functions can be reduced. Hence, how to apply energy metrics into ECS and ECBRP to make them more energy-aware without affecting the cluster stability and routing efficiency should be the next step of our research work.

- 3) Node mobility imposes additional challenges to the scalability of MANETs because the movement of mobile nodes is one of the major factors that invoke the route invalidation, and thus increases the processing overheads and adversely affects the packet delivery. Conventionally, node mobility in a MANET is viewed as the superposition of individual and independent random moves of each mobile node. However, in reality, especially in some military scenarios and disaster recovery environments, mobile nodes exhibit some collective group mobility patterns [86, 87]. In this situation, only one route towards a group of mobile nodes, who are located closely, move together and show the similar mobility pattern, is needed. By exploiting the group mobility pattern, the routing can be greatly simplified and the routing overheads can be greatly reduced without affecting the routing efficiency. However, even in a scenario where group mobility is common, individual moving mobile nodes still exist and different groups of mobile nodes may show different group mobility patterns [86]. Hence, one possible topic for future research work is to develop an efficient and scalable routing protocol for a MANET with heterogeneous group mobility. The idea is to use two different types of routing mechanisms for mobile nodes with group mobility behaviour and those with individual mobility behaviour. A mobile node, who is the most representative for its mobility group can be selected as a CH to

perform the inter-cluster routing to other mobility groups based on cluster based routing. And the pure on-demand routing can be implemented for maintaining routes to mobile nodes with the individual mobility pattern. The routing performance of such a MANET should be guaranteed when the network size or network traffic load increases.

- 4) A mobile node consumes its energy when it actively transmits or receives packets or when it stays idle listening to the medium. Statistics shows that a mobile node stays idle for a considerable portion of time in a typical MANET [8, 57]. Hence, reducing the energy consumption during idle periods by turning off its radio should be a promising way to reduce the energy consumption for a mobile node. By building a cluster structure for a MANET and maintaining a connected routing backbone based on mobile nodes' cluster status, mobile nodes outside the routing backbone can switch to sleep mode and thus save their energy. How to integrate the inactive energy saving metrics with ECS and ECBRP so that they can show better energy performance in an energy-limited MANET is with interest. This may require rotating the router's function among mobile nodes so that every mobile node gets chances to save its energy by switching to sleep mode. At the same time, this kind of rotation should not affect the cluster stability and routing efficiency much so that ECS and ECBRP can maintain their satisfying clustering and routing performance.

Author's Publications

Book Article

1. J. Y. Yu and P. H. J. Chong, "Clustering techniques for mobile ad hoc networks (MANETs), " in *Encyclopedia of Wireless and Mobile Communications*, edited by B. Furht, CRC Press, 2007.

Journal Papers

2. J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32-48, First Quarter, 2005.
3. J. Y. Yu and P. H. J. Chong, "An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs)," *Computer Communications*, vol. 30, no. 1, pp. 5-16, December, 2006.
4. Q. J. Gao, J. Y. Yu, P. H. J. Chong, P. L. So, and E. Gunawan, "Solutions for the 'silent node' problem in automatic meter reading system using powerline communications," *IEEE Transactions on Power Delivery*, vol. 23, no. 1, pp. 150-156, January, 2008.
5. P. H. J. Chong and J. Y. Yu, "Performance analysis of microcell/macrocell with reuse partitioning in TDMA-based cellular systems," *International Journal of Communication Systems*, vol. 21, no. 9, pp. 923-954, Sept. 2008.

6. J. Y. Yu and P. H. J. Chong, "Implementation and Performance Evaluation of CBRP Based on ECS in MANETs," submitted to *IEEE Transactions on Mobile Computing* under second review.

Conference Papers

7. J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks," in *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, vol. 1, pp. 318-321, Victoria, B.C., Canada, August 28-30, 2003.
8. J. Y. Yu and P. H. J. Chong, "Energy-aware routing in mobile ad hoc networks (MANETs)," in *Proc. of Global Mobile Congress (GMC'05)*, pp. 90-95, Chongqing, China, October 10-12, 2005.
9. J. Y. Yu, P. H. J. Chong, P. L. So, and E. Gunawan, "Solutions for the 'silent node' problem in automatic meter reading system using powerline communications," in *Proc. of 7th International Power Engineering Conference (IPEC'05)*, pp. 245, Singapore, November 29-December 2, 2005.
10. J. Y. Yu, P. H. J. Chong and M. Yang, "Performance of microcell/macrocell cellular systems with reuse partitioning," in *Proc. of 3rd IEE Mobility Conference*, Bangkok, Thailand, October 25-27, 2006.
11. J. Y. Yu and P. H. J. Chong, "Performance of Efficient CBRP in Mobile Ad Hoc Networks (MANETS)," To be presented at *IEEE VTC-Fall*, Calgary, Canada, September 2008.

Bibliography

- [1] C. E. Perkins, *Ad hoc networking*. Boston: Addison-Wesley, 2001.
- [2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for 'Smart Dust'," in *Proc. of the Annual International Conference on Mobile Computing and Networking (MOBICOM'99)*, pp. 271-278, Seattle, WA, August 15-19, 1999.
- [3] H. Xiaoyan, X. Kaixin, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *Network, IEEE*, vol. 16, no. 4, pp. 11-21, July-August, 2002.
- [4] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 1, pp. 32-48, First Quarter, 2005.
- [5] X. Kaixin, H. Xiaoyan, and M. Gerla, "An ad hoc network with mobile backbones," in *Proc. of IEEE International Conference on Communications (ICC'02)*, vol. 5, pp. 3138-3143, New York City, April 28-May 2, 2002.
- [6] J. Y. Yu and P. H. J. Chong, "An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs)," *Computer Communications*, vol. 30, no. 1, pp. 5-16, December, 2006.
- [7] C. K. Toh, *Ad hoc mobile wireless networks: protocols and systems*: Prentice Hall PTR, 2001.
- [8] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481-494, September, 2002.

- [9] C. K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 39, no. 6, pp. 138-147, June, 2001.
- [10] C. K. Toh, H. Cobb, and D. A. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proc. of IEEE International Conference on Communications (ICC'01)*, vol. 9, pp. 2824-2829, Helsinki, Finland, June 11-14, 2001.
- [11] J. Wu and H. L. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. of 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M'99)*, pp. 7-14, Seattle, WA, August 20, 1999.
- [12] Y. P. Chen and A. L. Liestman, "Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks," in *Proc. of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'02)*, pp. 165-172, Lausanne, Switzerland, June 9-11, 2002.
- [13] C. Ching-Chuan, W. Hsiao-Kuang, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," in *Proc. of IEEE Singapore International Conference on Networks (SICON'97)*, pp. 197-211, Singapore, April 14-17, 1997.
- [14] M. Chatterjee, S. K. Sas, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM'00)*, vol. 3, pp. 1697-1701, San Francisco, CA, November 27-December 1, 2000.

Bibliography

- [15] M. L. Jiang, J. Y. Li, and Y. C. Tay, "Cluster Based Routing Protocol," Internet-Draft, draft-ietf-manet-dsr-10.txt, work-in-progress, 2004.
- [16] M. Gerla and T. Jack Tzu-Chieh, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255-265, 1995.
- [17] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing for mobile ad hoc networks," Internet-Draft, draft-ietf-manet-dsr-10.txt, work-in-progress, 2004.
- [18] C. Woncheol and K. Seong-Lyun, "A fully distributed routing algorithm for maximizing lifetime of a wireless ad hoc network," in *Proc. of 4th International Workshop on Mobile and Wireless Communications Network*, pp. 670-674, Stockholm, Sweden, September 9-11, 2002.
- [19] H. Xiaoyan, M. Gerla, Y. Yunjung, X. Kaixin, and K. Taek Jin, "Scalable ad hoc routing in large, dense wireless networks using clustering and landmarks," in *Proc. of IEEE International Conference on Communications (ICC'02)*, vol. 5, pp. 3179-3185, New York City, April 28-May 2, 2002.
- [20] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *ACM SIGACT News*, vol. 33, no. 2, pp. 60-73, June, 2002.
- [21] A. B. McDonald and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1466-1487, August, 1999.
- [22] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 15, no. 7, pp. 1265-1275, September, 1997.

- [23] H. Ting-Chao and T. Tzu-Jane, "A access-based clustering protocol for multihop wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1201-1210, July, 2001.
- [24] C. H. R. Lin and M. Gerla, "A distributed control scheme in multi-hop packet radio networks for voice/data traffic support," in *Proc. of IEEE International Conference on Communications (ICC'95)*, vol. 2, pp. 1238-1242, Seattle, WA, June 18-22, 1995.
- [25] U. C. Kozat, G. Kondylis, B. Ryu, and M. K. Marina, "Virtual dynamic backbone for mobile ad hoc networks," in *Proc. of IEEE International Conference on Communications (ICC'01)*, vol. 1, pp. 250-255, Helsinki, Finland, June 11-14, 2001.
- [26] M. R. Pearlman and Z. J. Haas, "Determining the optimal configuration for the zone routing protocol," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1395-1414, August, 1999.
- [27] A. Iwata, C. Ching-Chuan, P. Guangyu, M. Gerla, and C. Tsu-Wei, "Scalable routing strategies for ad hoc wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1369-1379, August, 1999.
- [28] C. Wenli, J. Nitin, and S. Singh, "ANMP: ad hoc network management protocol," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1506-1531, August, 1999.
- [29] 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, pp. 56-73, January, 1987.

- [30] J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks," in *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, vol. 1, pp. 318-321, Victoria, B.C., Canada, August 28-30, 2003.
- [31] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. of International Conference on Distributed Computing Systems Workshop (ICDCS'01)*, pp. 413-418, Mesa, AZ, April 16-19, 2001.
- [32] A. B. McDonald and T. F. Znati, "Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks," in *Proc. of 34th Annual Simulation Symposium (SS'01)*, pp. 27-35, Seattle, WA, April 22-26, 2001.
- [33] A. D. Amis and R. Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proc. of IEEE 3rd Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00)*, pp. 25-32, Richardson, TX, March 24-25, 2000.
- [34] W. Jie, G. Ming, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proc. of International Conference on Parallel Processing (ICPP'01)*, pp. 346-354, Valencia, Spain, September 3-7, 2001.
- [35] T. Ohta, S. Inoue, and Y. Kakuda, "An adaptive multihop clustering scheme for highly mobile ad hoc networks," in *Proc. of 6th International Symposium on Autonomous Decentralized Systems (ISADS'03)*, pp. 293-300, Pisa, Italy, April 9-11, 2003.

Bibliography

- [36] K. Taek Jin, M. Gerla, V. K. Varma, M. Barton, and T. R. Hsing, "Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1210-1220, August, 2003.
- [37] E. M. Belding-Royer, "Hierarchical routing in ad hoc mobile networks," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 515-532, August, 2002.
- [38] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proc. of IEEE VTS 50th Vehicular Technology Conference*, vol. 2, pp. 889-893, Amsterdam, Netherlands, September 19-22, 1999.
- [39] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proc. of IEEE International Conference on Communications (ICC'97)*, vol. 1, pp. 376-380, Montreal, Quebec, Canada, June 8-12, 1997.
- [40] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad hoc networks using a spine," in *Proc. of the 6th International Conference on Computer Communications and Networks (ICCCN'97)*, pp. 34-39, Las Vegas, NV, September 22-25, 1997.
- [41] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *Proc. of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, vol. 1, pp. 481-491, San Francisco, CA, March 30-April 3, 2003.

- [42] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: supporting dynamic power controlled routing in wireless ad hoc networks," *Wireless Networks*, vol. 9, no. 5, pp. 443-460, September, 2003.
- [43] T. Ohta, S. Inoue, Y. Kakuda, and K. Ishida, "An adaptive multihop clustering scheme for ad hoc networks with high mobility," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 7, pp. 1689-1697, July, 2003.
- [44] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, pp. 85-97, Dallas, TX, October 25-30, 1998.
- [45] E. M. Royer and T. Chai-Keong, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 6, no. 2, pp. 46-55, April, 1999.
- [46] P. Guanyu, M. Gerla, and C. Tsu-Wei, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *Proc. of IEEE International Conference on Communications (ICC'00)*, vol. 1, pp. 70-74, Orleans, LA, June 18-22, 2000.
- [47] B. Bellur and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," in *Proc. of 18th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM'99)*, vol. 1, pp. 178-186, New York, NY, March 21-25, 1999.
- [48] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. of Conference on*

Bibliography

- Communications Architectures, Protocols and Applications (SIGCOMM'94)*, pp. 234-244, London, UK, August 31-September 2, 1994.
- [49] T. Chai-Keong, "Associativity-based routing for ad-hoc mobile networks," *Wireless Personal Communications*, vol. 4, no. 2, pp. 103-139, March, 1997.
- [50] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. of 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*, vol. 3, pp. 1405-1413, Kobe, Japan, April 7-11, 1997.
- [51] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pp. 90-100, New Orleans, LA, February 25-26, 1999.
- [52] P. Guanyu, M. Gerla, and H. Xiaoyan, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility," in *Proc. of 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing (MOBIHOC'00)*, pp. 11-18, Boston, MA, August 11, 2000.
- [53] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *Networking, IEEE/ACM Transactions on*, vol. 9, no. 4, pp. 427-438, August, 2001.
- [54] C. Ching-Chuan and M. Gerla, "Routing and multicast in multihop, mobile wireless networks," in *Proc. of IEEE 6th International Conference on Universal Personal Communications Record*, vol. 2, pp. 546-551, San Diego, CA, October 12-16, 1997.
- [55] G. Pei, M. Gerla, X. Hong, and C. C. Chiang, "A wireless hierarchical routing protocol with group mobility," in *Proc. of IEEE Wireless Communications and*

- Networking Conference (WCNC'99)*, pp. 1538-1542, New Orleans, LA, September 21-24, 1999.
- [56] M. Gerla, H. Xiaoyan, and P. Guangyu, "Landmark routing for large ad hoc wireless networks," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM'00)*, vol. 3, pp. 1702-1706, San Francisco, CA, November 27-December 1, 2000.
- [57] K. Dongkyun, J. J. Garcia-Luna-Aceves, K. Obraczka, J. C. Cano, and P. Manzoni, "Routing mechanisms for mobile ad hoc networks based on the energy drain rate," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 161-173, April-June, 2003.
- [58] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. of the Annual International Conference on Mobile Computing and Networking (MOBICOM'01)*, pp. 70-84, Rome, Italy, July 16-21, 2001.
- [59] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proc. of the Annual International Conference on Mobile Computing and Networking (MOBICOM'01)*, pp. 97-107, Rome, Italy, July 16-21, 2001.
- [60] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 12, no. 11, pp. 1122-1133, November, 2001.
- [61] W. Kyungtae, Y. Chansu, L. Dongman, Y. Hee Yong, and B. Lee, "Non-blocking, localized routing algorithm for balanced energy consumption in mobile ad hoc networks," in *Proc. of 9th International Symposium on Modeling, Analysis and*

Bibliography

- Simulation of Computer and Telecommunication Systems (MASCOTS'01)*, pp. 117-124, Cincinnati, OH, August 15-18, 2001.
- [62] L. Li and L. Chunlin, "A QoS multicast routing protocol for clustering mobile ad hoc networks," *Computer Communications*, vol. 30, no. 7, pp. 1641-1654, May, 2007.
- [63] L. Xuefei and L. Cuthbert, "Node-disjoint multipath routing and distributed cross-layer QoS guarantees in mobile ad hoc networks," in *Proc. of 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)*, pp. 243-248, Las Vegas, NV, June 19-20, 2006.
- [64] C. Maria, G. Jose Ramon, H.-S. Angela, and V. Antonio, "Cross-layer routing for QoS provision in multiservice mobile ad hoc networks," in *Proc. of IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'06)*, pp. 1-5, Helsinki, Finland, September 11-14, 2006.
- [65] Y. Guangsong, X. Mingbo, C. Huihuang, and Y. Yan, "A novel cross-layer routing scheme of ad hoc networks with multi-rate mechanism," in *Proc. of International Conference on Wireless Communications, Networking and Mobile Computing (WCNM'05)*, vol. 2, pp. 701-704, Wuhan, China, September 23-26, 2005.
- [66] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307-321, September, 2000.
- [67] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. of 4th Annual*

- ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, pp. 76-84, Dallas, TX, October 25-30, 1998.
- [68] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of the Annual International Conference on Mobile Computing and Networking (MOBICOM'00)*, pp. 243-254, Boston, MA, August 6-11, 2000.
- [69] D. Bertsekas and R. Gallager, *Data networks*. Englewood Cliffs, N. J. : Prentice-Hall PTR, 1987.
- [70] G. Malkin, "RIP version 2 -- carrying additional information," RFC 1388 (proposed standard), Internet Engineering Task Force, 1993.
- [71] J. J. Garcia-Luna-Aceves, "A unified approach to loop-free routing using distance vectors or link states," in *Proc. of ACM Symposium on Communications Architectures & Protocols (SIGCOMM'89)*, vol. 19, pp. 212-223, Austin, TX, September 19-22, 1989.
- [72] J. McQuillan, I. Richer, and E. Rosen, "The new routing algorithm for the ARPANET," *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 28, no. 5, pp. 711-719, May, 1980.
- [73] "IP option 3, loose source route," RFC 791: Internet Protocol, 1981.
- [74] C. Yu, B. Lee, and H. Y. Youn, "Energy efficient routing protocols for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 8, pp. 959-973, December, 2003.
- [75] T. S. Rappaport, *Wireless communications: principles and practice* 2ed: Prentice Hall PTR, 2001.

- [76] S. Singh and C. S. Raghavendra, "PAMAS-power aware multi-access protocol with signalling for ad hoc networks," *Computer Communication Review*, vol. 28, no. 3, pp. 5-26, July, 1998.
- [77] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, pp. 181-190, Dallas, TX, October 25-30, 1998.
- [78] C. Cheng-fu, "Energy efficient routing for clustered wireless sensors network," in *Proc. of 29th Annual Conference of the IEEE Industrial Electronics Society (IECON'03)*, vol. 2, pp. 1437-1440, Roanoke, VA, November 2-6, 2003.
- [79] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 46-56, January-March, 2004.
- [80] K. Fall and K. Varadhan, "ns Notes and Documents " The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, < <http://www.isi.edu/nanam/ns/ns-documentation.html>>, Feb. 2000.
- [81] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson, "The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1439-1453, August, 1999.
- [82] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483-502, August, 2002.

Bibliography

- [83] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 8, no. 1, pp. 16-28, February, 2001.
- [84] Z. Jihui, Z. Qian, L. Bo, L. Xiaonan, and Z. Wenwu, "Energy-efficient routing in mobile ad hoc networks: mobility-assisted case," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 1, pp. 369-379, January, 2006.
- [85] V. Srinivasan, C. F. Chiasserini, P. S. Nuggehalli, and R. R. Rao, "Optimal rate allocation for energy-efficient multipath routing in wireless ad hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 3, pp. 891-899, May, 2004.
- [86] L. Yeng-Zhong, X. Kaixin, H. Xiaoyan, and M. Gerla, "Hybrid landmark routing in ad hoc networks with heterogeneous group mobility," in *Proc. of 2nd International Symposium on Wireless Communication Systems (ISWCS'05)*, pp. 125-129, Siena, Italy, September 5-7, 2005.
- [87] H. Jiun-Long and C. Ming-Syan, "On the effect of group mobility to data replication in ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 5, pp. 492-507, September-October, 2006.