The Hong Kong Polytechnic University

Department of Computing

Design of Group-Oriented Protocols and Algorithms for Wireless Mobile Networks

By

CHENG HUI

A Thesis Submitted in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

July, 2007

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________(Signature)

CHENG Hui (Name of Student)

Abstract

In wireless mobile networks, mobile nodes normally form groups to coordinate their actions in a cooperative way. In this thesis, we develop group-oriented protocols and algorithms for two types of wireless mobile networks: mobile ad hoc network (MANET) and its Internet extension, i.e., iMANET. These protocols and algorithms together build the foundation of the research framework on group-oriented communication.

We classify mobile nodes into two types based on the distribution of group members. The first type is named as the aggregated group in which all group members stay together and move following a certain group mobility model. The second type is named as the distributed group in which group members are distributed in different locations in the network and need the support of group communication using multicast forwarding.

In an aggregated group, group mobility has significant effects on the design of network protocols and algorithms. In light of this, researchers have considered various issues in MANETs under the group mobility scenarios, e.g., multimedia streaming, network partition prediction, routing and data replication. We identify two properties of group mobility. The first property is the relative stability of distances between two neighboring nodes in the same group, which can be used to improve the overall system stability. Taking advantage of this property, we investigate the clustering problem in MANETs with group mobility, and propose a *S*tability-aware *M*ulti-*o*bjective *C*lustering algorithm (SMoC). In SMoC, the relatively stable topology is constructed first, and then multi-metric clustering is conducted with the help of a multi-objective evolutionary algorithm. SMoC shows better performance than the well known Weighted Clustering Algorithm (WCA). The second property of group mobility is motion affinity, meaning that the group can be regarded as a logical subnet. Using this property, we propose GrLS, a *Gr*oup-based *L*ocation *S*ervice protocol for MANETs with group mobility. In GrLS, only the group leader needs to recruit location servers and adaptively update its location to these servers while the other group nodes are exempted from these operations. Thus, the location update cost can be significantly reduced resulting in a dramatic protocol overhead reduction.

In a distributed group, members normally do not coordinate their movement but often need to communicate with each other. To support group communication, we need to develop multicast routing protocols which establish and maintain efficient multicast forwarding structures, meeting certain specified quality-of-service (QoS) requirements. We develop a hybrid multicast routing protocol for MANET, referred to as the *G*eography-aided *M*ulticast *Z*one *R*outing *P*rotocol (GMZRP). GMZRP discovers a multicast forwarding tree by on-demand *M*ulticast *R*oute *REQ*uest (MRREQ) propagation, which is guided by geographic information to reduce redundant transmission. For the iMANET heterogeneous network, we focus on the construction of a QoS-aware multicast tree in the backbone network and the integration of the effect caused by wireless transmission paths. We propose two algorithms, *D*elay and *D*elay *V*ariation *M*ulticast *A*lgorithm (DDVMA) and the improved CBT+SP (CBT: *C*ore-*B*ased *T*ree, SP: *S*hortest *P*ath) heuristic algorithm. Both algorithms can achieve better performance in terms of multicast delay variation than the existing well-known *D*elay and *D*elay *V*ariation *C*onstraint *A*lgorithm (DDVCA) under the same multicast end-to-end delay constraints.

Keywords: MANET; iMANET; group mobility; group communication; clustering; location service; multicast.

Publications

Journal Papers

1. **Hui Cheng**, Jiannong Cao, Xingwei Wang, "A Fast and Efficient Multicast Algorithm for QoS Group Communications in Heterogeneous Network", *Computer Communications* (Elsevier Science), Vol. 30, Issue. 10, pp. 2225-2235, 2007.

2. **Hui Cheng**, Jiannong Cao, Xingwei Wang, "A Heuristic Multicast Algorithm to Support QoS Group Communications in Heterogeneous Network", *IEEE Transactions on Vehicular Technology*, Vol. 55, Issue. 3, pp. 831-838, May 2006.

3. Xingwei Wang, Jiannong Cao, **Hui Cheng**, Min Huang, "QoS Multicast Routing for Multimedia Group Communications Using Intelligent Computational Methods", *Computer Communications* (Elsevier Science), Vol. 29, Issue. 12, pp. 2217-2229, 2006.

4. **Hui Cheng**, Jiannong Cao, Hsiao-Hwa Chen, "GrLS: Group-based Location Service in Mobile Ad Hoc Networks", accepted by *IEEE Transactions on Vehicular Technology* with minor revision.

5. **Hui Cheng**, Jiannong Cao, "A Design Framework and Taxonomy for Hybrid Routing Protocols in Mobile Ad Hoc Networks", accepted by *IEEE Communications Surveys and Tutorials*.

6. **Hui Cheng**, Jiannong Cao, Xingwei Wang, Sajal K. Das, "Stability-aware Multi-metric Clustering in Mobile Ad Hoc Networks with Group Mobility", *Wireless Communications and Mobile Computing*, under the 2nd round review.

Conference Papers

7. **Hui Cheng**, Jiannong Cao, Hsiao-Hwa Chen, "GrLS: Group-based Location Service in Mobile Ad Hoc Networks", *Proc. 42nd annual IEEE International Conference on Communications* (ICC 2007), June 2007, Glasgow, UK.

8. **Hui Cheng**, Jiannong Cao, Xingwei Wang, "Constructing Delay-bounded Multicast Tree with Optimal Delay Variation", *Proc. 41st annual IEEE International Conference on Communications* (ICC 2006), June 2006, Istanbul, Turkey.

9. **Hui Cheng**, Jiannong Cao, Xingwei Wang, Sajal K. Das, "Stability-based Multi-objective Clustering in Mobile Ad Hoc Networks", *Proc. The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks* (QShine 2006), August 2006, Waterloo, Ontario, Canada.

10. Jiannong Cao, Lifan Zhang, Guojun Wang, **Hui Cheng**, "SSR: Segment-by-Segment Routing in Large-scale Mobile Ad Hoc Networks", *Proc. 3rd IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (MASS'06), October 2006, Vancouver, Canada.

11. Xingwei Wang, Shuxiang Cai, **Hui Cheng**, Min Huang, Jiannong Cao, "A Fuzzy Decision Tree Based Mobility Prediction Mechanism in Mobile Internet", (Invited

Paper), *Proc. 1st International Conference on Wireless Broadband and Ultra Wideband Communications* (AusWireless 2006), March 2006, Sydney, Australia.

12. **Hui Cheng**, Jiannong Cao, Xingwei Wang, Srinivasan Mullai, “A Heuristic Multicast Algorithm to Support QoS Group Communications in Heterogeneous Network”, *Proc. The Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks* (QShine 2005), August 2005, Orlando, Florida, USA.

13. Jiannong Cao, Srinivasan Mullai, David Leung, **Hui Cheng**, “An Efficient Multicast Service Switching Protocol in Mobile IP”, *Proc. International Conference on Wireless Networks, Communications and Mobile Computing* (WirelessCom 2005), June 2005, Maui, Hawaii, USA.

Acknowledgements

I wish to thank a multitude of people who have been helping me. In particular, I would like to express my sincerest gratitude to my supervisor Prof. Jiannong Cao for his devoted guidance, constant encouragement, and invaluable suggestions, which helped me not only to complete this thesis but also to prepare me for my career in the long run. Prof. Cao is an outstanding researcher with broad knowledge, sharp intuition and grand vision. He has taught me to always have high expectations and demand more of myself. Working with him has been my invaluable and delightful experience. I have learned more from him than I can ever repay. I aim to hold myself to his high standards wherever I go.

I would also like to thank Prof. Xingwei Wang. He has supervised my Master study in Northeastern University, China. After I came to the Hong Kong PolyU, he still provides continuous encouragement, and cheers with me on every progress I have made.

I am indebted to my friends and colleagues Vaskar Ray Choudhury, Xiaopeng Fan, Joanna Siebert, Miaomiao Wang, Kirk Wong, Weigang Wu, Jin Yang, Gang Yao, Yu Zhou, and all other members in Prof. Cao's research group, for their insightful discussions and warm friendship.

Finally, I wish to thank my entire extended family for providing a loving environment for me. I would like to thank my wife, Yanyan Xi, for her love, understanding, patience, and encouragement when it was most needed. I am very grateful to my parents and my younger brother for their everlasting love and encouragement, and to Susan Cheng for being my other accomplishment.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADMR	Adaptive Demand-Driven Multicast Routing
AGP	Adjacently Grouped Pair
DDVCA	Delay and Delay Variation Constraint Algorithm
DDVMA	Delay and Delay Variation Multicast Algorithm
DLM	Distributed Location Management
DVBMT	Delay- and delay Variation-Bounded Multicast Tree
DVMRP	Distance Vector Multicast Routing Protocol
DSR	Dynamic Source Routing
EA	Evolutionary Algorithm
EGMP	Efficient Geographic Multicast Protocol
GA	Genetic Algorithm
GeoTORA	Geographic Temporally Ordered Routing Algorithm
GHLS	Geographic Hashing Location Service
GLS	Grid Location Service
GMP	Geographic Multicast routing Protocol
GMZRP	Geography-aided Multicast Zone Routing Protocol
GrLS	Group-based Location Service
HLS	Hierarchical Location Service
IZR	Independent Zone Routing
M-LANMAR	Multicast-enabled Landmark Ad Hoc Routing
MAODV	Multicast Ad hoc On-demand Distance Vector routing
MBC	Mobility-Based Clustering
MOEA	Multi-Objective Evolutionary Algorithm
MRREQ	Multicast Route REQuest
MRREP	Multicast Route REPly

MZR	Multicast routing protocol based on Zone Routing
MZRP	Multicast Zone Routing Protocol
ODMRP	On-Demand Multicast Routing Protocol
SPEA2	Strength Pareto Evolutionary Algorithm 2
VDPS	Virtual home region based Distributed Position Service
WCA	Weighted Clustering Algorithm
ZRP	Zone Routing Protocol

Chapter 1

Introduction

The main objective of this research is to investigate novel group-oriented protocols and algorithms for communication and cooperation in wireless mobile networks. In this thesis, a wireless mobile network refers to a mobile ad hoc network (MANET) [CHE04, MUR04, PER01, TOH02] or its Internet extension, i.e., iMANET [LIM04, LIM06, LIM07, TSE03]. Group-oriented protocols and algorithms developed can be classified into two types: novel protocols and algorithms in MANETs with group mobility, and multicast routing for group communications in both MANET and iMANET. This chapter provides an introduction to MANET and iMANET. We also describe the limitations of current research on group-oriented protocols, which motivate this research. Finally we summarize the contribution of this research work and outline the organization of this thesis.

1.1 Wireless Mobile Networks

There are two major types of wireless mobile networks [AFI03, BAT94, CHA03, ROB04]. One is base-station (BS) oriented [GAR02, MIS04, MUR01] and the other is mobile ad hoc network (MANET). In BS-oriented wireless networks, the mobile hosts (MHs) communicate with fixed base stations interconnected by a wired backbone. In this thesis, we use "mobile host" and "mobile node (MN)" interchangeably. BS-oriented wireless network is also called a single-hop (or cellular) network. MANET is a self-organizing and self-configuring multi-hop wireless network comprised of a set of MNs that can move around freely and cooperate in relaying packets on behalf of one another. MANET supports robust and efficient operation by incorporating routing functionality into MNs. In this thesis, we consider two types of wireless mobile networks: MANET and the iMANET heterogeneous network.

1) Mobile Ad Hoc Networks

An MANET has limited bandwidth availability and battery power, so the protocols and algorithms designed for it must conserve both bandwidth and energy. In addition, wireless devices usually use computing components- processors, memory, and I/O devices- that have low capacity and limited processing power. Therefore, the communication protocols should have lightweight computational and information storage needs [MOB04].

In an MANET, multi-hop forwarding paths are established for nodes beyond the direct wireless communication range. Routing protocols for MANETs must discover such paths and maintain connectivity when links in these paths break due to effects such as node movement, radio propagation, or wireless interference. So far, there are two major types of routing protocols in MANETs: topological routing and geographic routing. In topological routing [JOH96, MUR04, PER94, PER99, PER01, TOH02], mobile nodes utilize topological information to construct routing tables or search routes directly. In geographic routing [GIO03, MAU01, RAO03], each node knows its own position by location systems [HIG01] and makes routing decisions based on the destination's position and its local neighbors' positions.

2) Heterogeneous Network: iMANET

The explosive growth of mobile communications has attracted interests in the integration of wireless networks with wired networks and the Internet in particular. Providing Internet access capability for MNs in MANETs is necessary to pursue the dream of broadband wireless Internet access. Mobility of user devices connecting to the Internet is of major interest in today's research on networking [CHR03].

Tseng *et al*. [TSE03] proposed a heterogeneous network architecture which consists of multiple MANETs attached to the backbone Internet, as shown in Figure 1.1. This type of architecture extends the typical wireless access points to multiple MANETs, each as a subnet of the Internet, to create an integrated environment that supports both macro and micro IP mobility. This type of heterogeneous network has been formally named as iMANET [LIM04, LIM06, LIM07]. The iMANET architecture facilitates the current trend of moving to an all-IP wireless environment.

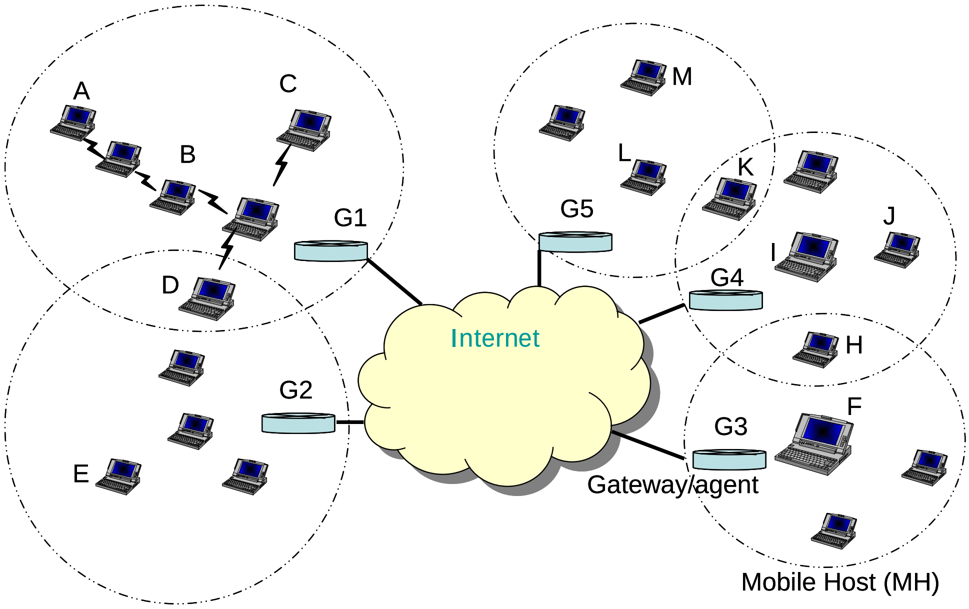


Fig. 1.1: The iMANET heterogeneous network architecture with five MANETs

1.2 Limitations of Current Research on Group-oriented Protocols

In wireless mobile networks, mobile nodes often form groups to work in a cooperative way. A group of nodes may aggregate together and move following a certain group mobility model [HON99, LI02]. A group of nodes may also be distributed in different locations of the network and they need the support of group communication using a multicast forwarding structure, e.g., a tree or a mesh [YAN05]. We name the former as the aggregated group and the latter as the distributed group. Accordingly, we name all the protocols and algorithms which are related to either the aggregated group or the distributed group, as group-oriented protocols and algorithms.

In reality, mobile nodes tend to show some degree of correlated (group oriented) motion behavior. Hence, in recent years, group mobility, where mobile nodes are organized into groups to coordinate their movement, has emerged from the demand of applications where a team of mobile users stay closely and move together. Examples include military and disaster recovery operations, vehicular communications, etc [GAL04, HUA06, LI03]. We name this type of group as an aggregated group.

This group mobility feature, once detected and understood in MANETs, can be exploited to help improve network performance, in particular, scalability [GER03]. We observe two properties from group mobility. The first one is the relative stability of distances between two neighboring nodes in the same group [LI02], which can be exploited to improve the stability of the existing protocols and algorithms. The second one is motion affinity, meaning that the group can be regarded as a logical subnet. Since the logical subnet shields the information of the internal group members from the outside nodes of the group, it can be exploited to reduce the protocol complexity and overhead.

In light of this, researchers have considered various issues in MANETs under the group mobility scenarios, e.g., multimedia streaming [LI03], network partition prediction [WAN02], routing [PEI00], and data replication [HUA06]. However, to our best knowledge, no works have addressed the stability-aware clustering and location service in MANETs with group mobility.

In wireless mobile networks, more and more collaborative applications appear, e.g., several military units or rescue teams form a group and work in different locations to accomplish the common tasks [LAW05, PRA98]. We name this type of group as a distributed group. Since the group members in a distributed group stay at different locations, efficient group communication [ANA01, BIR93] is required. Group communication needs the support of multicast routing which tries to deliver each message over each link on the multicast forwarding structure only once, creating copies only when the links to the destinations split.

When it became clear that group communication is one of the key applications in MANET environments, a number of multicast routing protocols, using a variety of basic routing algorithms and techniques, have been proposed [COR03, GAR99, YAN05]. Similar to unicast routing, multicast routing protocols can also be classified as either topological multicast routing (e.g., ADMR [JET01], FGMP [CHI98], MAODV [ROY99], ODMRP [LEE00]) or geographic multicast routing (e.g., EGMP [XIA06], GeoTORA [KO00], GMP [WU06]). Intuitively, hybrid multicast routing still has its advantage of combing the merits in both topological multicast routing and geographic multicast routing. However, to our best knowledge, few works have addressed the hybrid multicast routing in MANETs.

In the iMANET heterogeneous network, the group communication occurs when several MANETs distributed in different areas need to work cooperatively. It also involves one-to-many or multicast communication pattern. Different from MANETs, the group communication in the iMANET heterogeneous network consists of a two-tier communication architecture: the lower tier is the communication within each MANET and the higher tier is the communication between each MANET and its Internet gateway. To support group communication in the iMANET heterogeneous network, we need to address the multicast routing in the backbone wired network and the integration of the effect of the wireless transmission paths. In addition, the multicast routing in iMANET is expected to guarantee a set of QoS requirements such as delay, delay variation, etc. However, in this aspect, the current multicast routing algorithms can still be improved.

1.3 Contributions of the Thesis

We aim to design novel group-oriented protocols and algorithms to enable the operation of and support the cooperation among mobile nodes. In this section, we describe our contributions in this thesis, which consist of several group-oriented protocols and algorithms designed for both MANET and iMANET. These protocols and algorithms are not isolated work. All of them together build the foundation of the research framework on group-oriented communication in wireless mobile networks, as shown in Figure 1.2.

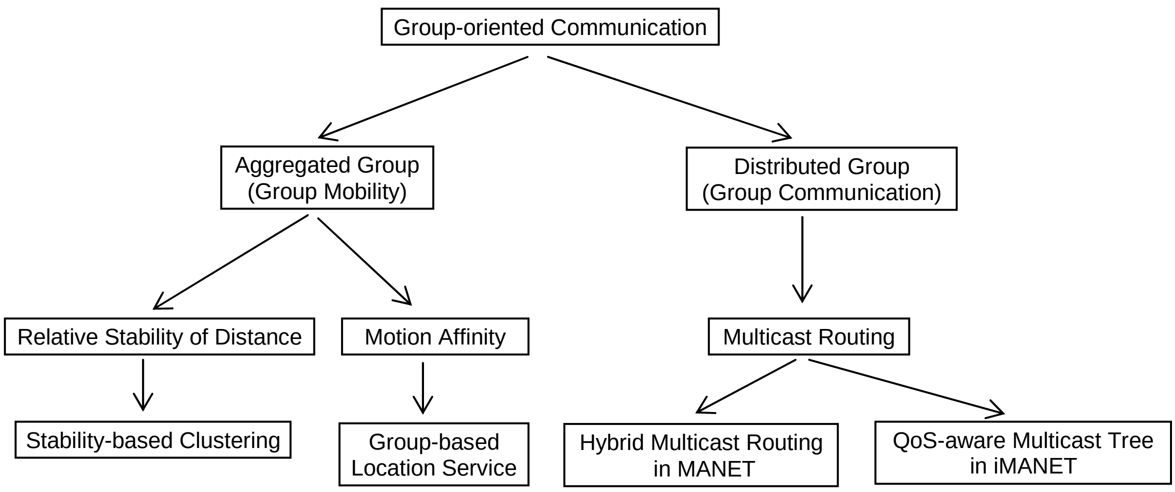


Fig. 1.2: The research framework on the group-oriented communication

We firstly classify the types of existing groups. For different types of groups, the research issues are also different. We reveal that there are normally two types of groups according to the distribution of group members. One is the aggregated group in which all the group members aggregate together and move as a whole. All the group members in an aggregated group follow the group mobility pattern. The other is the distributed group in which the group members stay in the network dispersedly. Group members in the distributed group do not coordinate their movement but coordinate the group communication. Hence, the protocols and algorithms to support group communication play the most important role for distributed groups.

To exploit group mobility in an aggregated group, we further discover two properties from it: the relative stability of distance between two neighboring group members, and motion affinity. The first property is used to improve system stability. The second property is used to reduce the protocol overhead when we regard each group as a logical subnet. To support group communication in a distributed group, we need to develop multicast routing. The primary work of multicast routing is to establish and maintain an efficient multicast forwarding structure. We also add QoS requirements (delay and delay variation) to the group communication in the iMANET heterogeneous network.

The thesis handles the above problems and makes the following contributions.

1) Taking advantage of the first property of group mobility, we propose a Stability-aware Multi-objective Clustering (SMoC) algorithm for MANETs. In SMoC, a mobile node joins the clusterhead which is its relatively stable neighbor. Thus, the lifetime of each cluster is prolonged and the stability of the clustering architecture is improved.
2) Taking advantage of the second property of group mobility, we propose a Group-based Location Service (GrLS) protocol for MANETs. In GrLS, within a group only the group leader needs to recruit location servers and update its location to these servers. All the other group members are exempted from the remote

location update. Thus, the location update cost is significantly reduced resulting in a dramatic protocol overhead reduction.

3) In MAENTs, we present a novel hybrid multicast routing protocol, referred to as the Geography-aided Multicast Zone Routing Protocol (GMZRP), which finds a multicast forwarding tree by on-demand Multicast Route REQuest (MRREQ) propagation. The MRREQ propagation is guided by geographic information to reduce redundancy.
4) In the iMANET heterogeneous network, we propose two algorithms, Delay and Delay Variation Multicast Algorithm (DDVMA) and the improved CBT+SP heuristic algorithm. Both of them construct QoS-aware multicast trees in the backbone network, which also consider the effect of wireless transmission paths. The QoS-aware multicast trees minimize the multicast delay variation under the multicast end-to-end delay constraint.

1.4 Outline of the Thesis

The remaining chapters of this thesis are organized as follows. Chapter 2 briefly presents the literature review of the relevant topics and provides some necessary background knowledge for works reported in this thesis. Chapter 3 presents the SMoC algorithm. Chapter 4 presents the GrLS protocol. Chapter 5 presents the GMZRP protocol. Chapter 6 presents the DDVMA algorithm and the improved CBT+SP heuristic algorithm. Chapter 7 gives the conclusions and a discussion of future works.

Chapter 2

Background and Literature Review

In this chapter, we provide a literature review and some background knowledge related to the research issues in this thesis. The organization of this chapter is as follows. Firstly, Section 2.1 gives an overview on clustering in MANETs. Section 2.2 describes the prevalent location service protocols. Section 2.3 describes group mobility, including its models and effects to MANETs. Finally, Section 2.4 describes multicast routing, the fundamental component in group communication.

2.1 Clustering in MANETs

Just like the Internet, a flat network structure encounters the scalability problem when the network size increases. Scalability is more challenging in MANETs due to node mobility. Therefore, efficient network management is extremely important. Analogous to the IP subnet concept, an MANET can also be organized into a hierarchical architecture by dividing nodes into clusters. Each cluster maintains and aggregates the information of the nodes within it. Each cluster can thus be seen as a logical node at the cluster level. The network layer only needs to maintain and manage the information of these logical nodes. Clearly, the control overhead can be reduced with the aid of clustering.

Clustering is defined as the method which attempts to organize unlabeled feature vectors into groups (clusters) such that points within a cluster are more similar to each other than to vectors belonging to different groups (clusters) [AN01]. A typical cluster structure for an MANET is shown in Figure 2.1. In a cluster, mobile nodes may play different roles, such as clusterhead, clustergateway, or clustermember. A clusterhead normally serves as a local coordinator for its cluster, performing intra-cluster transmission control, data forwarding, and so on. A clustergateway is a non-clusterhead node with inter-cluster links, so it can access neighboring clusters and forward data between clusters. A clustermember is an ordinary non-clusterhead node without any inter-cluster links.

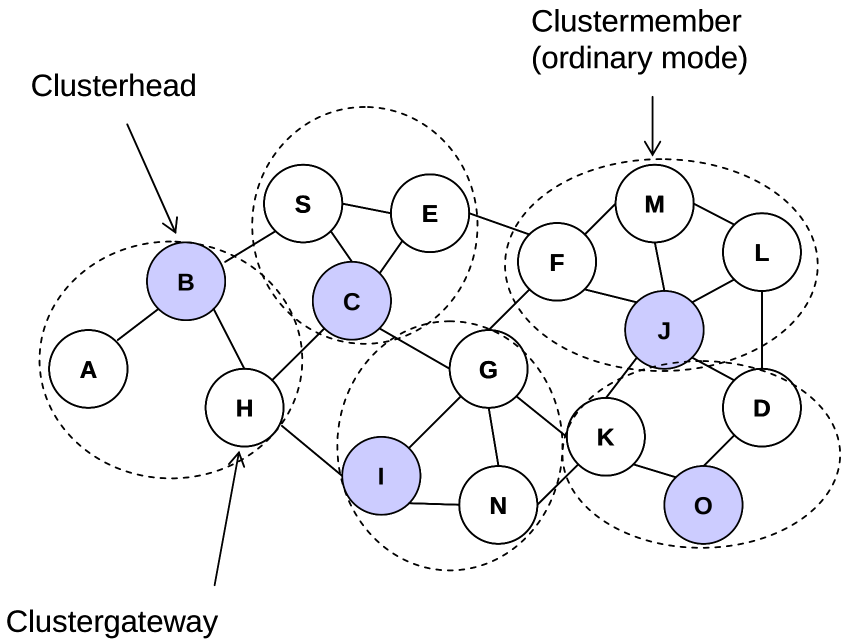


Fig. 2.1: Example of clustering in an MANET

In [CHA02], the clusterheads are supposed to work in dual power mode. They operate at a higher power mode (resulting in a higher transmission range) for inter-cluster communication and a lower power mode (resulting in a lower transmission range) for intra-cluster communication.

2.1.1 Clusterhead Selection

The primary step in clustering is the selection of clusterheads. The clusterhead can be the leader node, for example, the node with the maximum power. The selection is based on different criterion derived from specific communication requirements. For one-hop clustering, the cluster structure is determined once the clusterheads are determined. In the following, we formalize the clusterhead selection problem.

An MANET is represented as an undirected graph *G*=(*V*, *E*), where *V* represents the set of mobile nodes and *E* represents the set of links. *E* always changes with the creation and deletion of links. *N*(*v*) is the neighborhood of node *v*, defined as

$$N(v) = \bigcup_{v' \in V, v' \neq V} \{v' \mid dist(v, v') < r\}, \tag{2.1}$$

where *r* is the radio transmission range of node *v*.

The generalized procedure for selecting clusterheads is described below.

Step 1 From *G*, select one mobile node *v* as a clusterhead according to a certain rule.

Step 2 Delete node *v* and all its neighbors (all nodes in *N*(*v*)) from *G*.

Step 3 Repeat Steps 1-2 for the remaining nodes in *G* until *G* is empty.

The above three steps generate a set of clusterheads. In Step 1, the rule determines which node is selected as the clusterhead. Different clustering algorithm defines different rules, such as the lowest node-ID, the highest node-degree, the least node-weight, etc.

2.1.2 Mobility-based Clustering

A good clustering algorithm should be adaptive to node movement, that is, should not change cluster configuration drastically and often, when nodes are moving and/or the topology is slowly changing. Otherwise, high processing overheads for re-computation of clusterheads and communication overheads for frequent information exchange are paid. Moreover, the clusterheads lose control of their clusters and thus their roles as cluster managers.

Node mobility is an important characteristic of MANETs. In recent years, several application scenarios emerge, where mobile nodes are required to coordinate their movement. Further, some mobile nodes are organized into groups to move as a whole. Examples include vehicular area networks and personal area networks. Therefore, if the relative distance between two mobile nodes stays within the communication range, they appear stable with respect to each other. With group mobility, lots of mobile nodes move following the same mobility pattern. Hence, relative stability of distance can be exploited from group mobility to improve the system stability.

A mobility-based clustering (MBC) approach was proposed in [AN01]. In it, the relative mobility between any pair of nodes within time period T is defined as their absolute relative speed averaged over time T. Each node m selects node i as a tentative clusterhead. Node i has the lowest ID in m's L-hop neighborhood and the relative mobility between them is less than a pre-determined mobility threshold. If a tentative clusterhead, named as $TCH1$, is to be included in another cluster with clusterhead $TCH2$, then the child cluster will join the parent cluster together with all its current cluster members.

A mobile node can estimate its distance to one neighbor based on the received signal strength from that particular neighbor. The variation of the estimated distances between two nodes is observed over time. From the series of distance variations, the relative mobility pattern between two nodes can be predicted by statistical testing [ER04]. In

[LI02], the author also uses similarity of mobility patterns discovered over time to determine group membership.

2.1.3 Multi-metric Clustering

A well-known Weighted Clustering Algorithm (WCA), which optimizes a linearly combined weight consisting of four metrics, was presented in [CHA02]. It takes nodes with less mobility as a better choice for clusterheads. But this may not always be useful. Consider the case that all the nodes are moving rapidly except one slow-speed node, which lags behind. How can it play the role of the clusterhead? So instead of absolute mobility, relative mobility is a more reasonable metric. In addition, WCA specifies the values of weighing factors rather arbitrarily since it is hard to determine them precisely.

In [TUR02, TUR03], two intelligent optimization techniques- genetic algorithm (GA) and simulated annealing (SA), are used to optimize WCA such that the number of clusterheads is minimized while load in the network is as evenly balanced as possible among all the clusters. Both of these approaches optimize the WCA further, but they still use a weighted linear combination of the associated metrics. In other words, they still address the multi-metric clustering by single-objective optimization.

2.2 Location Service in MANETs

A challenging problem in geographic routing [BLA05, JAI01, KAR00, KO98] is how to provide location service so that a source node can obtain the location of the destination. A number of location service protocols have been proposed, including VPDS [WU05a], GHLS [DAS05], GLS [LI00], DLM [XUE01], and HLS [WOL04]. They can be divided into flooding-based and rendezvous-based approaches [DAS05]. In the flooding-based approach, the source floods the location query in the whole network. Clearly, the approach is simple, but not cost efficient. Therefore, most of the existing works focus on the rendezvous-based approach, in which any node can query the location of any other node from that node's location servers, called the rendezvous nodes. Rendezvous nodes record the location updates from mobile nodes and answer the location queries.

The rendezvous-based approach can be further divided into quorum-based and hashing-based [DAS05]. In quorum-based location service protocols, there are two quorums: update quorum and query quorum. These two quorums are designed in a way that they have non-empty intersection, so that the location query can be replied by the location servers lying in the intersection. An example of quorum-based location services is the column-row quorum-based protocol proposed in [STO99], and more methods to generate quorum systems can be found in [HAS99].

In hashing-based location service protocols, a publicly known hash function is always available. The input of the hash function is a node ID and the output can be either node IDs or geographic locations. The hash function is used to obtain the information about the location servers of any given node. There are two kinds of hashing-based protocols: hierarchical or flat. In hierarchical hashing-based protocols, the network coverage area is partitioned into hierarchical layers of subareas. Each node ID is hashed to the location servers residing in different subareas at different levels. In flat hashing-based protocols, the network coverage area is partitioned into different subareas without hierarchy. Each node ID is hashed to the location servers residing in one or more subareas. In the remainder of this section, we describe representative hashing-based location services in detail.

2.2.1 Hierarchical Hashing-based Location Service

Grid Location Service (GLS) [LI00] is a well-known hierarchical location service protocol. It partitions the network coverage area into a hierarchy of squares, and the smallest square is referred to as an order-1 square. In this hierarchy, an order-*n* square contains exactly four order-(*n*-1) squares. A node resides in one square at each hierarchy level. The other three squares at the same hierarchy level are the sibling squares. By the principle of the closest ID distance, a node recruits one location server in each sibling square at each hierarchy level. Hence, for each node, the density of location servers is high in the squares close to it and low in the squares far from it. Moreover, the nearby location servers are updated more frequently than remote location servers. When a source node needs to know the location of a destination node, it forwards the location query to the one whose ID has the least distance to the destination's ID, among all the nodes for whom it knows the locations. In this way, the location query process of GLS consists of a chain of nodes. Since nodes are moving, the node chain is unstable. As a result, GLS is very susceptible to node mobility. Moreover, the search for a node with the closest ID within a square is costly.

Distributed Location Management (DLM) [XUE01] is also a hierarchical hashing-based location service protocol. It partitions the entire network into a hierarchical grid. A hash function maps a node's ID directly to a set of minimum partitions in the network. The node recruits a location server in each minimal partition. In DLM, the location servers of a node are distributed in regions at different hierarchy levels. Different location servers may carry location information with different accuracy level. Only a small set of location servers needs to be updated when a node moves. DLM is scalable and robust to node mobility. The disadvantage of DLM is that the average query length is relatively large since only a small set of location servers can directly reply the location queries.

Hierarchical Location Service (HLS) [WOL04] is another hierarchical hashing-based location service protocol. The main idea of HLS is similar to DLM. The network coverage area is partitioned into cells, which are grouped hierarchically into regions of different levels. For a given node *A*, one responsible cell is selected at each hierarchy level by a hash function. An arbitrary node within or close to the responsible cell becomes *A*'s location server. A location server at level-*n* needs to be updated only when the node moves to another level-(*n*-1) region. So the location server at level-*n* only knows which level-(*n*-1) region the node is residing in. If a node *B* wants to determine the location of *A*, it queries the responsible cells of *A* in the order of the hierarchy until it receives a reply containing the current location of *A*. HLS is scalable and well suited for networks where communication partners tend to be close to each other. Since an indirect location scheme is used in HLS to reduce the cost of location updates, HLS has the same drawbacks as DLM.

2.2.2 Flat Hashing-based Location Service

Virtual home region based Distributed Position Service (VDPS) [WU05a] is a flat hashing-based location service system. In VDPS, each node is associated with a virtual home region (VHR), which is a geographic area. Nodes residing in a node's VHR function as its location servers at a probability. A VHR is further divided into subregions. A location update message arriving at the desired VHR is broadcast in each subregion to search the location servers and update them. The location query message is also broadcast in the subregions sequentially until it is received by a location server. Several approaches for improving the system robustness of VDPS are proposed and evaluated by detailed theoretical analysis. But the protocol overhead is high due to frequent message broadcast. In addition, the performance of VDPS is affected by node mobility because there is no handoff of location information when a location server leaves a VHR.

Geographic Hashing Location Service (GHLS) [DAS05] is another flat hashing-based location service protocol. Different from VDPS, the home region of a node consists of only one node whose location is the closest to the hashed location. A lightweight handoff procedure is introduced in GHLS. When a location server finds that another node is a better match for a subset of locations it stores, the location server hands off these locations to the new node. Another feature of GHLS is that it uses a hash function that generates locations within a scaled location server region near the center of the network. This can help alleviate a potential drawback of flat hashing-based protocols- a location server may be far away from both the source and destination nodes. Intuitively, a drawback of GHLS is that using a scaled location server region can create service load imbalance among the nodes in the whole network, i.e., higher load in the scaled region.

Compared to hierarchical hashing-based protocols, flat hashing-based protocols avoid the complexity of maintaining a hierarchy of grids and the consequent maintenance due to nodes moving across grid boundaries. GrLS proposed in this thesis is also a flat hashing-based protocol.

2.3 Group Mobility

Mobility is generally viewed as a major impediment in the control and management of large scale wireless networks. However, recently researchers have looked at mobility in a different way, trying to take advantage of it instead of protecting from it [GER03]. The most widely used individual-based mobility model is the random waypoint model [JOH96] where node movement is characterized by two factors: the maximum speed and the pause time. The model breaks the entire movement of a MH into repeating pause and motion periods. A MH first stays at a position for a certain time period then it moves to a new random-chosen position at a speed uniformly distributed between 0 and the maximum speed.

In reality, the motion behavior of mobile users is usually regular and follows some mobility patterns. In MANETs, some nodes may form groups and move together as a whole. This type of mobility pattern is named as group mobility and it is different from single mobility. For group mobility scenarios, members within the same group have similar mobility patterns. In each group, a mobile node may be designated or selected as the group leader, which serves as a gateway to other groups. Mobile nodes communicate with each other locally within the same group. Mobile nodes communicate with the nodes outside its group through the group leader. When a group member wants to communicate with one node outside its group, it first transmits the messages to its group leader. Then the group leader forwards the messages to the group where the destination locates.

Group mobility also brings some problems, such as network partition, network merging, and routing paths disruption [CHE03]. A few group mobility models were proposed in the previous literatures. In the following, we describe four representative ones.

2.3.1 Group Mobility Models

1) Reference Point Group Mobility (RPGM)

RPGM model was proposed by Hong *et al*. [HON99] and presents a general framework for group mobility. In this model, each group has a logical “center”. The center’s motion defines the entire group’s motion behavior, including position, speed, direction, acceleration, etc. Thus, the group trajectory can be determined once a path is provided for the center. Usually, nodes are uniformly distributed within the geographic scope of a group. Each node is assigned a reference point which follows the group movement. A node is randomly placed in the neighborhood of its reference point at each step. The reference point scheme allows independent random motion behavior for each node, in addition to the group motion.

The (x, y) physical positions of the group's logical center and its members are given by two levels of displacement vectors. The group motion vector $\overrightarrow{GM}$ maps out the position of the logical center, while the node-dependent random motion vectors $\overrightarrow{RM}$, added to the group motion vector, give the positions of the nodes. For a MH moving from time τ to $\tau+1$, first, the reference point of the node moves from $RP(\tau)$ to $RP(\tau+1)$ with the group motion vector $\overrightarrow{GM}$. Then the new node position is generated by adding a random motion vector $\overrightarrow{RM}$ to the new reference point $RP(\tau+1)$. Vector $\overrightarrow{RM}$ has its length uniformly distributed within a certain radius centered at the reference point and its direction uniformly distributed between 0 and 360 degree. This random vector $\overrightarrow{RM}$ is independent from the node's previous position.

The RPGM model defines the motion of groups explicitly by giving a motion path for each group. A path which a group follows is given by a sequence of check points defined along the path with respect to given time intervals. By proper selection of checkpoints, one can easily model many realistic situations, where a group must reach predefined destinations within given time intervals to accomplish its task. The checkpoint scenario file has the advantage of decoupling the mobility pattern from the model itself.

2) Reference Velocity Group Mobility (RVGM)

The preceding RPGM model has several disadvantages. The position information of nodes in a mobility group can only represent the physical position of nodes, but cannot predict the trend of the mobility and the change in network topology. The RVGM model was proposed by Wang *et al*. [WAN02]. It extends the RPGM model by proposing a velocity-based mobility model to improve it. This model is based on the observation that instead of geographic proximity, a more fundamental characteristic of a mobility group is the similarity of the member nodes' movements. The node movement can be characterized by the velocity $\mathrm{v} = (v_x, v_y)^T$, where v_x and v_y are the velocity components in the *x* and *y* directions.

This model is extended from RPGM model by proposing a velocity representation of the mobility groups and the mobile nodes. Each mobility group has a characteristic group velocity. The member nodes in the group have the velocities close to the characteristic group velocity but deviate slightly from it. Hence, the characteristic group velocity is also the mean group velocity. The membership of the *i*th node in the *j*th group is then described as:

- *Group velocity*: $\mathrm{W}_j(t) \sim \mathrm{P}_{j,t}(w)$ (2.2)
- *Local velocity deviation*: $\mathrm{U}_{j,i}(t) \sim \mathrm{Q}_{j,t}(u)$ (2.3)
- *Node velocity:* $\mathrm{V}_{j,i}(t) = \mathrm{W}_j(t) + \mathrm{U}_{j,i}(t)$ (2.4)

The RVGM model further extends RPGM model by modeling the group velocity and the local velocity deviation of the member nodes as random variables each drawn from the distribution $\mathrm{P}_{j,t}(w)$ and $\mathrm{Q}_{j,t}(u)$, respectively. The distributions can be arbitrary type, e.g., as Gaussian distribution, normal distribution, etc, in order to model the various mobility patterns that may exist for different mobility groups and for the nodes within a mobility group. Analogous to the RPGM model, the characteristic group velocity serves as a reference velocity for the nodes in the group.

The RVGM model has the following advantages. First, it directly provides the mobility parameters of each mobility group, such as the mean group velocity and the variance in the node velocities within the group. Second, by modeling the node velocities in a mobility group as a random variable with the distribution $Q_{j,t}(u)$, the group membership of any mobile node can be determined given the node velocity and the velocity distributions of the existing mobility groups.

3) Improved RPGM

The improved RPGM model was proposed by Li [LI02]. A major drawback of the preceding RPGM model is its assumption that all nodes have prior knowledge of group membership, i.e., they know which group they belong to, and that the group membership is static. However, this assumption is unrealistic. For example, when a new node is first introduced to an MANET, it has no prior knowledge about the reference points, or even which group it should belong to.

In the improved model, each node only has access to its local states, which include the distances to all its neighboring nodes. With this assumption, the RPGM model needs to be redefined so that it is characterized based on fully distributed states, e.g., distances between nodes, rather than the availability of a reference point. This new assumption is based on the intuition that nodes within the same group tends to have a high probability of keeping stable distances from each other.

In this model, it is assumed that all nodes have identical and fixed radio transmission range *r*. Then if the distance between two nodes $\|AB\| \le r$, they are named as in-range nodes that can communicate directly with a single-hop wireless link, denoted by $\overline{AB} = 1$. Otherwise, they are named as out-of-range nodes with $\overline{AB} = 0$, i.e., the distance between *A* and *B* is beyond the one-hop transmission range. If there exists a multi-hop wireless communication path between *A* and *B* interconnected by in-range wireless links, *A* and *B* are claimed as mutually reachable.

The author first defines the term AGP (Adjacently Grouped Pair) of nodes.

Definition 2.1 Nodes *A* and *B* form an Adjacently Grouped Pair (AGP), denoted by $A \overset{0}{\sim} B$, if $\|AB\|$ obeys normal distribution with a mean $\mu < r$, and a standard deviation $\sigma < \sigma_{\max}$, where $\|AB\|$ denotes the distance between *A* and *B*.

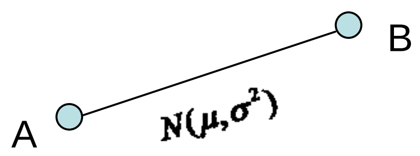


Fig. 2.2: Two nodes as Adjacently Grouped Pair

Figure 2.2 shows such a pair of nodes. Intuitively, this definition captures the fact that if two adjacent nodes are in the same group over a period of time, the distance between them stabilizes around a mean value μ with small variations, while $\mu < r$ so that they can communicate wirelessly.

Definition 2.2 Nodes *A* and *B* are *k*-related, denoted by $A \overset{k}{\sim} B$ $(k \ge 1)$, if there exist intermediate nodes $C_1, C_2, \cdots C_k$, such that $A \overset{0}{\sim} C_1$, $C_1 \overset{0}{\sim} C_2$, $\cdots$, $C_i \overset{0}{\sim} C_{i+1}$, $\cdots$, $C_k \overset{0}{\sim} B$.

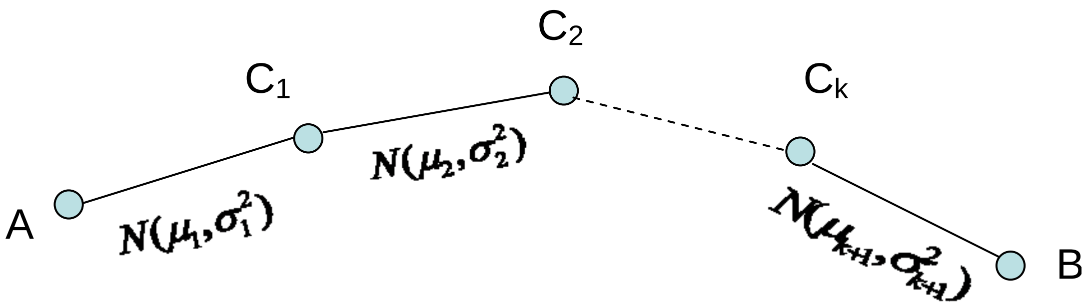


Fig. 2.3: *K*-related nodes defined by AGP relations

Figure 2.3 illustrates such a definition. Li [LI02] further defines nodes *A* and *B* as *related*, denoted by $A \sim B$, if either $A \overset{0}{\sim} B$ or there exists $k \geq 1$, such that $A \overset{k}{\sim} B$. Note that even if $A \sim B$, $\|AB\|$ does not necessarily obey normal distribution. In addition, it may be straightforwardly derived that the relation $A \sim B$ is both *commutative* (in that if $A \sim B$, then $B \sim A$), and *transitive* (in that if $A \sim B$ and $B \sim C$, then $A \sim C$). Based on the above definitions, the author formally defines the term *group* in the improved RPGM model.

Definition 2.3 Nodes $A_1, A_2, \cdots, A_n$ are in one group *G*, denoted by $A \in G$, if $\forall i, j$, $1 \leq i, j \leq n$, $A_i \sim A_j$.

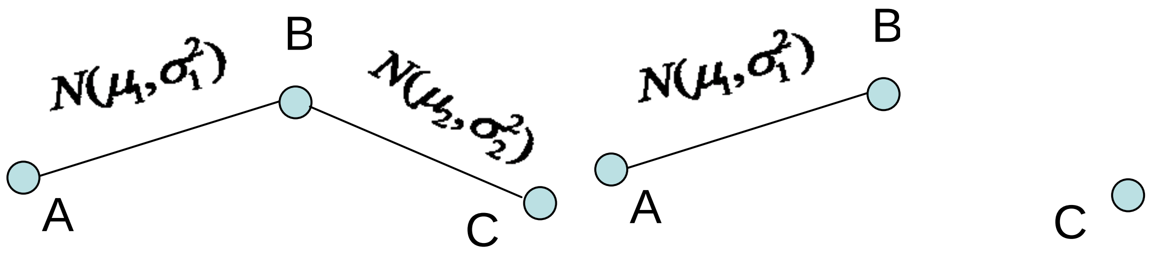
The following rules were derived in [LI02]:

- *if* $A \in G$ *and* $A \sim B$, *then* $B \sim G$.
- *if* $A \in G$ *and* $\neg(A \sim B)$, *then* $B \notin G$.
- *if* $A \in G_1$, $B \in G_2$ *and* $A \sim B$, *then* $G_1 = G_2$.
- *if* $A \in G_1$, $B \in G_2$ *and* $\neg(A \sim B)$, *then* $G_1 \neq G_2$.

- *if* $A \in G_1$ *and* $A \in G_2$, *then* $G_1 = G_2$.

These properties ensure that groups defined by Definition 2.3 are disjoint sets of nodes in an MANET. Note that the novelty of Definition 2.3 is that group memberships are determined by relative stability of distances (or similarity of mobility patterns) discovered over time, not geographic proximity at any given time. This rules out the misconception that as long as *A* and *B* are neighboring nodes, they belong to the same group.

Figure 2.4 gives an example. Figure 2.4(a) shows that $A \overset{0}{\sim} B$ and $B \overset{0}{\sim} C$, hence $A \sim B \sim C$, which forms one group {*A*, *B*, *C*}. In comparison, Figure 2.4(b) shows only $A \overset{0}{\sim} B$. In this case, although *A* and *C* (or *B* and *C*) are neighboring nodes, $A \sim C$ (or $B \sim C$) does not hold. Thus there have two disjoint groups {*A*, *B*} and {*C*}. This scenario may arise when two groups are briefly merged geographically but separated again, due to different directions of travel.



Group: {A, B, C}
(a)

Groups: {A, B}, {C}
(b)

Fig. 2.4: Grouping node

4) Improved RVGM

This model was proposed by Chen *et al.* [CHE03]. In this model, it takes the variation in velocity into consideration. With the variation in velocity, the movement of nodes can be represented more accurately. The variation in velocity is useful to predict the network partition time and to solve the mobility problem.

In this model, the variation in velocity over time is called *acceleration*, which changes with time and is dependent on the velocity. Each group has a characteristic group velocity $\mathrm{V}_i(t)$ and acceleration $\mathrm{A}_i(t)$. Group velocity is also the mean group velocity of nodes within a group. The member nodes in the group have velocities close to the group velocity but deviate lightly from it. Group velocity $\mathrm{V}_i(t)$, group acceleration $\mathrm{A}_i(t)$ and local velocity acceleration $\mathrm{L}_{i,j}(t)$ are random variables and change with time. The distribution of the random variables can be arbitrary type. The distributions should be able to reflect the real world scenarios of mobile ad hoc networks. In order to model the various mobility patterns, many types of distributions may exist for the nodes within a mobility group. For example, the velocity of j_{th} node in the i_{th} group is described as follows.

- *Node velocity:* $$\mathrm{N}_j(\mathrm{t}) = \mathrm{V}_i(0) + \mathrm{A}_i(\mathrm{t}) * \mathrm{t} + \mathrm{L}_{i,j}(\mathrm{t}) \qquad (2.5)$$

Where $\mathrm{V}_i(\mathrm{t})$ represents group velocity, $\mathrm{A}_i(\mathrm{t}) = d\,\mathrm{V}_i(\mathrm{t}) / d\,\mathrm{t}$ represents group acceleration, and $\mathrm{L}_{i,j}(\mathrm{t})$ represents the local velocity deviation.

The RVGM model is further extended by modeling the group acceleration $\mathrm{A}_i(t)$. The acceleration acts as a random variable and depends on the group velocity. With the acceleration, the mobility model can represent more accurate relationship of velocities between nodes in a group and the partition prediction scheme can predict the partition time more accurately.

This model has the following advantages. First, it can provide mobility parameters of each mobility group, including the mean group velocity, the node velocity and the acceleration of each group. Second, with the new mobility parameter, acceleration, the model can accurately determine the group membership of any mobile node. Third, by adding the acceleration parameter, the network partition time can be predicted accurately.

5) Group Membership

An important thing for group mobility is to determine the group membership of each node, thereby each node can know which group it belongs to. Then it can follow the mobility pattern of the group. Group membership can be static or dynamic. For static group membership, each mobile node is assigned a Group ID first. When two nodes notice that they are within the direct transmission range, they exchange the information including Group ID. If they hold the same Group ID, they know that they belong to the same group. For dynamic group membership, mobile nodes are not assigned a predefined Group ID. The group membership is determined by the similarity among the behaviors of the mobile nodes. The behaviors can be represented by the velocity, the velocity deviation, the stability of distances, etc.

The methods for determining the group membership used in the above four group mobility models are summarized as follows.

- *RPGM: static group membership, i.e., a predefined Group ID.*
- *RVGM: determined by the similarity of the member nodes' velocities. There exist both group velocity distribution and the local velocity deviation distribution for a mobility group.*
- *Improved RPGM: determined by the relative stability of distances (or similarity of mobility patterns) discovered over time. For example, the distance between two neighboring nodes in the same group should conform to normal distribution.*

- *Improved RVGM: determined by the similarity of the member nodes' velocities. There exist both group acceleration distribution and the local velocity deviation distribution for a mobility group.*

2.3.2 Effect of Group Mobility

As discussed in Chapter 1, group mobility brings specific effects to the existing protocols and algorithms in MANETs. Researchers have reconsidered a variety of issues in MANETs under the group mobility scenarios, e.g., network partition prediction [WAN02], data replication [HUA06], routing [PEI00], and multimedia streaming [LI03]. In the following, we take the network partition prediction and data replication as examples to show the effect of group mobility.

1) Network Partition Prediction

Network partition, a large-scale topology change, would occur unexpectedly and disrupt the on-going routing paths as well as application connections in both wired networks and wireless networks [CHE03]. However, in MANETs, clearly network partition occurs more frequently due to node movement or energy exhaustion. To predict the partitions is a very useful feature that can be provided to applications.

In MANETs, group mobility leads to more severe network partition. For example, when many mobility groups exist in the network, the distinct mobility pattern of each group causes them to separate, and network partition eventually occurs. The sub-networks resulted from the network partition are the different mobility groups. Hence, the prediction of network partition should utilize the knowledge of group mobility.

In [WAN02], based on the RVGM model, the authors investigated the effect of group mobility to the network partition prediction problem. To simplify the problem, the following assumptions are made. First, each mobility group occupies a circular coverage area of diameter *D* wherein the mobile nodes are uniformly distributed. Second, both the group velocity and node velocity are time-invariant. Based on these two assumptions, the network topology can be viewed as a collection of equal sized "circles" that are initially stacked on top of each other. For network prediction, the time, at which the "circles" completely uncover each other using the velocity of each "circle", can be calculated. In the following, the partition prediction algorithm is described.

For example, a simple network consisting of only two groups C_j and C_k, moving at velocity W_j and W_k., respectively. Assume C_j is stationary, the relative velocity W_{jk} between C_j and C_k is:

$$W_{jk} = W_k + (-W_j) \tag{2.6}$$

and

$$W_{jk} = (w_{jk,x}, w_{jk,y}) \tag{2.7}$$

where $w_{jk,x} = w_{k,x} - w_{j,x}$, $w_{jk,y} = w_{k,y} - w_{j,y}$.

Without loss of generality, assume groups C_j and C_k completely overlap with each other initially. Obviously, the time required for C_j and C_k to change from total overlapping to complete separation should be:

$$T_{jk} = \frac{D}{\sqrt{w^2{}_{jk,x} + w^2{}_{jk,y}}} \tag{2.8}$$

Hence, given the mean group velocities, the time of separation T_{jk} can be calculated for any pair of mobility groups. As a result, the occurrence of network partition can be predicted as a sequence of the expected time of separation T_{jk}'s between the various pairs of mobility groups in the network.

However, the two assumptions based on the RVGM model were criticized by Chen *et al.* [CHE03]. First, they think the assumption that the coverage of a group is a kind of circular coverage area seems unmatched to the real situation. Since each group has different number of members and node distribution, it can not be concluded that the number of group members is in direct proportion to the coverage occupied by a group. A narrow coverage may have larger number of nodes than a wide coverage due to the tight distance between each other. Under such conditions, the above predicted scheme may fail. Second, in real-life applications, it is not practical to take the velocity as an invariant because groups do not always move at a fixed velocity.

Based on the above observations, Chen *et al.* [CHE03] proposed both the improved RVGM model and another partition prediction scheme, in which each group has to exchange mobility parameters to each other to define the relationship between the groups overlapping with it. Each group uses the mobility information received from other groups to calculate the relative velocity and acceleration.

In this partition prediction scheme, the acceleration is used since the improved RVGM model uses the acceleration to represent the motion behavior of a group. They also use the same example to demonstrate their partition prediction scheme. Group C_j overlaps with group C_k and the distance for them to change from overlapping to complete separation is also D. The velocity of group C_k that is relative to group C_j is W_{jk}. The acceleration of group C_k that is relative to group C_j is A_{jk}. With these mobility parameters, the amount of time for group C_k and group C_j to change from overlapping to complete separation is revised as follows:

$$T_{jk} = -W_{jk,0} \pm \sqrt{W_{jk,0}{}^{2} + 2A_{jk}(D - D_0)} \Big/ A_{jk} \qquad (2.9)$$

When the relative acceleration, A_{jk}, is 0, Equation 2.9 is just the same as Equation 2.8 and the partition prediction time is also the same. When the relative acceleration is not 0, this scheme can calculate more accurate partition time than the scheme based on the RVGM model.

It is concluded that this partition prediction scheme has three advantages. First, the acceleration is taken into consideration for partition prediction. With acceleration, the partition time can be predicted accurately. Second, the prediction time changes only when the acceleration changes. Hence, the partition prediction time will not be recalculated unless the acceleration changes. The system cost paid on calculation can be reduced. Third, depending on the proposed group detection technique, the coverage occupied by a group does not need to be fixed.

2) Data Replication

In distributed database system, data replication is a promising technique to improve data accessibility and system performance [HUA06]. In MANETs, the issue of data replication has also attracted the researchers' attention due to its potential applications. In the future, a mobile device may have stronger storage capability to store data items. In many collaborative works, mobile devices need to access the data items stored in other mobile devices. However, since each node in an MANET can move freely, the network topology often changes dynamically and disconnection occurs frequently. When an MANET is separated into several disconnected partitions, the data accessibility is hence reduced. To alleviate this problem, efficient and effective data replication is required. Clearly, the data replication scheme should also consider the effect of node mobility. Intuitively, group mobility has more specific effect on data replication than single mobility since a group of nodes tends to move together.

In view of this, Huang *et al.* [HUA06] proposed a replica allocation scheme DRAM (standing for Decentralized Replica Allocation with group Mobility) to allocate replicas by considering group mobility. The employed group mobility model is RPGM and the movement of each group follows a waypoint model. DRAM has two major phases, the allocation unit construction and replica allocation phases.

The first phase is allocation unit construction phase. For an allocation unit, it has been formally defined as follows.

Definition 2.4 An *allocation unit* is a set of mobile nodes which share their storage and do not store repeated data item unless all data items have been allocated in this allocation unit.

By this definition, the number of non-repeated data items allocated in one allocation unit is expected to be as large as possible. Hence, each allocation unit should contain as many mobile nodes as possible. Moreover, the connectivity of the allocation unit should be maintained as stable as possible to avoid the severe degradation of data accessibility caused by disconnected partitions. Hence, it is important to construct large and stable allocation units.

Each node is assumed to be equipped with a GPS device and, hence, the position of the node is always available. Each node records a certain amount of historical location information, from which the motion behavior can be inferred. Five states are defined for mobile nodes: INITIAL, ZONE-MASTER, ZONE-MEMBER, CLUSTER-MASTER, CLUSTER-MEMBER. For each mobile node M_i, a broadcast zone is defined, which is a set of mobile nodes whose distances to M_i are smaller than or equal to a predefined TTL. A mobile node in the INITIAL state first broadcasts messages containing its host ID and motion behavior to all other nodes in its broadcast zone. If M_i in the INITIAL state finds its host ID is just the smallest one among all nodes within its broadcast zone, M_i will enter the ZONE-MASTER state. Then all the other nodes in the zone enter ZONE-MEMBER state. Each node M_i in the ZONE-MASTER state then clusters its member nodes by a decentralized clustering algorithm named as VectorCluster, which clusters mobile nodes with similar motion behavior into the same mobility groups. VectorCluster is composed of two major procedures, ClusterByAngle and ClusterByLength, which are developed in accordance with two different heuristics derived from the RPGM model.

After the clustering algorithm is executed, for each resulting cluster each zone master selects one cluster master, which then enters the CLUSTER-MASTER state. Then the other nodes in the same cluster enter CLUSTER-MEMBER state. Since some cluster members may change their motion behaviors, the resulting clusters are maintained in an adaptive manner, which can generate relatively low network traffic. In addition, clusters, which are likely to connect with one another later, are merged into an allocation unit to save the aggregate storage cost.

After the allocation unit construction phase, DRAM enters the replica allocation phase. The employed replica allocation algorithm adopts greedy scheme. For each allocation unit, all data items are replicated according to their allocation weights in this unit. Due to the frequent topology change, DRAM is executed periodically to adapt the replica allocation according to the network connectivity.

2.4 Multicast Routing

In this thesis, we investigate the multicast routing for group communication in both MANET and iMANET. Therefore, in this section, we review the related work and preliminary knowledge on multicast routing in different network scenarios. In mobile ad hoc networks, we develop a multicast routing protocol, GMZRP, which is partially motivated by Zone Routing Protocol (ZRP) [HAS01]. In the iMANET heterogeneous network, we develop algorithms to construct QoS-aware multicast trees. Hence, firstly, we introduce ZRP and its multicast extension. Then we introduce the background knowledge on QoS and multicast tree.

2.4.1 ZRP and Its Multicast Extension

1) ZRP

To the best of our knowledge, ZRP is the first one to propose the concept of routing zone. A node's routing zone is defined as a collection of nodes whose minimum hop distance from the node in question is no greater than a parameter referred to as the zone radius [HAS01]. Each node maintains its own routing zone and proactively maintains routes to destinations within its routing zone. An important consequence is that the routing zones of neighboring nodes overlap. In [SAM04], Haas *et al.* further enhanced the Zone Routing framework by developing an Independent Zone Routing (IZR), in which each node can independently configure its own zone radius based on local measurements only.

In ZRP, when a node bordercasts a query, the node's entire routing zone is effectively covered by the query. However, since neighboring routing zones heavily overlap with each other, excess route query traffic is generated as a result of query messages returning to covered zones. Thus, the design objective of query control mechanisms should be to reduce route query traffic by directing query messages outward from the query source and away from covered routing zones, as shown in Figure 2.5. In [HAS01], ZRP is enhanced with a collection of query control mechanisms to generate less control traffic than purely proactive route information exchange or purely reactive route discovery do. The query control mechanism includes Query Detection (QD1/QD2), Early Termination (ET), and Random Query Processing Delay (RQPD). However, since all these mechanisms are based on topological information, they cannot solve the problem completely.

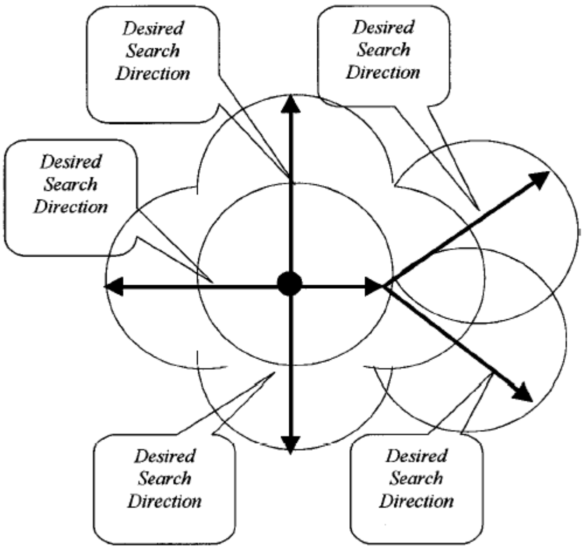


Fig. 2.5: Guiding the route search along different directions

With the ever-increasing advancement in location systems [HIG01], mobile nodes can easily obtain their own positions with high accuracy by indoor or outdoor location technology. In addition to mobile nodes' positions, the partition of the network coverage area is also widely exploited in geographic routing. It is observed that geographic information can significantly improve the routing performance in MANETs. For ZRP route query, an idea case is that the desired search along different directions passes through different nodes. If we partition the network coverage area into small zones and let each route search goes through different zones, the duplicate route queries can be avoided. Hence, geographic information can help a lot in the guidance of the route search.

2) Multicast Extension to ZRP

ZRP has been extended to multicast scenarios in some prior works, e.g., MZR [DEV01] and MZRP [ZHA04]. MZR stands for Multicast routing protocol based on Zone Routing. It is also a source-initiated on-demand protocol, in which a multicast delivery tree is created using a concept called the zone routing mechanism. The multicast tree creation is done in a two-stage process. The source initially forms the tree inside its zone by a proactive protocol running inside each zone. Then the source tries to extend the tree to the entire network by identifying all the border nodes in its zone and sending a TREE-PROPAGATE message to each one of them. Each of the border nodes repeats the same operation. This procedure is similar to the route discovery in ZRP. Hence, MZP has the same problem as ZRP in query control. Both MZR and GMZRP belong to the family of source tree based multicast protocol.

MZRP stands for Multicast Zone Routing Protocol. In MZRP, if one node finds that it is a multicast tree member, it will broadcast multicast tree membership messages within its local routing zone. Thus, nodes keep track of the groups and group members within their local routing zones. For a node wishing to join a multicast group, it firstly checks if it has a valid route to the multicast tree (i.e., to any node on the tree). If so, it sends a unicast MRREQ along the route to the multicast tree and waits for a Multicast Route REPly (MRREP) packet. Otherwise, the node initiates a bordercast MRREQ, which is sent out via the bordercast tree of the node. When the bordercast MRREQ reaches the peripheral nodes, the same procedure is repeated. Clearly, MZRP has also utilized the same method as ZRP to discover the route from a candidate group member to the multicast group. Therefore, MZRP does not improve the route discovery mechanism in ZRP.

2.4.2 Construction of QoS-aware Multicast Trees

In the Internet, multimedia group communications [GER01, TOJ04, YAN01] become an important research topic, which is driven by more and more popular multimedia collaborative applications such as video conference, content distribution, and distributed games. In wireless mobile networks, these and other uses are expected to also be important. In multimedia group communications, a source is required to send multimedia information to multiple destinations through a communication network. Real-time and fair delivery of multimedia data from the source to all the destinations is often required. To efficiently support QoS multimedia group communications, the most important issue that needs to be addressed is QoS multicast [JIA97, KUI02, LOW00, TYA03, WAN00, XUE03]. An efficient QoS multicast algorithm should construct a multicast routing tree, by which the data can be transmitted from the source to all the destinations with guaranteed QoS.

1) QoS

QoS requirements [XIA99] are often versatile in multimedia applications. For example, each link on the routing tree should guarantee a minimum available bandwidth when multimedia data flows are transmitted in a video conference application. End-to-end delay is also an important QoS parameter in data communications to guarantee that the messages transmitted by the source can reach the destination(s) within a certain period of time. For example, in an on-line game, users must receive fresh images and sounds in a very short time and they cannot tolerate delay of even several seconds. In video-on-demand (VoD) applications, the transmission delay of the first video frame from the sender (source) to one receiver (destination) should not exceed a certain value; otherwise, the user will feel that there is an obvious waiting procedure for the coming video.

Another important QoS parameter, multicast delay variation, is defined as the difference between the maximum and the minimum multicast end-to-end delays on the multicast tree. Real-time delivery of multimedia data is often desired. It evaluates the consistency and fairness of receiving messages among all the destinations. For some multimedia applications, the need for bounded multicast delay variation arises. For example, during a real-time streaming of stock quotes, the messages should be received at different destinations simultaneously, otherwise, the destinations who receive the messages later probably lose the chance of making profit. During a video conference, it is necessary that the speaker is to be heard by all audiences at the same time, otherwise, the communication may lack the feeling of an interactive face-to-face discussion. When multicast messages are used to update multiple copies of a replicated data item in a distributed database system, the less the delay variation, the shorter the length of time during which the database is in an inconsistent state. Driven by the application, many research efforts are devoted to the problem of delay or delay variation constrained multicast [CHE06, GEO97, LOW00, SHE02].

In addition to QoS, the tree cost, used to evaluate the utilization of network resource, is also a deterministic metric for evaluating multicast trees. A minimum-cost multicast tree [XUE03] can transmit multicast messages with the least utilization of network resources such as node’s CPU, buffer and link bandwidth.

2) Multicast Routing Tree

Multicast routing trees can be classified into two types, i.e., Steiner minimum tree (SMT) [HWA92] and shortest path tree (SPT) [NAR00]. An SMT is also the minimum-cost multicast tree. SPT is constructed by applying the shortest path algorithm to find the shortest (e.g., minimum cost or delay) path from the source to each destination and then merging them. The problem of finding an SMT has been proved to be NP-Complete [JIA97] and lots of approximation algorithms [AHA98, ESB95, HEL00, KAP93, ROB00] have been developed. SPT provides a good solution for finding delay-constrained multicast tree because it determines the minimum delay path from the source to each destination. Inspired by SMT and SPT, many heuristic algorithms [JIA98, KHU95, PAR98] have been proposed to construct a QoS-aware multicast tree by making a tradeoff between them. QoS multicast routing is still a challenging problem due to its intractability and comprehensive application backgrounds. The research on it has lasted for decades and is still going on [CHA04, LEE05, LI04, ORD05, SIA05, ZAP04].

Intelligent computational method [PAH03] is a type of promising technique to solve combinatorial optimization problems [PAP98] including the SMT problem. Genetic algorithm [GEN00], simulated annealing algorithm [LEV04] and Tabu search (TS) [GLO89] are three representative intelligent computational methods. Some previous studies [CUI03, HAB02, LEU98, MIC99, ZHA99] were done to apply them to solve the problem of unicast and multicast routing. As one of our previous works [WAN06], we also developed a unified framework for achieving QoS multicast trees using intelligent computational methods and proposed three QoS multicast algorithms based on genetic algorithm, simulated annealing, and Tabu search, separately. But this is not reported in this thesis.

Chapter 3
Stability-aware Multi-metric Clustering in Mobile Ad Hoc Networks with Group Mobility

In this chapter, we introduce the proposed *S*tability-aware *M*ulti-*o*bjective *C*lustering algorithm, SMoC. This chapter is organized as follows. Firstly, a brief overview is given in Section 3.1. Section 3.2 provides the preliminary knowledge on multi-objective evolutionary optimization. Section 2.3 describes the system model. The design of SMoC is presented in Section 3.4. Performance evaluation is conducted in Section 3.5. Finally, Section 3.6 summarizes this chapter.

3.1 Overview

A clustering algorithm [YU05] is to find a feasible interconnected set of clusters covering the entire set of nodes in MANET. At any instant, one mobile node can only belong to one cluster. A cluster may have a clusterhead or not. Since the recruiting of clusterheads brings the advantage of easy management, most of the prior research is on clustering with clusterhead. In this chapter, our algorithm also generates the clusters with clusterheads assigned.

Despite the fact that node mobility is an intrinsic characteristic in MANETs, the cluster structure should be maintained as stable as possible. Otherwise, frequent cluster change or re-clustering adversely affects the performance of radio resource allocation and scheduling protocols. By stability, we mean that the cluster structure remains unchanged for a given reasonable time period. Clearly, stability is an important requirement on a clustering algorithm. To maintain stable cluster structure, node mobility and group mobility [LI02] must be investigated. Group mobility has emerged from applications where a team of mobile users stay closely and move together. Mobile nodes are organized into groups to coordinate their movement. Examples include military and disaster recovery operations, vehicular communications, etc. Although existing work [AN01, ER04] addressed the relative mobility, yet the effect of group mobility on clustering has not been studied.

Furthermore, clustering must be associated with one or more metrics such as node ID, node degree, and energy (battery power), which are defined based on the application requirements. Early work in the literature has focused on single metric clustering. For example, in the highest-degree heuristic [GER95], the node with the maximum number of neighbors (highest-degree) is chosen as the clusterhead. But for a complex system like MANET, single metric is far from reflecting the whole network dynamics. Clustering algorithms optimizing only one metric commonly lose generality and have low performance in terms of other metrics.

Multi-metric clustering aims to create a cluster structure that optimizes several metrics simultaneously. Some existing works [CHA02, TUR02, TUR03] considered multi-metric clustering, but adopted the traditional method of linear combination (weighted sum) of multiple metrics. It is known that a single scalar objective function on ad hoc basis not only makes the solution highly sensitive to the chosen weight vector but also requires the user to have some knowledge about the priority or influence of a particular objective parameter over another [ROY04]. For multi-metric clustering, the same problem occurs because different metrics measure different capabilities of mobile nodes. Moreover, the evaluation criterion is different for different metrics. Hence, it is difficult to determine the weighing factors for the metrics in the linear combination formula. If an algorithm uses the weighted sum as a single metric, in our opinion, it is a single-metric clustering approach since it results in only one final solution. This solution cannot always optimize all the metrics simultaneously.

In this chapter, we propose a stability-aware multi-metric clustering algorithm for MANETs with group mobility. The motivation comes from the property of group mobility: *the distances between two neighboring nodes in the same group exhibit the relative stability*. To exploit this property, we define the concept of relatively stable neighbors, and based on it construct a relatively stable network topology. Then we run the multi-metric clustering procedure on the relatively stable topology to achieve stable clusters. Hence, the proposed clustering algorithm considers both stability and multi-metric optimization.

We define three clustering metrics as optimization objectives: total node degree differences, total power consumption, and minimum remaining battery lifetime. They respectively represent three important requirements for clustering: load balance, energy efficiency, and maximum lifetime. Our algorithm adopts a promising multi-objective evolutionary algorithm (MOEA), called Strength Pareto Evolutionary Algorithm 2 (SPEA2), that provides Pareto-optimal solutions with elaborate problem-specific design and modification [ZIT02]. We conduct simulations to evaluate the performance in terms of stability and multi-metric optimization. The results show that our proposed algorithm can generate stable cluster structures and high-quality clusterhead sets regarding all the clustering metrics.

Recently, MOEAs have been extensively used in research on networking, e.g., mobile multicast [ROY04], RSVP performance evaluation [KOM05], and so on. To our best knowledge, the proposed clustering algorithm is the first to optimize multiple metrics based on MOEA. It can produce a set of good solutions instead of a single solution to meet the requirements of multi-metric clustering.

3.2 Multi-objective Evolutionary Optimization

Conventional search techniques, such as hill-climbing, are often incapable of optimizing non-linear multimodal functions. In such cases, a random search method might be required. Evolutionary algorithm (EA, also called genetic algorithm) is a well-known guided random search and optimization technique. It is based on the basic principles of evolution: survival of the fittest and inheritance. Generally, EA is applied to find an approximate optimal solution with respect to a fitness function for NP-hard problems.

Many real-life optimization problems have multiple objectives. In such optimization problems, the objectives often conflict across a high-dimensional problem space. Solving these problems is generally very difficult and may require extensive computational resources. The presence of multiple objectives in a problem, in principle, gives rise to a set of compromised solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution. The definition of Pareto-optimal is as follows [SRI95].

Definition 3.1 A point x^* is Pareto-optimal if for every x either $\bigcap_i (f_i(x) = f_i(x^*))$ or there is at least one i such that $f_i(x) > f_i(x^*) \, \forall i \in I$ (set of integers), where $f_i(x)$ is the fitness function. In other words, x^* is Pareto-optimal if there exists no feasible vector x which would decrease some criterion without causing a simultaneous increase in at least one other criterion.

Solution *A* is said to dominate solution *B* if *A* is better than *B* in at least one objective value and is no worse in all other objective values. A Pareto-optimal solution is called a non-dominated solution. Table 3.1 gives a simple example to explain it. There are three solutions *A*, *B*, *C*. Each solution has three objective values. Suppose that the less the objective value, the better it is. In our example, *A* dominates *B*. For both *A* and *C*, since no other solutions dominate them, they are non-dominated solutions, i.e., Pareto-optimal solutions. The goal of multi-objective optimization is to find as many Pareto-optimal solutions as possible.

Table 3.1: An example of Pareto-optimal solution

Solution	Object value 1	Object value 2	Object value 3
A	1.5	3	2
B	1.6	4	3
C	0.5	4	4

The particular MOEA used in this work is SPEA2. As shown in [ZIT02], SPEA2 provides good performance in terms of convergence and diversity, and compares well to other representative MOEAs such as PESA [KNO00] and NSGA-II [DEB02] on various well-known test problems.

3.3 System Model

In this section, we introduce the system model used in the proposed SMoC clustering algorithm. The model consists of two parts: stability-aware clustering and the metrics used by each node in the clustering algorithm.

3.3.1 Stability-aware Clustering

An MANET can be dynamically organized into clusters to maintain a relatively stable and effective topology. If clusters exist, the distances between the clusterhead and cluster members should stabilize over a certain period of time. As shown in [LI02], two neighboring mobile nodes in the same mobility group show relative stability of distances. Assume that all nodes have identical and fixed transmission range r. If the distance between two mobile nodes is within r, they can communicate with each other directly. However, this does not necessarily mean that they belong to the same group. Imagine that two mobile nodes briefly fall in the transmission range geographically and separate again, due to different moving directions. Based on these observations, the term Adjacently Grouped Pair (AGP) of nodes is defined, as described in Section 2.3.1. The definition of AGP reveals that if two adjacent nodes are in the same group over a period of time, the distance between them stabilizes around a mean value μ with small variations, where $\mu < r$. Inspired by the AGP definition, we propose the concept of relatively stable neighbors.

Definition 3.2 Node A and B are *relatively stable neighbors* if they form an Adjacently Grouped Pair (AGP).

The relatively stable neighbors of a node can be determined by measuring the distances between it and its neighbors for a fixed number of rounds *l*, where *l* is a pre-determined size of the sampling buffer. By using relatively stable neighbors, the relatively stable topology can be constructed for clustering. The construction method is described in Section 3.4.3.

3.3.2 Node Metrics

In the proposed clustering algorithm, we only consider relatively stable neighbors. If one node is selected as the clusterhead, only its relatively stable neighbors can join this cluster. If node *i* is node *j*'s relatively stable neighbor but has already joined another cluster, node *i* cannot join the cluster served by node *j* again. Hence, when calculating the clustering metrics for node *j*, node *i* should be excluded from node *j*'s available relatively stable neighbors. Each node can decide how well suited it is for being a clusterhead by the following three metrics.

1) Degree Difference

In our algorithm, the degree of a node is only the number of its available relatively stable neighbors. Suppose D_v is the number of relatively stable neighbors of node *v*. Given a threshold δ, which represents the number of neighbors that a clusterhead can ideally handle. Then the degree-difference Δ_v is used as one metric for node *v*.

$$\Delta_v = |D_v - \delta| \tag{3.1}$$

Here we do not impose strict restriction on the cluster size (the number of cluster members). Obviously, the less Δ_v, the more suitable for node *v* to be a clusterhead.

2) Power Consumption

It is known that the power required for supporting a link is inversely proportional to some exponent power of the distance in wireless communications. Since the distance between two neighboring nodes in an MANET is usually rather small (approximately hundreds of meters) comparing to the distance between mobile devices and base stations (the order of 2-3 miles), the power required for supporting a wireless link can be regarded as being proportional to the geometric distance in MANETs [GER95]. Therefore, we use $Dist_v$, the sum of the distances between node *v* and each available relatively stable neighbor, to measure the power consumption required for supporting the communication between node *v* and all its available neighbors.

$$Dist_v = \sum_{v' \in N(v)} \{dist(v, v')\} \tag{3.2}$$

where $N(v)$ is the set of available relatively stable neighbors of node *v*. $dist(v, v')$ is the measured average distance between node *v* and v'.

3) Remaining Battery Lifetime

Assume that each mobile node *v* can estimate its remaining battery energy E_v. Since the power consumption required for node *v* to communicate with its relatively stable neighbors is approximately measured by $Dist_v$, the remaining battery lifetime Rbl_v of node *v* can be represented as:

$$Rbl_v = E_v / Dist_v \tag{3.3}$$

It is expected that the nodes with longer remaining battery lifetime are selected as clusterheads.

3.4 Description of the SMoC Algorithm

3.4.1 Problem Encoding

Chromosome is the basic element in an evolutionary algorithm. A certain number of chromosomes form a population. The encoding of a chromosome is important. First, each chromosome should represent a feasible solution, which is randomly distributed in the solution space. Second, a good encoding method benefits the realization of genetic operations.

Each solution produced by our algorithm stands for a set of clusterheads, which are selected from all the nodes in the network. Hence, a random permutation of node IDs results in a random set of clusterheads. In this algorithm, we use random permutation of node IDs to represent a chromosome. It is important to guarantee that there is no duplicate node ID in each chromosome. Each node ID in the chromosome is called a gene. For example, in an MANET consisting of eight nodes with IDs ranging from 1 to 8, a random permutation (4 3 8 7 1 6 2 5) represents a chromosome.

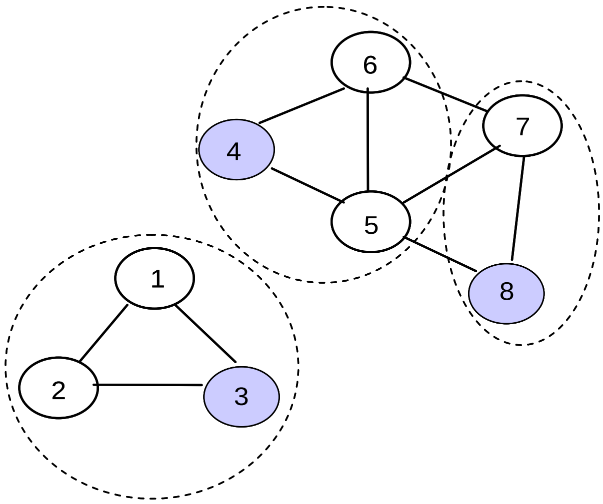


Fig. 3.1: A relatively stable topology derived from a network of 8 nodes

We need to derive a set of clusterheads from each chromosome. Let us explain this method with an example. Assume the chromosome is (4 3 8 7 1 6 2 5). Figure 3.1 shows the relatively stable topology constructed from the network. First, we add the first gene 4 into the clusterhead set. Then all the relatively stable neighbors of node 4 are no longer allowed to be clusterheads. From Figure 3.1, we know the relatively stable neighbors of node 4 are nodes 5 and 6. We continue to check the next gene and add node 3 into the clusterhead set. The relatively stable neighbors of node 3 are nodes 1 and 2. So nodes 1, 2, 5, 6 are not considered as clusterheads anymore. Then we add node 8 into the clusterhead set. The available relatively stable neighbors of node 8 are node 7. Hence node 7 is also forbidden as the clusterhead. Until now, all the nodes have been checked and a clusterhead set {4, 3, 8} is generated. Table 3.2 illustrates the above procedure of clusterhead selection and Figure 3.1 illustrates the clustering results.

Table 3.2: Procedure of deriving a clusterhead set from a chromosome

Step	Candidate genes for clusterheads	Clusterhead set
1	(4 3 8 7 1 6 2 5)	{}
2	(- 3 8 7 1 - 2 -)	{4}
3	(- - 8 7 - - - -)	{4,3}
4	(- - - - - - - -)	{4,3,8}

Since we consider an MANET with group mobility, without loss of generality, we assume all the nodes form groups and the average group size is $\bar{g}$. Hence, the number of groups is $n/\bar{g}$. Due to the fact that a clusterhead and all its relatively stable neighbors belong to the same group, there are no inter-group clusters. Within a group, each group member needs to send the message about its relatively stable neighbors information to the group leader. Normally, the group leader stays at the group center. We assume the group diameter is *d*. Thus, the average number of hops that a message travels is $(1+d/2)/2$. Then the total number of clustering messages in a group is given by $(\bar{g}-1)*[(1+d/2)/2]$. Since there is no need to send inter-group clustering messages, the total number of clustering messages in the network is $(n/\bar{g})*(\bar{g}-1)*[(1+d/2)/2]$. As a result, the time complexity of the technique used to derive a clusterhead set is $O(n*d)$.

3.4.2 Optimization Objectives

In Section 3.3.2, we define three metrics to measure the suitability of a node as the clusterhead. Since each node can calculate these metrics based on its local information, we assume that every node is aware of the current values of its metrics. In our problem, we should evaluate each clusterhead set instead of each single clusterhead. Hence, we need to give an overall evaluation on the clusterhead set in terms of each metric. Since both degree difference and power consumption are additive metrics, it is natural to use the sum of the metric value of each clusterhead as the overall optimization objective (i.e., clustering metric). The sum of degree difference of each clusterhead reflects the overall deviation of the node degrees from the ideal case. The sum of power consumed by each clusterhead reflects the total power consumed by all the clusterheads. However, the metric for the remaining battery lifetime is a concave function. Hence, the reasonable evaluation object is the minimum remaining battery lifetime among all the clusterheads because it determines the maximum lifetime of the whole clusterhead set.

Assume $s\_CH = \{c_1, c_2, \cdots, c_m\}$ is a clusterhead set. We formally define the three optimization objectives (i.e., clustering metrics) mentioned above as follows.

1) The total degree differences of *s_CH*, Δ_{s_CH}:

$$\Delta_{s\_CH} = \sum_{c_i \in s\_CH} \Delta_{c_i} \tag{3.4}$$

2) The total power consumption of *s_CH*, D_{s_CH}:

$$D_{s\_CH} = \sum_{c_i \in s\_CH} D_{c_i} \tag{3.5}$$

3) The cluster lifetime of *s_CH*, Rbl_{s_CH}:

$$Rbl_{s\_CH} = Min\{Rbl_{c_i} \mid c_i \in s\_CH\} \tag{3.6}$$

For both Δ_{s_CH} and D_{s_CH}, the less the value, the better the clusterhead set. This is due to the fact that we expect each clusterhead to serve just δ cluster members and consume as little power as possible for intra-cluster communication. However, for Rbl_{s_CH}, we expect its value as large as possible. Hence, our objective is to minimize both Δ_{s_CH} and D_{s_CH}, and maximize Rbl_{s_CH}.

3.4.3 Formal Description of SMoC

We first construct a relatively stable network topology for an MANET by the following method.

Step 1 For each node *v*, find out all its relatively stable neighbors *N*(*v*).
Step 2 For each node $w \in N(v)$, if there is no link between *v* and *w*, add a bidirectional link to connect them.
Step 3 Repeat Step 1-2 until all the mobile nodes have been examined.

The relatively stable topology can be regarded as a "quasi-static" network topology over a certain period of time. The following multi-metric clustering procedure just runs on this relatively stable topology. Just like the distributed clustering algorithm [BAS99], we assume that the network topology does not change during the execution of the clustering algorithm.

In the following, we present the formal description of the proposed clustering algorithm as shown in Figure 3.2. In the beginning, the algorithm constructs the relatively stable topology, which is used to discover the relatively stable neighbors by each node. Line 3 creates the initial population P_0 and the empty Pareto set Q_0. Thus, P_0 consists of a certain number of chromosomes, which are represented by random permutation of node IDs, whereas Q_0 is the final output of this algorithm and initialized to be empty. Both *P* and *Q* have constant size over time in the algorithm. In Line 4, *T* denotes the maximum number of evolutionary generations and *t* denotes the current generation number that the population has evolved to. The algorithm stops when *T* is reached.

From each chromosome $i \in P_t \cup Q_t$, we first derive the corresponding clusterhead set, *s_CH*. Then each clusterhead in this set calculates its three node metrics Δ_v, $Dist_v$, and Rbl_v according to Equations 3.1-3.3 respectively. After all these values are obtained, the algorithm calculates the three optimization objectives (i.e., clustering metrics) Δ_{s_CH}, D_{s_CH}, and Rbl_{s_CH} following Equations 3.4-3.6. Thus, for each chromosome $i \in P_t \cup Q_t$, the values of its three optimization objectives are determined. Based on these values, all the non-dominated (i.e., Pareto-optimal) chromosomes in $P_t \cup Q_t$ are determined.

In Line 12, all the non-dominated chromosomes in $P_t \cup Q_t$ are copied to Q_{t+1}, the Pareto set at the (t+1)th evolutionary generation. It is possible that the number of non-dominated chromosomes in $P_t \cup Q_t$ is not equal to the specified size of Q_{t+1}. To solve this problem, the SPEA2 algorithm adopts the so called environmental selection method. If the number of non-dominated chromosomes exceeds the size of the Pareto set, an archive truncation procedure is invoked, which iteratively removes chromosomes from Q_{t+1} until its size satisfies the requirement. The chromosome, which has the minimum distance to another chromosome, is removed at each iteration. If the non-dominated chromosomes cannot fulfil Q_{t+1}, the best dominated individuals in $P_t \cup Q_t$ will be added into Q_{t+1}. The algorithm then checks if the maximum generation number is reached. If so, it stops. Otherwise, the algorithm enters the SPEA2 mating selection phase, where chromosomes from Q_{t+1} are selected by means of binary tournaments to generate the mating pool.

begin
construct relatively stable topology
create the initial population P_0 and the empty Pareto set Q_0
let T be the maximum generation number and set counter $t = 0$
while $t < T$ do
  for each chromosome $i \in P_t \cup Q_t$ do
    derive the clusterhead set *s_CH* from i
    for each node $v \in s\_CH$
      node v calculates its own metric values: Δ_v, $Dist_v$, and Rbl_v
    end of for each node $v \in s\_CH$ loop
    calculate the three optimization objectives for *s_CH*: Δ_{s_CH}, D_{s_CH}, and Rbl_{s_CH}
  end of for each chromosome $i \in P_t \cup Q_t$ loop
  copy all non-dominated chromosomes in both P_t and Q_t to Q_{t+1}
  perform SPEA2 environmental selection on Q_{t+1}
  if ($t >= T$) then break
  perform SPEA2 mating selection on Q_{t+1} to generate the mating pool
  apply crossover and mutation operators to the mating pool and then set the resulting population to be P_{t+1}
  $t = t + 1$
end of while loop
return Q_t
end of the algorithm

Fig. 3.2: The formal description of SMoC

Once the mating pool is formed, the algorithm applies crossover and mutation operators to the chromosomes in it. Crossover and mutation are two important genetic operators. Crossover helps generate two offspring chromosomes from two parent chromosomes. All the genes in each offspring chromosome are inherited from different parts of the two parent chromosomes. In this algorithm, we employ the well-known X-Order1 method [TUR02]. Mutation generates an offspring chromosome from only one parent chromosome by changing some genes' values. We employ the simple and effective gene swapping method for mutation. Finally, the Pareto set Q_t is output as the set of solutions, each of which corresponds to a clusterhead set.

Since the initial population consists of chromosomes, which are randomly generated, there may be some duplicate chromosomes in it. In addition, the crossover and mutation operators applied to the mating pool may also produce some duplicate chromosomes in the resulting population. Therefore, there may have some duplicate chromosomes in the final Pareto set.

3.4.4 Cluster Reconfiguration

In this chapter, we consider an MANET with group mobility, in which most of the nodes form groups. The relatively stable topology is discovered based on the property that we exploit from group mobility, i.e., the relative stability of distances between two neighboring nodes belonging to the same group. Since an MANET is a dynamic system, we assume dynamic group membership. Hence, it is allowed that a new node joins a group or a group node leaves its group.

The dynamic group membership leads to the cluster reorganization. When a new node v joins a group, it first finds out all its relatively stable neighbors, $N(v)$, by the method mentioned in Section 3.3.1. A clusterhead periodically broadcasts a HELLO packet indicating its role and its cluster size. If there is no existing clusterhead in $N(v)$, node v claims itself as a clusterhead. Otherwise, among all the clusterheads in $N(v)$, node v joins the one with the smallest cluster size.

A group node intending to leave its group may be an ordinary group member or a clusterhead. When an ordinary group member leaves, its clusterhead detects its departure and deletes it from the list of relatively stable neighbors. The clusterhead then reduces its cluster size by one. When a group leader leaves, each of its clustermembers searches its own list of relatively stable neighbors. Similar to a new node, each node *v* claims itself to be a clusterhead or joins the clusterhead with the smallest cluster size in *N*(*v*).

However, since mobile nodes form groups purposely, the group membership change rarely occurs. Hence, the effect of dynamic group membership to cluster stability is trivial.

3.5 Performance Evaluation

We conduct simulation study to demonstrate the effectiveness of the proposed stability-aware multi-metric clustering algorithm. The standard SPEA2 source codes [LAU01] are adopted for function modules of environmental selection and mating selection; thus the correctness of multi-objective evolutionary process is guaranteed.

Table 3.3: The parameter values

Parameter	Value
population size	10
Pareto set size	20
crossover rate	0.8
mutation rate	0.1
maximum number of evolutionary generations	50

The main parameters used in the algorithm are population size, Pareto set size, crossover rate, mutation rate and maximum evolutionary generation number. In our simulation experiments, the parameters are set as shown in Table 3.3. Both the initial population size and the Pareto set size are set to be two times the normal population size. We run extensive simulations by adjusting different combinations of parameter values to achieve the best one. The simulation area is a square of 1km X 1km. The network size varies from 20 to 200 in different simulation scenarios. Each mobile node has the same radio transmission range of radius r = 100m.

3.5.1 Stability Evaluation

Cluster stability is an important performance metric in our algorithm. We evaluate it in an MANET of 200 nodes. Most of the mobile nodes form groups with various sizes. We allow 1% of the nodes to be single nodes and move freely. The group nodes follow the improved Reference Point Group Mobility model [LI02], in which the nodes in the same group share the common group motion vector. We assume that the group velocity conforms to the random waypoint model [BET03] and the maximum speed varies from 5m/s to 25m/s in different simulation scenarios. The single nodes also move with a random speed, which is less than the maximum speed in each simulation.

To evaluate the stability performance of the clustering algorithm, we define a new metric – the number of remaining stable clusters. It counts the number of clusters whose structures have not been changed after a time period T since the clustered topology is created. T is determined as follows.

$$T = k * \frac{r}{0.5 * MaxSpeed} \tag{3.7}$$

where r is the radio transmission range, *MaxSpeed* is the specified maximum speed during each simulation, k is a constant.

In each simulation scenario with different *MaxSpeed*, the network is clustered using the proposed algorithm with and without considering relatively stable topology. After time *T*, we count the number of remaining stable clusters in these two cases. Each simulation is run three times with the averages recorded. The comparison results are shown in Figure 3.3.

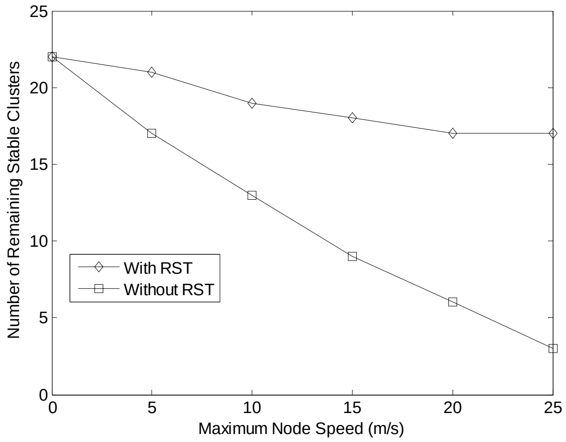


Fig. 3.3: Comparison of the number of remaining stable clusters for clustering with and without RST (relatively stable topology) constructed

From Figure 3.3, when the network is a static ad hoc network, i.e., the *MaxSpeed* is 0, the stability performance is the same for both cases due to no mobility. However, when the nodes move, with the relatively stable topology constructed, the stability performance achieved by the clustering algorithm is better than the case without the construction of relatively stable topology. When the network becomes more and more dynamic, i.e., the node movement speed increases, the advantage of exploiting group mobility is more distinguished.

3.5.2 Multi-objective Optimization Evaluation

It has been proved that finding an optimal set of clusterheads with one or more clustering metrics is NP-hard [CHA02]. For small-scale network topology, since the optimal solution can be found by exhaustive search, we compare the solutions achieved by our algorithm with the optimal solutions. However, for large-scale network topology, the exhaustive search for the optimal solution becomes infeasible due to exponential time complexity. Hence, we compare the proposed algorithm with WCA [CHA02] and its two improvements by GA [TUR02] and SA [TUR03].

1) Evaluation on Small-scale Network Topology

An MANET consisting of 20 nodes is used as the small-scale network in our experimental study. We run the proposed multi-metric clustering algorithm on its relatively stable topology. Then count the number of Pareto-optimal solutions regarding each clustering metric obtained at various evolutionary generation numbers. Figure 3.4 shows the results.

Since the Pareto set size, i.e., the number of solutions, is 20, we finally get 20 clusterhead sets, some of which may be duplicate. Figure 3.4 shows that after only 2 generations of evolution, the algorithm can achieve the optimal clusterhead sets regarding each clustering metric on the small-scale network. For the total degree differences, the ratio of the number of optimal solutions to the Pareto set size is above 50% when the number of evolutionary generations exceeds 5. For power consumption, the number of optimal solutions fluctuates

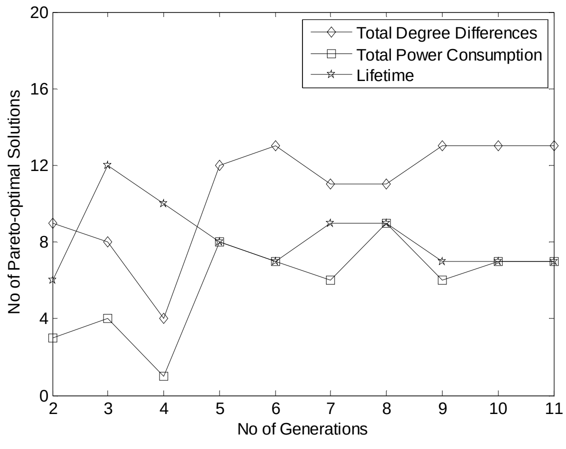


Fig. 3.4: The number of Pareto-optimal clusterhead sets regarding different clustering metrics

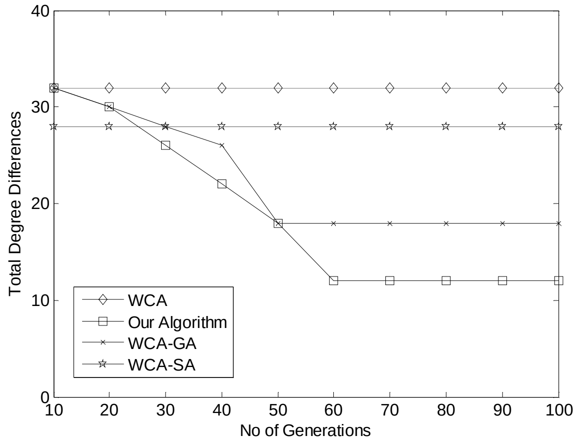
when the generation number is less than 8. But when it exceeds 8, the ratio of the number of optimal solutions to the Pareto set size stabilizes around 35%. For the remaining battery lifetime, the ratio also stabilizes around 35% when the number of generations exceeds 4.

2) Comparison with WCA and Its Two Improvements

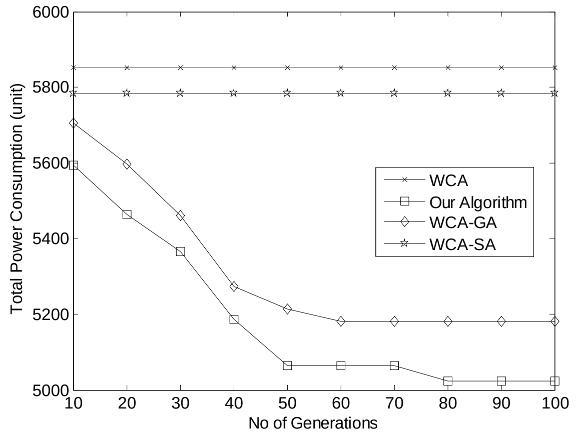
WCA uses the combined weight of four clustering metrics as the single optimization objective. The WCA algorithm has been further optimized by genetic algorithm [TUR02] and simulated annealing [TUR03]. Here these two improvement algorithms are named as WCA_GA and WCA_SA, respectively. We implement WCA, WCA_GA and WCA_SA, and compare our algorithm with them to evaluate its performance in terms of the solution quality in an MANET of 200 nodes.

In the following, we simply describe the basic ideas of WCA, WCA_GA, and WCA_SA. First, WCA marks the node with the best weight as a clusterhead and all its neighbors as the cluster members. Then WCA deletes both the clusterhead and all its neighbors from the network topology. The weights of the remaining nodes are recalculated based on the remaining network topology. The above process is repeated until no node is left in the network topology.

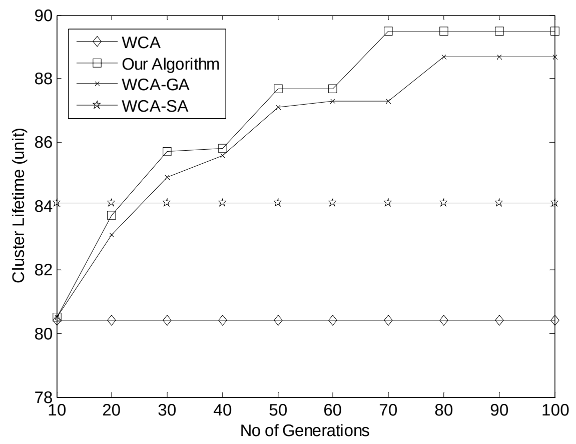
In WCA_GA, there is also a population of chromosomes, each one of which represents a random clusterhead set. After evolving a certain number of generations, the algorithm stops since it meets one of the following termination requirements: the maximum generation number is reached or the population converges. The Elitist model is employed in WCA_GA to record the current best solution among the population. In WCA_SA, there is only one initial solution instead of a population. At each iteration, the algorithm randomly searches a solution neighboring to the present one in the solution space. If the



(a)



(b)



(c)

Fig. 3.5: The comparison of: (a) the total degree differences, (b) the total power consumption, (c) the cluster lifetime

neighbor solution is better than the current one, WCA_SA will replace the present solution by the neighbor solution. Otherwise, the algorithm will accept the neighbor solution with a probability.

Since WCA, WCA_GA, and WCA_SA are actually heuristic clustering algorithms considering single metric, we use them to find a clusterhead set for each optimization objective. Then regarding the same clustering metric, the best solution in the final Pareto set of our algorithm is compared with the clusterhead sets obtained by WCA, WCA_GA and WCA_SA, respectively. Figure 3.5 shows the comparison results for various evolutionary generation numbers. In both WCA and WCA_SA, there is no concept of evolution. Hence, the performance of these two algorithms is not related to the number of evolutionary generations. For comparison purpose, we still show the same value of one metric with respect to different generation number.

From Figure 3.5, we observe that the proposed clustering algorithm outperforms WCA, WCA_GA and WCA_SA in terms of all the clustering metrics. When the evolutionary generation number exceeds 50, our algorithm stabilizes around some Pareto-optimal solutions. More importantly, our algorithm finds these good solutions in one Pareto set. However, WCA, WCA_GA and WCA_SA cannot.

3.6 Summary

The selection of the optimal clusterhead set is proved to be a NP-hard problem. Hence, even for single clustering metric, we cannot find the best solution using an algorithm with polynomial time complexity. For multi-metric clustering, the problem becomes more difficult to solve. Only heuristics can be developed for multi-metric clustering in MANETs.

In this chapter, we first exploit the relatively stable topology resulted by group mobility to improve the stability of the cluster structure. We then define three clustering metrics. Based on the relatively stable topology and the three clustering metrics, a stability-aware multi-metric clustering algorithm for MANETs is proposed. The algorithm can achieve a population of solutions, which are the Pareto-optimal clusterhead sets with respect to the three clustering metrics. Moreover, it can generate the Pareto-optimal solution that does not provide best possible value for any individual metric, yet it offers Pareto-optimal solution when considering all three metrics together. Such a solution is often useful for applications with a fair compromise between multiple optimization objectives.

Performance evaluations are conducted on both stability and multi-metric optimization. Simulation results show that our algorithm has good stability performance and achieves better clusterhead sets than a well-known clustering algorithm WCA and its two improvements WCA_GA and WCA_SA.

Chapter 4

GrLS: A Group-based Location Service Protocol in MANETs with Group Mobility

In this chapter, we introduce the proposed *Gr*oup-based *L*ocation *S*ervice protocol, GrLS. This chapter is arranged as follows. Firstly, a brief overview is given in Section 4.1. Section 4.2 describes the system model. Section 4.3 presents the design of location management schemes in GrLS. Section 4.4 describes the strategies used to handle node mobility. Theoretical analysis and performance evaluation on GrLS are discussed in Section 4.5 and Section 4.6, respectively. Finally, Section 4.7 summarizes this chapter.

4.1 Overview

In the rendezvous-based location service protocols, all the nodes in the network need to keep a publicly known mapping, which maps each node's unique ID to its location servers. Each mobile node recruits at least one other node as its location server and, whenever necessary, sends location updates to the location servers to update its location. Once a source node wants to know the location of the destination, it will send a location query to the location servers of the destination. At least one of the location servers should receive the location query and send the location reply to the source. Clearly, a rendezvous-based distributed location service protocol needs to address the following issues:

(1) how to recruit location servers;

(2) when to send location update;

(3) how to determine the location servers of a node given its node ID.

Node mobility is one of the intrinsic characteristics in MANETs. By single mobility, a node moves according to its own mobility pattern. As we have discussed before, in recent years, group mobility, where mobile nodes are organized into groups to coordinate their movement, has emerged from the demand of applications where a team of mobile users aggregate and work together. In the group, all the group members stay closely and move according to the same mobility pattern. Since a group of nodes always move as a whole and have the similar location tracks, group mobility can be further exploited to improve the efficiency of location management. This motivates our research.

To the best of our knowledge, GrLS is the first location service protocol that has exploited group mobility. GrLS consists of two major components: single location management and group location management. In single location management, each node with single mobility sends location updates to the location servers in its home region, which also handle all the location queries for it. For nodes with group mobility, group location management applies, which consists of micro and macro group location management. With micro group location management, each group member is aware of the locations of all other group members. Thus, intra-group communications can be conducted immediately. With macro group location management, a designated group leader updates its location to the location servers in the group home region and replies all the location queries for the group members. Thus, the overhead of location updates to the home regions can be saved for all the group members except the group leader.

The major contributions of this chapter are as follows:

- *A novel network partition method is proposed to allocate the home regions for mobile nodes. The partition originates from the network center and spreads outward. Thus, all the home regions symmetrically spread around the network* *center. On an average basis, a home region can potentially be close to both the source and destination nodes.*

- *A novel strategy of recruiting location servers is proposed for both single nodes and group leaders. The strategy allows the load of maintaining the location service to be spread evenly across all the nodes in the network. Moreover, when a location server moves, only one message is needed for location information handoff.*
- *An effective and efficient group location management strategy is proposed. By micro and macro group location management, both communication locality-awareness and low protocol overhead can be achieved. To manage the group membership information, ID servers are recruited. Correspondingly, ID update, query, and reply are designed. When nodes change their roles upon joining or leaving a group, a seamless handoff between single location management and group location management is supported.*
- *Other desirable features of GrLS include: (1) an adaptive location update scheme, which can achieve a reasonable tradeoff between location accuracy and the protocol overhead; (2) an optimal strategy of forwarding location update in the home region without using broadcasting or flooding, which shows the best spatial and temporal performance; (3) effective methods to handle empty regions.*

4.2 System Model

Without loss of generality, we assume that all the mobile nodes are aware of their own locations and have the same radio transmission range *r*. Periodic *HELLO* messages are used to exchange node IDs and location information between neighboring nodes. There are two types of nodes in the network: single nodes and group nodes. A single node moves according to its own mobility pattern. A group node joins a group and moves according to the group mobility pattern. In a group, one node is designated as the leader and all other nodes are group members.

The role of each node can change as time goes. A single node can become a group node by joining a group, and on the other hand a group node can become a single node by leaving a group. A group leader can become a group member or leave the group, requiring a new group leader to be designated and the handover of leadership performed.

In GrLS, the coverage area of an MANET is partitioned for allocation of home regions to mobile nodes. A network center based partition method is proposed to achieve the effect that a home region can potentially be close to both the source and destination nodes. The center of the network coverage area is roughly estimated at the time the MANET is initialized.

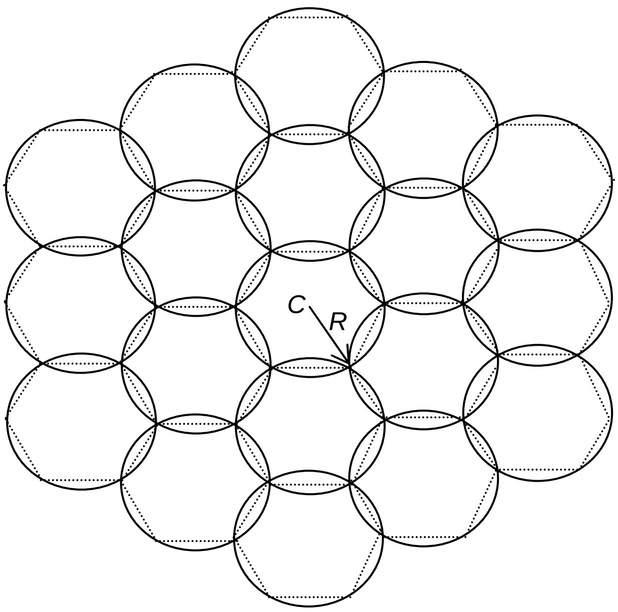


Fig. 4.1: Illustration of the network coverage area partition

The partition originates from the network center and spreads outward. As shown in Figure 4.1, the area is partitioned into equal circle-shaped regions. As the dotted lines show, each circle contains a hexagon with the side length equal to the radius of the circle. These hexagons are non-overlapping but can completely cover the entire network. Each circle has six neighboring circles since a hexagon has six sides. We denote the radius of the circle as *R*, $R = \sqrt{7}/2\,r$. Thus, a central region stays at the network center and other regions spread around it symmetrically. Each region is assigned a unique region ID.

At startup, all nodes know the network center and the partition method. Thus, based on its location, a node can calculate the region where it is staying. We assume that there exists a publicly known hash function that maps a node's ID to a specific region (called its home region),

$$F(\text{Node ID}) \rightarrow \text{Region ID},$$

F is a many-to-one mapping.

The central region is selected as the group home region by all the group leaders. Each group leader recruits both location servers and ID servers in the group home region. All the other regions are selected as home regions by single nodes, which recruit location servers there. All the home regions spread around the network center, which can alleviate the drawback of flat hashing-based protocols, i.e., location servers in a home region can potentially be far away from both the source and destination nodes. A circle-shaped home region can further benefit the location management, as shown later in Section 4.3.1.

4.3 Location Management

4.3.1 Recruiting Location Servers

A mobile node needs to determine which nodes in the home region should be recruited as its location servers. Generally, there are three options: 1) one node, 2) all the nodes, and 3) part of the nodes. The first option has a number of problems, e.g., the centralized server is a single point of failure. The second option produces the heaviest protocol load since all the nodes in the home region are involved into the location service for all the nodes which have been hashed to this home region. The third option seems to be the best and GrLS also adopts it.

In [WU05a], a node functions as location server at a probability. Clearly, it leads to uncertainty and incur high searching overhead. We propose a strategy to evenly distribute the load of location services across all the nodes in the home region. As shown in Figure 4.2, we further partition a circle-shaped home region into seven circle-shaped subregions with subregion ID ranging from 0 to 6. A node recruits one location server in each subregion of its home region. Each subregion is a small circle with a radius of ½ *r*. Clearly, a node knows all its neighbors in the same subregion through beaconing. A node can directly send messages to all other nodes in the same subregion. This is just the reason why we partition the network into circle-shaped regions with radius of $\sqrt{7}/2\,r$.

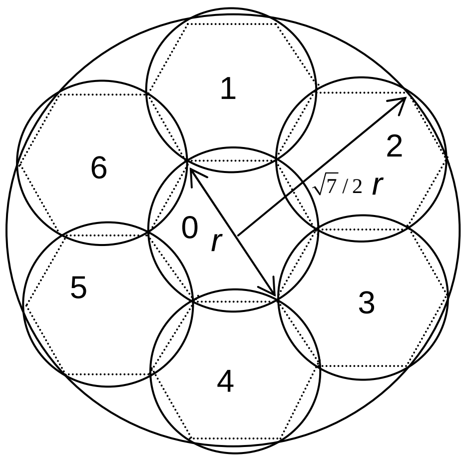


Fig. 4.2: The division of a home region

As above mentioned, the location servers of a node *A* are evenly distributed in its home region. To further balance the load among the nodes in the same subregion, node *A* recruits the node whose ID is the closest to its own ID. We define the ID "close" relationship as follows. The node ID space is assumed to be circular in clockwise direction from small IDs to large IDs. In the space, to its "closest" neighbor, a node has the least ID distance, which is measured in clockwise direction from the node's ID to the neighbor's ID. For example, there are 60 nodes with ID ranging from 1 to 60. Now node 20 wants to recruit a location server in a subregion with nodes 16, 25 and 40. According to the rule, node 25 is recruited. Hence different nodes recruit different location servers in the same subregion. Overall, the responsibilities of acting as location servers are evenly shared among all the nodes in a subregion.

Through the above analysis, the proposed location server recruitment strategy has two desired properties: (1) each node selects the same number of location servers that are evenly distributed in its home region, and (2) each location server in the home region also serves approximately the same number of nodes. As a result, our strategy of recruiting location servers is scalable and load balanced.

4.3.2 Basic Location Management

1) Adaptive Location Update

Generally, similar as cellular network, there are two kinds of schemes to trigger location update in MANETs: time-based and distance-based. In the time-based scheme, a mobile node updates its location periodically, e.g., every tens of seconds. In the distance-based scheme, a mobile node tracks the distance it has moved since last update and triggers the location update when the distance reaches a predefined threshold.

GrLS adopts an adaptive location update scheme combining the advantages of both time-based and distance-based schemes. Initially, we set the minimum and maximum location update intervals and define the distance threshold of location update. If the distance the node has moved since last update reaches the threshold value, but the time has not exceeded the minimum location update interval, the node will not send any location update; if the time is between the minimum and the maximum interval, location update will be sent. On the other hand, if the maximum location update interval is reached, but the distance the node has moved is less than the threshold value, the node will trigger the location update immediately. For the distance threshold of location update, according to [WU05b], it can be approximately half of the radio transmission range of mobile nodes.

The adoption of a minimum interval can help reduce the frequency of location update when nodes are moving with high speeds. In highly mobile networks, if no restriction is put on minimum interval, many location update messages will be generated, probably leading to network congestion. The maximum interval aims to guarantee a certain frequency of location update when nodes are moving slowly or staying stationary. Because many location servers set expiry timer for the location information they have stored, if a node has not updated its location for a long time, the location server will remove it from the database. Hence, when a node moves with low speed or stays stationary, it should still periodically send location update to its location servers upon the expiration of the maximum location update interval.

Now, we describe the basic location update mechanism in GrLS. Both single nodes and group leaders send the location update messages toward their home regions using geographic forwarding where the center of each home region is the destination. Once a node residing in one subregion of the home region receives the message, this node becomes a proxy for this subregion. The proxy knows all its neighbors in the same subregion through beaconing. According to the strategy of recruiting location servers, given the source ID in the location update message, the proxy can easily determine which node is the desired location server in its subregion. The proxy then forwards the message to this location server. It is possible that the proxy itself is the desired location server.

Upon receipt of the location update message, the location server updates the corresponding location information and appends its subregion ID to the location update message, and continuously forwards it toward the center of the central subregion, which is also the center of the home region. Since the location update message may traverse some other subregions before it arrives at the central subregion, the same process of message forwarding by the proxy and location update to the desired location server are performed in each visited subregion. The subregion ID of each visited subregion is also appended to the message. When the location update message finally reaches the central subregion, a proxy forwards it to the desired location server. Then the location server in the central subregion unicasts the location update message to the remaining unvisited subregions separately, where their centers are the destinations. Finally, the location servers are updated in these unvisited subregions, and then the location update of the node to its home region is completed. Figure 4.3 shows how a location update message arrives at all the desired location servers once it reaches the home region, which is named as center-based forwarding.

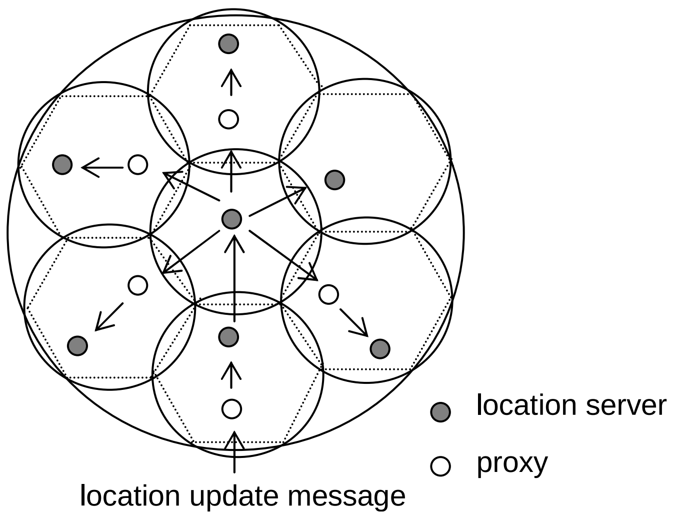


Fig. 4.3: Center-based forwarding of location update message

In a home region, our strategy of forwarding location update message incurs low overhead because neither broadcasting nor flooding is used. Moreover, it has good spatial and temporal performance. To show this, we compare it with two other possible forwarding strategies, which are illustrated in Figure 4.4. In Figure 4.4(a), the first location server unicasts the location update message to other six subregions separately. We name it as parallel forwarding. In Figure 4.4(b), the first location server unicasts the location update message to its neighboring subregion in clockwise direction. The neighboring subregion continuously forwards the message to its own neighboring subregion also in clockwise direction. The location server in each visited subregion is required to append its subregion ID to the message. This forwarding procedure is repeated until one location server finds that the message has already visited its neighboring subregion in clockwise direction. Then the location server sends the location update message to the central subregion. Until now, the location update procedure is completed. We name it as sequential forwarding.

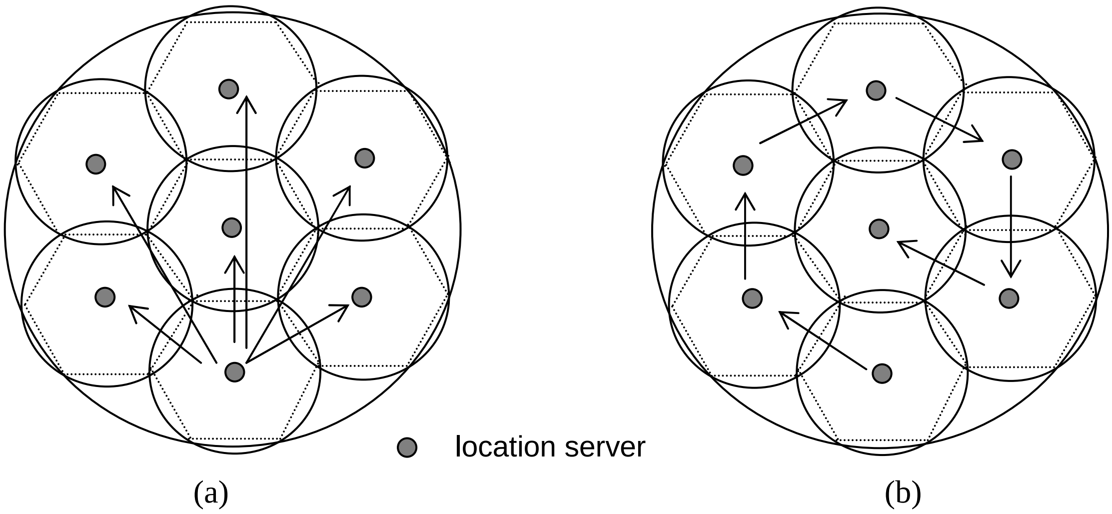


Fig. 4.4: Another two possible strategies of forwarding location update in a home region:
(a) parallel forwarding, (b) sequential forwarding

Without loss of generality, we assume a simplified model. In the model, each location server just lies at the center of the subregion. Thus the forwarding between two neighboring centers is just one-hop transmission, which also brings two-hop reachability between two non-neighbor centers. We count the total number of hops traversed by the location update message in each forwarding strategy. In addition, we assume the average time for one-hop transmission is *t*. Then we get the approximate time spent on the location update procedure for each strategy. The counting begins from the time that the first location server receives the message. The comparison is shown in Table 4.1.

Table 4.1: Comparison results of three strategies of forwarding location update message

Strategy of forwarding location update message in the home region	Hops traversed by the location update message	Total time spent on the location update procedure
Parallel forwarding	9	2*t*
Sequential forwarding	6	6*t*
Center-based forwarding	6	2*t*

Table 4.1 shows that our forwarding strategy is the best one, which traverses the minimum number of hops and spends the least time. These two values are also the optimal for any possible forwarding strategies, which can be easily proved. Clearly, our forwarding strategy has good spatial and temporal performance.

2) Location Query

If a source node *s* wants to query the location of a destination *d*, it will first query its own location database. If *d*'s location can be found, there is no need to trigger a location query. Otherwise, *s* sends a location query message to *d*'s home region. Since *s* knows the hash function, *d*'s ID and the network center, *s* can easily calculate the location of the center of *d*'s home region, which is just the destination of the location query message. The location query message also carries *s*'s location, which is useful when a location server sends a location reply to *s*. Since *d* may be a single node or a group node (group member or leader), different location query strategies are proposed. Here, we describe the location query for single nodes. The strategy for querying group nodes will be described in Section 4.3.3.

For a single node *d*, the message for querying its location is firstly received by a node in one subregion of *d*'s home region. Then the node acts as a proxy in this subregion. By *d*'s ID, which is carried in the location query message, the proxy can easily determine which node is the desired location server of *d*. If the desired location server is just the proxy, reply can be sent back to *s* immediately. Otherwise, the proxy directly forwards the location query message to *d*'s location server in this subregion since the location server is one neighbor of the proxy. Upon receiving the location query, the location server sends a location reply message to the source *s* through geographic forwarding.

4.3.3 Group Location Management

As pointed out in [AN01], in practical MANET environments, random mobility and group mobility occur simultaneously. By far, group mobility has not been addressed in previous location service protocols. In GrLS, we propose specific group location management for nodes which have formed groups. The group location management consists of two parts: micro group location management which helps each node acquire the locations of all other nodes in the same group, and macro group location management in which only group leader updates its location to location servers and answers the location query for any node in the group.

Each group can be regarded as a logical subnet. Initially, a group leader is selected. The group leader can be the node that is most stable and stays at the approximate center of the group. Here, "the most stable" means that the group leader has the most approximate velocity to group velocity. A group leader like this can guarantee that each group member has an average minimum distance to the group leader. However, the detailed method of group leader selection is out of the scope of this research work.

1) Micro Group Location Management

a) Group Initialization

Once a group leader is determined, it broadcasts its ID and location information to all the group members. Then each group member is aware of the group leader. Upon receiving the announcement of the group leader, each group member makes a reply by sending its own ID and location information to the group leader. When the group leader has collected the information of all the group members, it generates a GroupView message containing both ID and location information of all the group members. The GroupView message is then broadcast to all the group members. Here, a location-guided multicast tree [CHE02] from the group leader to all the group members can also be constructed to transmit messages. Once a group member receives the GroupView message, it can maintain a consistent view about the group and know the location of any other group member. Then the group initialization is completed.

b) Group Maintenance

We define a new concept- group relative location. In addition to the actual location, each group member also has a group relative location, which is the relative location of its actual location to the actual location of the group leader. Clearly, the group relative location of the group leader itself is (0, 0). Each group member periodically calculates its group relative location. Once the distance change of its group relative location has reached a predefined threshold, the group member will send a location update to the group leader. In addition, when the maximum location update interval is reached, the group member also needs to send a location update to the group leader immediately.

If a group member has not updated its location for a predefined time period (i.e., the location expires), the group leader will think it has left the group and then remove its ID and location information from the database. When the group leader finds that the number of group members, whose locations have changed or expired, has reached a certain percentage of the group size, it broadcasts a GroupViewChange message to all the group members to refresh the group view. The group leader also broadcasts its own location update to all the group members, but based on the distance change of its actual location.

Since group initialization and group maintenance are necessary components in most of group management protocols, we piggyback the location information into the group management messages to realize micro group location management. Thus the extra overhead caused by micro group location management is trivial. Furthermore, the communications within the same group are locality-aware since each group member directly knows the locations of all other group members.

2) Macro Group Location Management

a) Group Home Region

In Section 4.3.1, we have introduced how to recruit location servers for single nodes. GrLS does not provide home regions for group nodes except group leaders. All the group leaders share the same group home region, i.e., the central region at the network center. Similar to other home regions, the group home region is also divided into seven subregions with subregion ID ranging from 0 to 6. Each group leader recruits one location server, which has the closest ID to its own ID, in each subregion of the group home region.

As we have mentioned, one drawback of flat geographic hashing protocols is that a home region can potentially be far away from both the source and destination nodes, causing location update and query with high overhead. To alleviate this problem, we let all the group leaders recruit location servers in the central region. The number of group leaders is exactly the same as the number of groups, which is intuitively small. Thus the nodes within the central region will not be overloaded. Even when we want to further reduce the load in the group home region, we can scale it to the central region plus its six neighboring regions.

b) Reactive ID Update

In each subregion of the group home region, the node with the least ID is recruited as ID server by all the group leaders. Totally, there are seven ID servers in the group home region. ID server is used for group membership management. It stores the group membership information of each group, i.e., the IDs of both group leader and all the group members.

ID update, a new type of update message, is created to update the group membership information stored in the ID server. An ID update message is generated on-demand by the group leader when a new node joins the group or a group member leaves the group. Since most groups are formed by nodes purposely, group membership does not change drastically. Hence, ID update is triggered much less than location update. The overhead incurred by ID updates is also much lower than location updates.

c) Location Service Handoff

(1) When a node joins a group, it will notify its home region to disable the location service for it. Then it sends its ID and location to the group leader.

The node sends a location update to nullify its location information stored in its location servers, but the node ID is still kept in the location servers to indicate that the node has joined a group. It is different from the case that all the information of a node is removed from the location servers due to expiry. Once receiving the message from the new group member, the group leader sends an ID update to the ID servers.

(2) When a group node leaves its group, if it becomes a single node, it will notify its home region to enable the location service for it; if it joins a new group, it sends its ID and location to the new group leader.

When a group node leaves its group, its old group leader needs to report the group membership change to the ID servers. If the node becomes a single node, it sends a location update to its original location servers in its original home region. Thus, the location query for it can be directly answered by its location servers. If the node joins a new group, its new group leader also reports the group membership change to the ID servers.

Thus, GrLS can support seamless handoff between single location management and group location management.

d) Query for Group Nodes

If *d* is a group member, its original location servers have been disabled. However, the source *s* does not know it due to distributed location service. So the location query message will still be sent to the original home region of *d*. When an original location server of *d* receives this message, it finds that the location information of *d* has been disabled. It then forwards the message toward the group home region, where the network center is the destination. Once the location query is received by a node in one subregion of the group home region, the node acts as a proxy. Since an ID server exists in each subregion, the proxy sends an ID query message to the ID server requesting the ID of *d*'s group leader. The ID server sends the requested group leader ID back to the proxy by an ID reply message. Then according to the strategy of recruiting location servers, the proxy can determine which node is the desired location server of *d*'s group leader. If the desired location server is just the proxy, it forwards the location query to *d*'s group leader. Otherwise, it forwards the location query message to the desired location server, which continuously forwards the message to *d*'s group leader. When *d*'s group leader receives the location query message for *d*, it sends a location reply to the source *s* directly.

If *d* is a group leader, the location query procedure is the same as other group members before the location query message arrives at one of the desired location servers of *d*. When the location server finds that it has the knowledge of *d*'s location, it directly sends a location reply to the source *s*.

No matter for single nodes or group nodes, neither broadcasting nor flooding is used in the location query procedures. Even local flooding is unnecessary. Location query messages from different network areas can be processed by the location servers in different subregions because the group home region lies at the network center. Therefore, on average, each border subregion of the group home region handles 1/6 of the location queries for group nodes. When one border subregion is empty, the location servers in the central subregion will be queried. So the query load can also be evenly distributed over the entire group home region. In addition, both ID query and ID reply only experience one hop transmission and incur trivial overhead.

4.4 Handling Mobility

4.4.1 Location and ID Information Handoff

Due to node mobility, nodes may move into or out of subregions in a home region. When a location server moves out of the subregion it resides in, the location information stored in it needs to be migrated to other nodes in the same subregion. Since the leaving node is aware of all other nodes in the same subregion, it can know which one has the next closest ID to each source for whom it is acting as the location server. Thus, each source will have a new location server. The leaving node then hands off the location information of each source to its new location server separately. If the leaving node is the ID server, it directly hands off the group membership information to the node with the next least ID in the same subregion. Then this node becomes the new ID server.

When a new node enters a subregion, all the other nodes in the subregion know it by beaconing. If one node finds that the new node has a “closer” ID to the sources of some location information it stores, it will hand off these location information to the new node. Thus the new node will play as the location server for all the sources of the migrated location information. If the ID server in this subregion finds that the new node has smaller ID than it, it will hand off all the group membership information to the new node, which becomes the new ID server.

Actually, only one handoff message is needed when an old location server leaves a subregion or a new node enters a subregion and becomes the new location server. This point is proved in Theorem 4.1. It is benefited from our strategy of recruiting location servers. Through such a simple location and ID information handoff procedure, the location update or query message can still reach the desired location servers regardless of the change of location servers or ID servers.

Theorem 4.1 Assume there exists a set of nodes $\{s_1, s_2, \cdots, s_{m-1}, s_m \mid s_1<s_2<\cdots<s_{m-1}<s_m\}$in subregion *SR*. When (1) node s_i ($i\in\{1, 2, \cdots, m-1, m\}$) leaves *SR* or (2) node s_j ($j\notin\{1, 2, \cdots, m-1, m\}$) enters *SR*, only one location information handoff message is needed.

Proof: As described in Section 4.3.1, the node ID space is assumed to be circular in clockwise direction from small IDs to large IDs, as shown in Figure 4.5. According to our strategy, a node s_k ($k\in\{1, 2, \cdots, m-1, m\}$) will be recruited as location servers by the nodes, which have selected this home region and have the ID between s_{k-1} and s_k. Here, s_0 represents s_m.

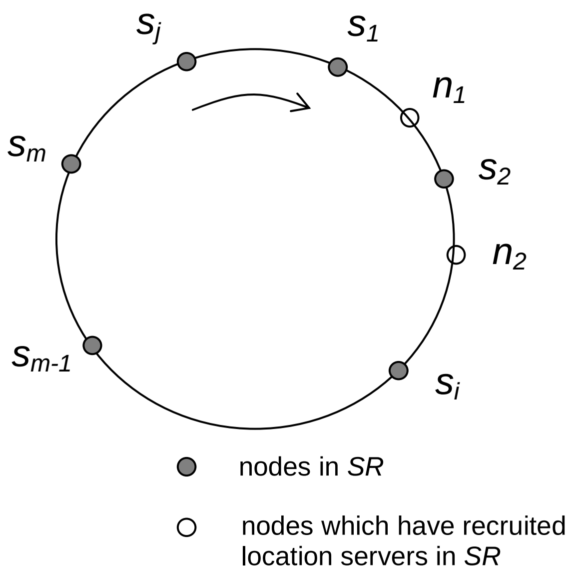


Fig. 4.5: A circular node ID space

When node s_i ($i \in \{1, 2, \cdots, m\text{-}1, m\}$) leaves *SR*, it needs to handoff all the location information stored in it to other nodes in *SR*. As we have analyzed, the IDs of all the sources which have stored location information in s_i are between s_{i-1} and s_i. Since s_i leaves, their new location servers will be the same node, s_{i+1}, by our strategy. Hence, s_i will handoff all the location information to s_{i+1}. So only one location information handoff message from s_i to s_{i+1} is needed.

Without loss of generality, we assume $s_k<s_j<s_{k+1}$ ($k \in \{1, 2, \cdots, m\text{-}1, m\}$ and $j \notin \{1, 2, \cdots, m\text{-}1, m\}$). Here, s_{m+1} represents s_0. When node s_j enters *SR*, it will be the new location server for all the nodes which have selected this home region and have the ID between s_k and s_j. Since all of these nodes have recruited the same location server s_{k+1}, only s_{k+1} needs to handoff their location information to s_j. So only one location information handoff message from s_{k+1} to s_j is needed. ■

4.4.2 Handling Empty Regions

1) Handling Empty Subregions

A home region consists of seven subregions: one central subregion and six border subregions. In a home region, one subregion may be empty, i.e., no nodes in it. The case that all the subregions are empty except the central one will not happen since we assume a connected ad hoc network.

If some border subregions are empty but the central subregion is non-empty, the location update message still can arrive at the desired location server in the central subregion. Upon receipt of the location update message, the location server in the central subregion unicasts it to the remaining unvisited subregions separately. The messages to empty subregions will be finally dropped. If the central subregion is empty, our location update forwarding strategy still works by exploiting the advantage of geographic forwarding. For example, GPSR [KAR00], a widely-used geographic routing protocol, can route packets around the perimeter of an empty region. If the central subregion is empty, the location update message is routed along a perimeter formed by nodes surrounding the central subregion. In each non-empty border subregion, at least one node will lie on the perimeter. So a location update message can be received by its location servers in all these non-empty subregions. Figure 4.6 illustrates an example of the location update procedure in a home region with an empty central subregion. Similarly, an ID update message can also arrive at all the ID servers in non-empty subregions.

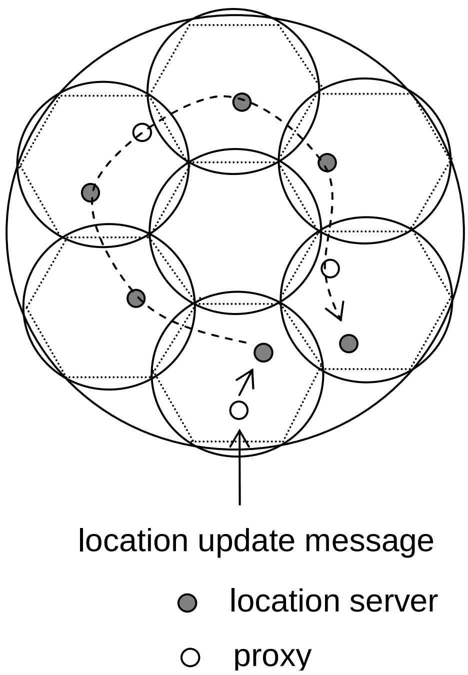


Fig. 4.6: The forwarding procedure of a location update in a home region with an empty central subregion

For the location query message, since its destination is also the center of the home region, similarly it can arrive at one non-empty subregion by exploiting the advantage of geographic forwarding. As a result, one location server will receive the location query if there exists at least one non-empty border subregion in the home region. The location server will directly send the location reply to the source or continuously forward the location query to the group home region. Figure 4.7 shows how a location query message arrives at a desired location server in a home region with empty subregions. In the group home region, once a location query message arrives at a non-empty subregion, the proxy node will send an ID query to the ID server in the same subregion.

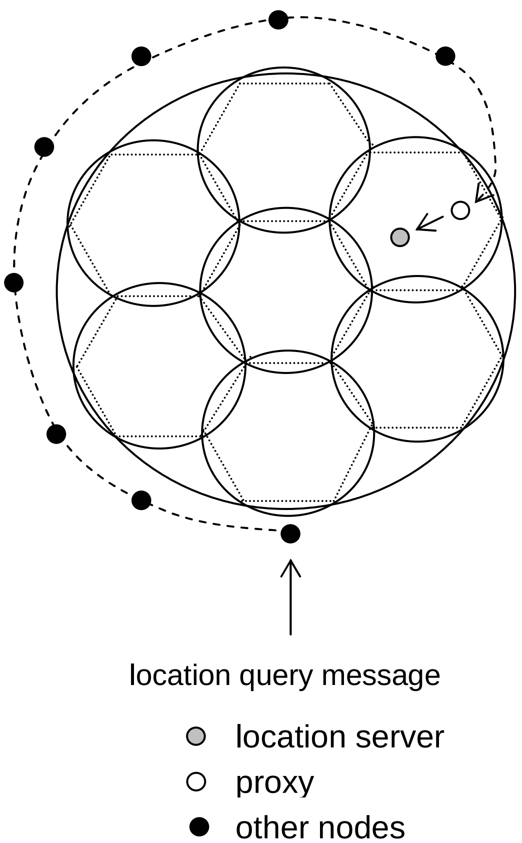


Fig. 4.7: The forwarding procedure of a location query in a home region with empty subregions

2) Handling Empty Home Region

Normally it is rare that the entire home region is empty for dense and large-scale MANETs. In the following, we give some suggestions to handle an empty home region.

If a node is the last node in a home region, when it wants to leave the region, it needs to hand off location information or group membership information stored in it to one neighboring region of the home region. The nearest neighboring region to the leaving node will be selected. Once a node in this neighboring region receives the migrated information from a different region, it forwards them to the node with the least ID in its subregion. The node with the least ID will store these migrated location or group membership information.

To make a node know it is the last node in the entire home region, the advantage of GPSR can be exploited again. As mentioned in Section 4.3.2, a location server appends its subregion ID to the location update message before forwarding it. When node *x* receives a location update or query message destined to the home region where it resides, and it finds that greedy forwarding is impossible**,** it will set the message to enter perimeter mode. By GPSR perimeter mode, if *x* is the only node in the home region, a message destined to the center of the home region will return to *x* again with only *x*'s subregion ID appended. So if *x* receives a duplicate location update message with only its subregion ID appended, *x* will know it is the last node in the home region. After that, when *x* receives the first location update message with other subregion ID appended or a forwarded message is not looped back, *x* will know some other nodes have entered this home region.

Since the nodes that have selected an empty region as their home region do not necessarily know about its emptiness, they may continue to send location update messages to it. Similarly, the location queries to these nodes will still be sent to the empty home region. When a location update or query message finds that the destined home region is empty, it will search the neighboring regions of the empty region. The node with the least ID in each subregion of each neighboring region is checked till the node, which has the migrated location information for the source of the message, is found.

When a new node enters an empty home region and receives a location query message which it cannot answer, the node will append its subregion ID to the message and continue to forward it toward the center of the home region. Similar to the last node in a home region, if the new node receives the duplicate location query message with only its subregion ID appended, it knows that there is still no location servers to answer the location query. Then it will forward the location query to the neighboring regions to search the desired location information as mentioned above. Clearly, the location update and query overhead incurred by empty home regions is higher than non-empty home regions.

4.5 Theoretical Analysis

In this section, we conduct theoretical analysis on the performance of GrLS. The feature of GrLS is that it exploits group location management to reduce protocol overhead. If group location management is not adopted in GrLS, all the nodes then need to send location update messages to their home regions even they have formed groups. This kind of scheme still works as a location service protocol and we denote it as GrLS-. Since GrLS- is a simplified version of GrLS, we first analyze GrLS-, and then extend the analysis to GrLS. To make the analysis tractable, referring to [DAS05], we also assume the network is static, the nodes are uniformly distributed in the geographic area, and location information is not cached at forwarding nodes. Since all the nodes stay stationary, the location update is only triggered by the maximum location update interval. Before proceeding further, let us introduce the following notations used in the analysis.

- N — *network size, i.e., number of nodes in the network*
- $T_{\text{loc\_upd\_max}}$ — *maximum location update interval*
- $T_{\text{net\_time}}$ — *network lifetime*
- $\overline{L_{upd}}$ — *average location update path length in GrLS-*
- $\overline{L_{que}}$ — *average location query path length in GrLS-*

- $\overline{L_{rep}}$ *average location reply path length in GrLS-*
- $\overline{g}$ *average group size in GrLS*
- n *number of groups in GrLS*
- *n_que* *number of location queries*

In GrLS-, a source node sends a location query to the home region of the destination node. Once the query reaches one border subregion of the destination's home region, a location server in this subregion will receive the query message and no longer forward it. By the above definition, the average hop number a query message travels is denoted as $\overline{L_{que}}$. Since a location reply message will be sent back to the source by the location server, the average hop number that a query message travels is also $\overline{L_{que}}$, i.e., $\overline{L_{rep}} = \overline{L_{que}}$. For a location update message, on average it also travels $\overline{L_{que}}$ hops to reach a location server in a border subregion of its home region. However, since the location servers located in the remaining six subregions also need to be updated, the update message will be further forwarded in the home region as shown in Figure 4.3. In each remaining subregion, the update message either directly reaches the location server or be forwarded to the location server by a proxy node. Hence, on average, the extra transmissions for updating the remaining six location servers are 9 (1.5 X 6) hops. So, we have

$$\overline{L_{upd}} = \overline{L_{que}} + 9 \qquad (4.1)$$

In GrLS, only single nodes and group leaders need to send location update messages to their location servers. For the single nodes, their average location update path length is also $\overline{L_{upd}}$. For the group leaders, they have the common group home region at the network center. Hence, their average location update path length is less than $\overline{L_{upd}}$. However, since $n << N$, to simplify the analysis we just assume that their average location update path length is also $\overline{L_{upd}}$.

We denote the total location update message transmissions in GrLS- and GrLS as U' and U, respectively. Based on the above analysis, we have

$$U' = N * \frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}} * \overline{L_{upd}} \quad (4.2)$$

and

$$U = (N - n\overline{g} + n) * \frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}} * \overline{L_{upd}} \quad (4.3)$$

In GrLS-, all the queries are for single nodes. However, in GrLS, the cases are different since the queries may be for the single nodes or group nodes. For queries to single nodes, the average hop length is also $\overline{L_{que}}$. The queries to group nodes can be divided into two categories: one is for group leaders and the other is for group members. Since $n << n\overline{g}$, to simplify the analysis we assume all the queries to group nodes are for group members. According to GrLS, a query for a group node is firstly forwarded to one of its original location servers with the average path length $\overline{L_{que}}$. Then it is forwarded to the group home region with average path length 3.8 hops ($(12*2\sqrt{3}R + 6*\sqrt{3}R)/(18r)$, referring to Figure 4.1). Finally, the query message is forwarded to the group leader of this group node with average path length 3 hops ($(2\sqrt{3}R + R)/(2r)$, also referring to Figure 4.1). Hence, in GrLS a query message for a group node travels $\overline{L_{que}} + 6.8$ hops on average.

We denote the total location query message transmissions in GrLS- and GrLS as Q' and Q, respectively. Based on the above analysis, we have

$$Q' = n\_que * \overline{L_{que}} \quad (4.4)$$

and

$$\begin{aligned} Q &= n\_que * (1 - \frac{n\overline{g}}{N}) * \overline{L_{que}} + n\_que * \frac{n\overline{g}}{N} * (\overline{L_{que}} + 6.8) \\ &= n\_que * \overline{L_{que}} + n\_que * \frac{n\overline{g}}{N} * 6.8 \end{aligned} \quad (4.5)$$

In GrLS, the average location reply path length is approximately the same as $\overline{L_{rep}}$. Hence, for the same *n_que* queries, the total location reply message transmissions triggered in GrLS are the same as GrLS-. Since we assum a static network and static groups, there is no ID update, ID query, ID reply, and location or ID information handoff messages triggered. In mobile networks, these control messages will be triggered. However, the number of these control messages is trivial compared to both location update and query messages, as shown and explained in Section 4.6.2.

Hence, compared to GrLS-, the reduction in the protocol overhead (i.e., number of control messages) of GrLS is

$$\begin{aligned}(U'+Q')-(U+Q) &= N*\frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}}*\overline{L_{upd}}+n\_que*\overline{L_{que}} \\ &\quad -(N-\overline{ng}+n)*\frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}}*\overline{L_{upd}}-(n\_que*\overline{L_{que}}+n\_que*\frac{\overline{\overline{ng}}}{N}*6.8) \\ &= (\overline{ng}-n)*\frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}}*\overline{L_{upd}}-n\_que*\frac{\overline{ng}}{N}*6.8\end{aligned} \tag{4.6}$$

Since $n \ll \overline{ng}$, we have

$$(U'+Q')-(U+Q) \approx \overline{ng}\left(\frac{T_{\text{net\_time}}}{T_{\text{max\_loc\_upd}}}*\overline{L_{upd}}-n\_que*\frac{6.8}{N}\right) \tag{4.7}$$

From Expression (4.7), since all the parameters except $\overline{ng}$ are the same for both GrLS- and GrLS, we can see $\overline{ng}$, i.e., the number of group nodes plays the most important role in the reduction of control messages. Hence, in theory, with more group nodes in the network, GrLS can reduce more protocol overhead. The following simulations also verify the theoretical declaration.

4.6 Performance Evaluation

To study the performance of GrLS, we implement it as well as geographic forwarding in the GloMoSim 2.03 [ZEN98]. GloMoSim is a widely used wireless network simulator with a comprehensive radio model. It is designed using the parallel discrete-event simulation capability provided by Parsec. For comparison purpose, we also implement GLS. Geographic forwarding adopts GPSR with activated perimeter mode. We use 802.11 MAC protocol with DCF and a transmission range of 250m. The network coverage area is a square of 3km X 3km, which can be partitioned into 19 full regions as shown in Figure 4.1. In the mobility scenarios, single nodes follow the random waypoint mobility model, where each node moves at a constant speed chosen randomly from a predefined speed range. The speed range is different for each simulation scenario. For group mobility, we use the RPGM model [HON99], where different group motion vectors are assigned for different groups. As mentioned in Section 4.3.2, the predefined update threshold is fixed at 125m, half of the transmission range. The minimum update interval is set to be 12.5 seconds, which is the approximate result of the update threshold divided by the average node speed (125m/10ms^{-1}). The simulation duration is 900 seconds. All these important simulation parameters are listed in Table 4.2.

We assume two types of network models: quasi-static ad hoc networks and mobile ad hoc networks. In the mobility model followed by quasi-static ad hoc networks, the pause time is set to be 30 seconds and the node speed range is [0ms^{-1}, 5ms^{-1}]. Since the location update threshold is 125m, in the quasi-static ad hoc networks, the nodes send the location update messages using the maximum update interval if their speeds are less than 3.125ms^{-1} (125m/40s). Quasi-static ad hoc networks simulate networks where nodes stay stationary or move slowly. In the mobility model followed by mobile ad hoc networks, the pause time is 0 seconds and the node speed range is set to be [5ms^{-1}, 20ms^{-1}]. We evaluate GrLS in four ad hoc network scenarios: a 450-node quasi-static, a 450-node mobile, a 900-node quasi-static, and a 900-node mobile ad hoc network.

Table 4.2: Simulation parameters

Parameter	Value
Simulation Time	900 sec
Simulation Area	3km x 3km
Transmission Range	250m
Speed Range	0 – $20ms^{-1}$
MAC Protocol	IEEE 802.11
Mobility Model	Random Waypoint, RPGM
Update Threshold	125m
Minimum Update Interval	12.5 sec
Maximum Update Interval	40 sec

4.6.1 Load Balance

In GrLS, by the hash function, each home region is selected by the approximate same number of nodes. Further, each home region is divided into seven subregions. Then, each node, which has selected this home region, recruits a location server in each of its subregions. Therefore, a location update message is received by the location servers evenly distributed in the home region. When a location query message arrives at the destined home region, it is replied directly by the location server in the first non-empty subregion that the message has reached. Thus the load of acting as location servers is well balanced over the entire network.

In the network, we count the total number of location update and query messages received by the desired location servers in all the subregions with the same subregion ID. Here, ID update and query messages are also counted as location update and query messages, respectively. To guarantee the fairness, we only use the subregions which belong to the 19 full regions. In GrLS, since there are seven subregions with subregion ID ranging from 0 to 6 in each home region, seven numerical values regarding these seven subregion IDs can be collected. These values are normalized by the total number of location update and query messages generated by all the nodes during the simulation. These normalized values are termed as the normalized LS load.

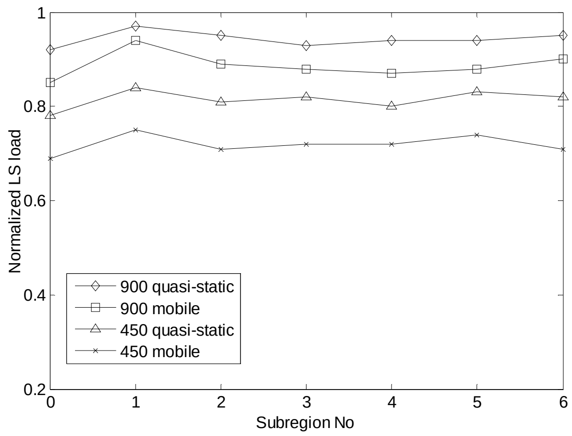


Fig. 4.8: Comparison of normalized LS load borne by different subregions

Figure 4.8 plots the normalized LS load in different kinds of subregions under the four network scenarios. It shows that the LS load is approximately evenly distributed in the network. For each network scenario, the load borne by subregion 0 is always less than other subregions. This is because most of the location query messages are received by border subregions (subregions 1-6) and will not reach the central subregion. Compared to both 900-node networks whose node density is 100 nodes per km^2, the LS load borne by subregions in both 450-node networks is lower. This is because some subregions are empty in 450-node networks due to low node density. From Figure 4.8, in both 900-node and 450-node networks, the LS load in mobile networks is always lower than the quasi-static networks. We think the reason is that high node mobility causes more drops of location update and query messages. Hence, for GrLS, the location service load is more evenly distributed in the networks with both higher node density and slower node mobility.

4.6.2 LS Protocol Overhead

Here, we compare the LS protocol overhead of GLS, GrLS-, and GrLS. In each of the four networks, every node initiates a location query to look up the location of a randomly chosen destination at times randomly distributed between 45 and 900 seconds. The first 45 seconds are used for nodes to send the initial location update messages to their location servers. When a node sends out a location query message, a location query timer is also set for this message. If no location reply is returned when the timer expires, the node does not re-send the location query. If a location reply is successfully received before the timer expires, the node sends a data packet of size 128 bytes to that destination using the replied location.

In each network, we count all the LS protocol messages for each location service protocol. The LS protocol messages of GrLS include location update, query, reply messages, ID update, query, reply messages, and both location and ID handoff messages. The LS protocol overhead is measured by the number of LS protocol messages transmitted, with each hop-wise transmission of the protocol message as one transmission. Then we evaluate the normalized LS protocol overhead (normalized by the number of LS protocol messages generated by GLS). Hence, the normalized LS protocol overhead of GLS is always 1.

Figure 4.9 plots the normalized protocol overhead of all these three protocols. It shows that GLS always has the maximum overhead. In mobile networks, the gap between GLS and other two protocols is much larger than in static networks. GLS incurs high protocol overhead because it relies on node chain consisting of mobile nodes to update and query location information. In a gird with high hierarchy level, a location update message needs to travel almost the whole grid to search its location server. Furthermore, huge amount of location update messages are triggered in highly mobile networks because nodes frequently cross grid boundaries. Both GrLS- and GrLS rely on home regions with fixed locations to update and query location information. Hence, they are more robust to node mobility. In addition, only one location update message needs to be sent to its home region per location update and at most seven location servers need to be updated within the home region. So both GrLS- and GrLS incur lower protocol overhead than GLS.

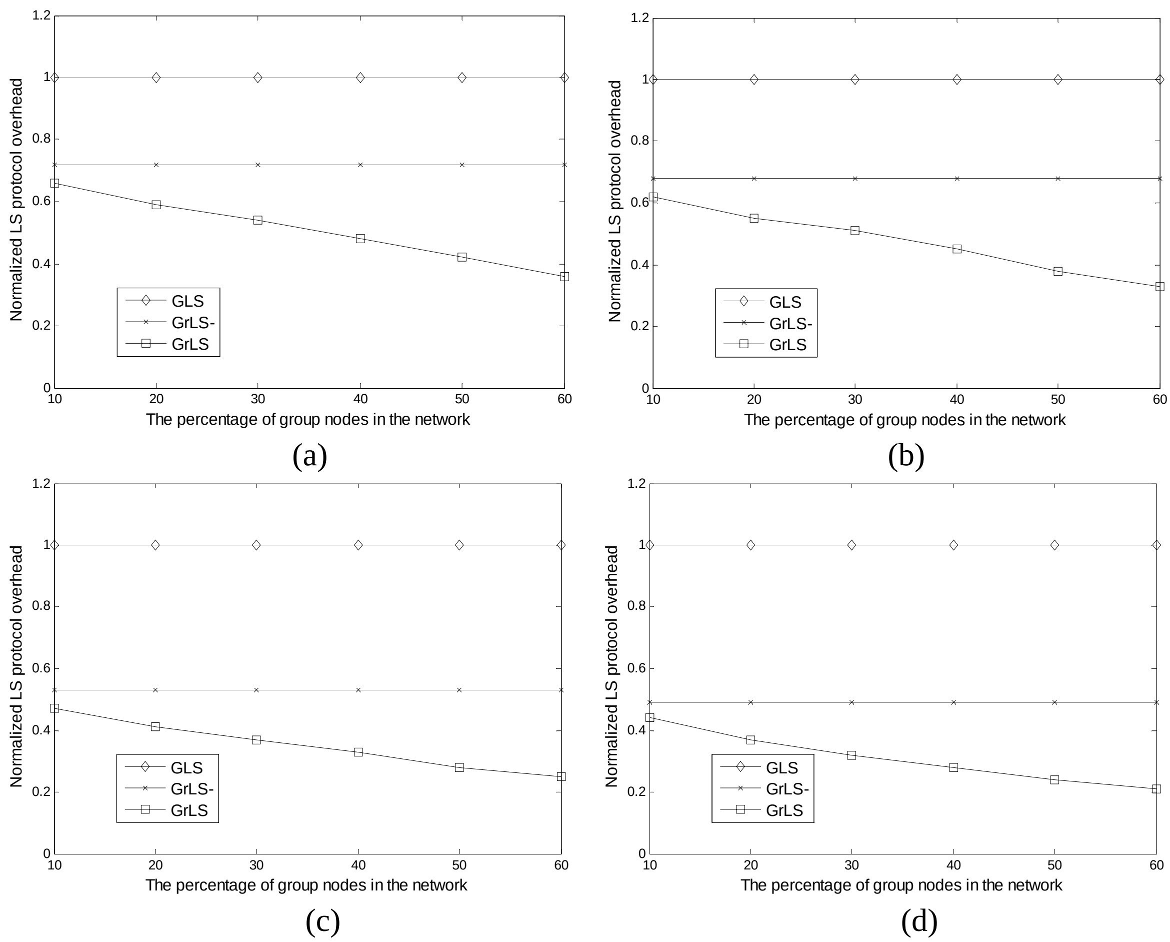


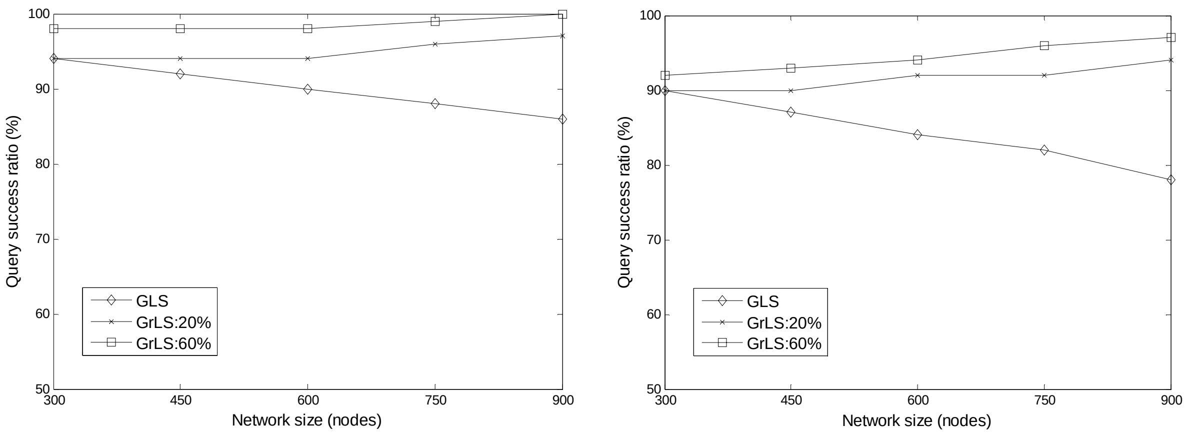
Fig. 4.9: Comparison of normalized LS overhead in: (a) 450-node quasi-static network, (b) 900-node quasi-static network, (c) 450-node mobile network, (d) 900-node mobile network

Compared to GrLS-, the protocol overhead of GrLS is reduced significantly. This is because the nodes which have formed groups except the group leaders do not need to send location update messages to their home regions in GrLS. ID update, query, and reply are firstly introduced by GrLS. Since we assume relatively stable groups, the reactive ID update rarely occurs. Both ID query and ID reply are triggered only when a location query message sent for a group member has reached the group home region. Moreover, both of them are just one-hop transmission. So these three new control messages account for a small portion of the LS protocol messages. In addition, we prove that only one handoff message is needed each time a node leaves or enters one subregion. The amount of handoff messages also depends on the node mobility. Higher node mobility leads to more handoff messages. Hence, the amount of handoff messages stays roughly the same in both GrLS- and GrLS. So the saving of location update messages contributes to the reduction in the protocol overhead of GrLS. As the percentage of group nodes increases in the network, more reductions in LS protocol overhead are achieved by GrLS.

4.6.3 Query Success Ratio

The objective of the location service is to help the source node get the location of a destination. Hence, an important metric to evaluate the location service protocol is the query success ratio. The query success ratio is the ratio of the number of location replies received by all the sources to the number of location queries initiated by all the sources. As stated in Section 4.6.2, each node initiates a location query to look up the location of a randomly chosen destination. If the location query fails, no retransmission is triggered. Here, we compare GrLS with GLS. To investigate the effect of group location management on GrLS, we choose two cases for evaluation: one is the percentage of group nodes is 20% in the network, and the other is 60%.

Figure 4.10(a) and (b) depict the query success ratio as a function of the network size for GLS, GrLS with 20% group nodes, and GrLS with 60% group nodes. The difference between these two figures is that the networks used in Figure 4.10(a) are all quasi-static, but the networks used in Figure 4.10(b) are all mobile. The results show that the query success ratio of GLS is always the lowest and drops quickly as the network size increases. Moreover, the query success ratio of GLS is much lower in the mobile networks than in the quasi-static networks. As we have explained in Section 4.6.2, GLS is the most susceptible to node mobility by relying on node chains. Furthermore, as the network size increases, the node chains for both location update and query become longer and weaker, which reduce the query success ratio and the location information accuracy.



(a) (b)

Fig. 4.10: Comparison of query success ratio in: (a) quasi-static networks, (b) mobile networks

With the network size increasing, the node density also becomes higher. In GrLS, more nodes can play as location servers in each home region due to high node density. Since the query success ratio is relatively high at 300 nodes for GrLS, it increases slowly when the network size goes beyond 300. In addition, as the percentage of group nodes increases from 20% to 60%, more source-destination pairs are within the same group. It increases the probability that the source node can get the location of the destination immediately, which also helps improve the query success ratio. By using group location management, GrLS has very good performance under traffic patterns with locality. Like other protocols, high node mobility also reduces the performance of GrLS as shown by the query success ratios in Figure 4.10.

4.6.4 Average Query Hop Length

If the location query is for a group node, GrLS will forward it to the group home region when the location information of the group node is found to be disabled in the original location servers. If the group node is exactly a group leader, the location server in the group home region will answer the location query immediately; otherwise, the location query will be continuously forwarded to the leader of the group node. Thus the query hop length of GrLS is incremented when querying group nodes. To investigate how much the query length is affected, we compare the average query hop length in GLS, GrLS- and GrLS. Similarly, to see the effect of group location management on GrLS, we still adopt the two different cases of GrLS as used in Section 4.6.3.

Figure 4.11 plots the average query hop length under the four protocol cases. It shows for all except GLS, the average query hop length drops as the network size increases. For GLS, the query success ratio becomes lower as the network size increases, which also leads to longer average query hop length. The average query hop length of GrLS- is shorter than GrLS due to no need forwarding the location query messages for group nodes to their group leaders. For GrLS with 60% group nodes, its average query hop length is longer than GrLS with 20% group nodes but not much. This also benefits from the locality. When most of the nodes have formed groups, the probability that a source-destination pair is within the same group becomes higher. When it happens, the query hop length is 0, which helps reduce the average query hop length. So the increment in the average query hop length is trivial although the percentage of group nodes increases from 20% to 60% in the network.

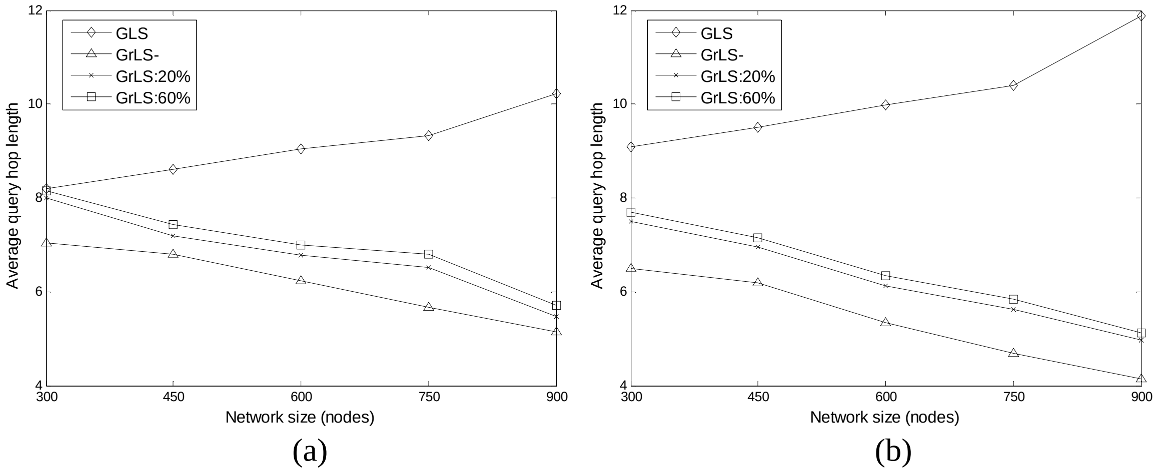


Fig. 4.11: Comparison of average query hop length in: (a) quasi-static networks, (b) mobile networks

4.7 Summary

In this chapter, we propose the first group-based location service protocol, GrLS. By exploiting group mobility, GrLS provides group location management for nodes which have formed groups, thereby reducing the protocol overhead significantly. Moreover, GrLS supports seamless handoff between single location management and group location management. Extensive simulations were conducted to compare GrLS with GLS and GrLS-, which is GrLS without group location management. The results show that GrLS has decent load balance, low protocol overhead, and high query success ratio. The cost that GrLS pays for the performance improvement is the increase of the average query hop length compared to GrLS-. Even so, it is still smaller than GLS. A good location service protocol should be efficient, scalable, robust, load balanced, and locality aware. GrLS shows all these characteristics.

Chapter 5

GMZRP: Geography-aided Multicast Zone Routing Protocol in MANETs

In this chapter, we introduce the proposed *G*eography-aided *M*ulticast *Z*one *R*outing *P*rotocol (GMZRP). This chapter is arranged as follows. Firstly, a brief overview is given in Section 5.1. Section 5.2 describes the system model and data structure. The detailed design of the GMZRP protocol is presented in Section 5.3. Performance evaluation is presented in Section 5.4. Finally, Section 5.5 summarizes this chapter.

5.1 Overview

Two types of routing protocols for mobile ad hoc networks have been studied in the literature: topological routing and geographical routing. Topological routing can usually find the shortest path, in number of hops, between each pair of nodes. However, it has difficulties in handling large-scale ad hoc networks with frequently changing connectivity among nodes. Geographic routing only needs to maintain local tables about its neighbors, and hence it is scalable to a large number of network nodes and efficient when network topology may change frequently. However, geographic routing needs the support of high accuracy location technology and location service protocol. In addition, it needs to handle the “void” problem encountered by a node, which fails to find a neighboring node that is geographically closer to the destination.

Since both topological routing and geographic routing have their own characteristics, intuitively, efficient hybrid routing can be developed by exploiting the strength and avoiding the weakness of each type. Here, hybrid routing refers to a novel type of routing methodology which combines both topologically driven route optimization and geographically driven route optimization. Recently, researchers have proposed a few hybrid routing protocols [CAO06, Du06, GAO05, GAO06, HSI04, LIM05, MAR99, SIV05, STO01, ZHO05].

In MANETs, group communications are important due to the trend of mobile nodes to work in a cooperative way [LAW05]. Group communications need the support of multicast routing protocols. GMZRP is a hybrid multicast routing protocol, which operates in an on-demand fashion and utilizes geographic partition to reduce route discovery overhead. The main idea of GMZRP is inspired by the well-known routing protocol ZRP [HAS01], in which a route query is originated at the source and spreads throughout the network. Unfortunately, in ZRP, a node will receive a lot of duplicate queries due to zone overlapping even under the case that the nodes are uniformly distributed in the network. GMZRP partitions the network coverage area into small zones and guides the route request packets outward using geographic information. It can guarantee that each zone is queried only once given an even distribution of the network nodes. In addition, the unicast route request procedure in ZRP has been extended to a multicast tree discovery procedure in GMZRP. To the best of our knowledge, GMZRP is the first one that well exploits the symmetrical geographic zone partition in the guidance of the multicast route query so as to discover the multicast forwarding tree with the lowest overhead and within the shortest time.

Another important feature of GMZRP is that it maintains a multicast forwarding tree at two types of granularities: the zone granularity (the sequential geographic zones that the tree spans in a source routing manner) and the node granularity (the sequential nodes that the tree spans in a hop-by-hop way). With the zone granularity, for each receiver, the source keeps a zone ID chain connecting the source zone to the corresponding receiver zone. An intermediate forwarding node also keeps a zone ID chain connecting its own zone to each downstream receiver zone, which is part of the zone ID chain kept by the source to the same receiver zone. Therefore, at the zone level GMZRP looks like source routing. With the node granularity, the source or an intermediate forwarding node only keeps the information about its child nodes. The multicast packet is forwarded along the multicast tree hop by hop. Owing to simplicity of the zone ID chains, the additional geographic information kept by the source and forwarding nodes occupies small storage and incurs a little overhead. However, they can help recover route breakage easily as shown later.

We compare the performance to that of ODMRP [LEE00] running on the same simulation scenarios. The results of the performance evaluation of GMZRP show that, comparing with a well-known multicast protocol On-Demand Multicast Routing Protocol (ODMRP), GMZRP has lower protocol overhead and, meanwhile, achieves competing packet delivery ratio and shorter delivery latency.

5.2 System Model and Data Structures

In the network, each node is equipped with GPS devices and knows its own position. In GMZRP, geographic partition is utilized to help reduce the route discovery overhead, as will be shown in Section 5.3.2. We propose a network center based partition method as introduced below, which is similar to the partition method proposed in Section 4.2.

The center of the network coverage area is roughly estimated at the time the MANET is initialized. The partition starts at the network center and spreads outward. As will be shown in Figure 5.1, the area is partitioned into equal circle-shaped zones. As the dotted lines show, each circle contains a hexagon with the side length equal to the radius of the circle. These hexagons are non-overlapping but can completely cover the entire network. Thus, there is a central zone at the network center and many other zones spread around the central zone symmetrically. Each zone has a unique zone ID. Each zone has six neighboring zones since a hexagon has six sides. We denote the radius of the circle as R, $R = 0.5*r$, where r is the radio transmission range of the mobile nodes. As a result, each node is the direct neighbor of any other node within the same zone. At startup, all nodes know the network center and the partition method. Hence, each node knows its own zone ID and any other node's zone ID given that node's position.

The primary fields of the Multicast Route REQuest (MRREQ) packet used by multicast forwarding tree discovery in GMZRP are as <*Broadcast_ID*, **s**, **G**, *Source_Zone*, *Hop_Cnt*>. The *Broadcast_ID*, together with the source node's ID **s** and multicast group address **G**, uniquely identifies each MRREQ packet. The *Broadcast_ID* is incremented for each MRREQ packet the source initiates for the same group. The *Hop_Cnt* is initialized by **s** to 0 and is incremented by each node forwarding the packet. The primary fields of the MRREP (Multicast Route REPly) packet used by multicast forwarding tree discovery are as <*Broadcast_ID*, **s**, **G,** **R**, *Receiver_Zone*, *Zone_ID_Chain*>. The *Zone_ID_Chain* records the IDs of the zones that the MRREP packet has passed and the latest one will be placed at the beginning of the chain.

The multicast forwarding state for GMZRP is maintained locally by each node in the following three tables:

- *Source Table*: Logically contains one entry for each multicast group **G** for which this node is an active sender. Each entry in the Source Table includes the node ID and zone ID of each receiver in **G** and zone ID chains, each of which connects the source zone to a receiver zone. Each entry also includes the node ID and zone ID of each child node of the source on the multicast forwarding tree.

- *Membership Table*: Logically contains one entry for each combination of multicast group **G** and source **s** for which this node is either a receiver member or a forwarding node. Each entry in the Membership Table includes a flag to indicate if this node is a receiver, and a flag to indicate if this node is a forwarding node. If it is a forwarding node, the entry includes zone ID chains, each of which connects the forwarding node's zone to a downstream receiver zone. The entry also includes the node ID and zone ID of its each child node on the multicast forwarding tree.
- *Node Table*: Logically contains one entry for each other node in the network from which this node has received a MRREQ packet. Each entry in the Node Table includes the *Broadcast_ID* from the most recent MRREQ packet, plus a bitmap representing a number of previous *Broadcast_ID* of MRREQ packets from this source, used to detect and discard duplicate MRREQ packets. Each entry in the Node Table also includes the node ID and zone ID of the previous hop as the previous hop address, taken from the transmitting source address of the MRREQ packet received from this source with this *Broadcast_ID* that contained the minimum hop count.

5.3 Description of the GMZRP Protocol

Multicast sources and receivers using GMZRP cooperate to establish and maintain forwarding state in the network to allow multicast communication. In GMZRP, the multicast forwarding state for a given multicast group **G** and source **s** is conceptually represented as a loosely-structured multicast forwarding tree rooted at **s**. Each multicast packet is dynamically forwarded from **s** through the tree to the receiver members of the multicast group **G**.

In GMZRP, source-based multicast forwarding trees are created whenever there is at least one source and one receiver in the network. GMZRP is designed to work independently of the geographic unicast protocol used in the networks and can thus work with any geographic unicast protocol or even a simple greedy geographic forwarding. GMZRP currently operates only over bidirectional links.

5.3.1 Multicast Packet Forwarding

When a node **R** receives a data packet originated from **s** and destined to a multicast group **G**, it firstly checks its Membership Table entry for this group and source to determine if it should forward the packet. If it is only a forwarding node, the packet thus flows along the multicast forwarding tree from the source to the downstream receivers. Then all its child nodes on the tree will receive the data packet.

In addition to forwarding the packet if required, node **R** also checks its Membership Table entry for this group and source to determine if it is a receiver member. If so, then **R** will pass the packet up to the next layer within the protocol stack to allow the packet to be further processed as a received multicast packet.

5.3.2 Multicast Forwarding Tree Discovery

A MRREQ packet, is initiated by a source node **s** that has data packets for **G** but no a Source Table entry for this multicast group. In this case, node **s** creates and initializes a new Source Table entry for **G**. The source **s** then waits for replies in the form of MRREP packets from the receivers. If no reply is received from one or more receivers after a waiting period, **s** sends a new MRREQ. To avoid congesting the network, sending of the

MRREQ packet is separated by an increasing interval using binary exponential backoff [VAL05]. If the source still cannot receive any reply after a specified number of retries, the upper layer is informed that **G** is not reachable.

Once **s** receives at least one MRREP packet, **s** then begins sending normal multicast packets. However, it is possible that some interested receivers did not receive this initial MRREQ packet from **s** or some other receivers wish to leave the group **G** after a certain time. To allow for such occurrences caused by dynamic group membership, node **s** will rebroadcast the same MRREQ packet with incremented *Broadcast_ID* after a period of time. The time between each MRREQ broadcast is increased until reaching a slow background rate, designed to tolerate factors such as intermittent wireless interference or temporary partition of the network. By means of this, the multicast tree might be refined in reaction to membership change. Stale routes may be purged and new ones created.

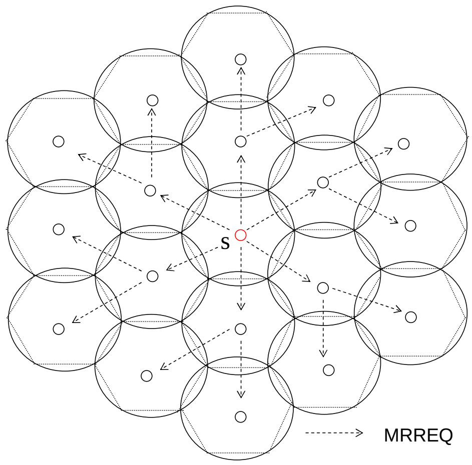
1) Inter-Zone Multicast Route Request Propagation

In most prior works [CHI98, JET01, LEE00], the route request was flooded in the network, which incurs high overhead. For ZRP, as introduced in Section 2.4.1, a few mechanisms based on topological information have been proposed to help reduce the duplicate queries. However, those solutions still have intrinsic problems since the network topology cannot provide effective information for guidance. Based on the above observations, GMZRP utilizes geographic partition to guide the multicast route request propagation. Clearly, the idea case is that each zone is to be queried only once.

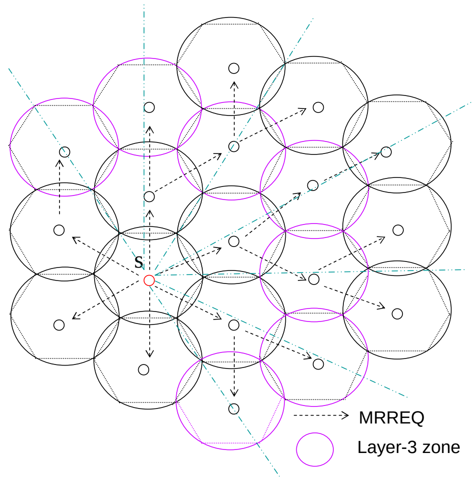
As shown in Figure 5.1, by our partition method the zones are distributed around the network center symmetrically. We mark the central zone as the first layer zone, the 6 zones neighboring to the first layer zone as the second layer zones, and the 12 zones neighboring to the second layer zones as the third layer zones, and so on. Based on the symmetrical zone partition, a simple but effective strategy is developed to avoid duplicate queries during the MRREQ packet propagation. In the following, we firstly describe and illustrate the strategy in a network covered by three layers of zones as shown in Figure 5.1. Then we derive the formal strategy used in a more general case where the network is covered by zones of *n* layers.

Assume that the source node **s** is residing in the central zone, which is hence claimed as the source zone. It firstly forwards the MRREQ packet to each layer-2 zone using geographic forwarding where the center of each zone is the destination. Then each layer-2 zone needs to further forward the MRREQ packet to layer-3 zones. How to avoid the case that two layer-2 zones forward the same MRREQ packet to the same layer-3 zone, i.e., duplicate queries? Firstly, we draw a line to connect the center of the source zone and the center of a layer-2 zone. We let the line pass through layer-3 zones and clearly each such line passes through only one layer-3 zone. Because the number of layer-3 zones is double the number of layer-2 zones, each layer-2 zone will forward the MRREQ packet to two layer-3 zones: the zone the line passes and this zone's neighboring zone in clockwise direction. As shown in Figure 5.1(a), there is no duplicate query in the network.

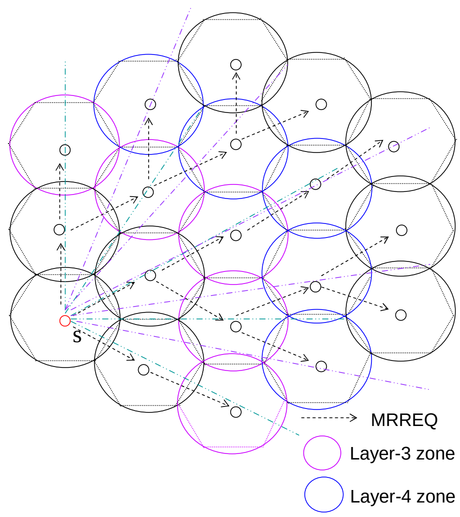
If the source node **s** resides in one layer-2 zone with respect to the network center, this zone becomes the source zone and we regard this source zone as the first layer zone during the propagation. Then we can see that there are three additional layers of zones around the source zone. Benefited from the partition, no matter where the source zone is, the zone partition covering the network is still part of a symmetrical zone partition around the current source zone. Hence, the above mentioned strategy of MRREQ packet propagation from layer-2 zones to layer-3 zones still applies. But, we need to decide how the layer-3 zones forward the MRREQ packets to layer-4 zones. The method is similar. We firstly draw a line to connect the center of the source zone and the center of a layer-3 zone. We let this line pass through layer-4 zones. Then there are two different cases: one is that the line passes through only one layer-4 zone and the other is that the line passes through the intersection part of two neighboring layer-4 zones. For the first case, the layer-3 zone will forward the MRREQ packet to that layer-4 zone only; and for the second case, the layer-3 zone will forward the MRREQ packet to those two layer-4 zones, respectively. As shown in Figure 5.1(b), there is also no duplicate query in the network.



(a) From the first layer source zone



(b) From the second layer source zone



(c) From the third layer source zone

Fig. 5.1: The propagation of the MRREQ packets originated from zones staying at different layers

Finally, if the source node **s** is residing in a layer-3 zone with respect to the network center, accordingly this zone becomes the source zone, which is also regarded as the first layer zone during the propagation. We can see that there are four additional layers of zones around the source zone. The above mentioned strategy of MRREQ packet propagation from layer-3 zones to layer-4 zones still applies. What remains is that we need to decide how the layer-4 zones forward the MRREQ packets to the layer-5 zones. The method is exactly the same as the one used from layer-3 zones to layer-4 zones. We firstly draw a line to connect the center of the source zone and the center of a layer-4 zone. We let this line pass through layer-5 zones. Then there are also two different cases: one is that the line passes through only one layer-5 zone and the other is that the line passes through the intersection part of two neighboring layer-5 zones. For the first case, the layer-4 zone will forward the MRREQ packet to that layer-5 zone only; and for the second case, the layer-4 zone will forward the MRREQ packet to those two layer-5 zones, respectively. As shown in Figure 5.1(c), there is also no duplicate query in the network.

From the above description, we can see that an intermediate zone makes the MRREQ forwarding decision based on both its own position and the source zone's position. We derive the following rules, which build up the strategy of the MRREQ packet propagation. Here, the source zone is denoted as the first layer zone.

a) A layer-1 zone forwards a MRREQ packet to each layer-2 zone, separately;
b) A layer-2 zone respectively forwards a MRREQ packet to two layer-3 zones if both of them exist. These two layer-3 zones are the one passed by the line connecting the center of the source zone and the center of the corresponding layer-2 zone, and its neighboring zone in clockwise direction.
c) A layer-3 zone forwards a MRREQ packet to one or two layer-4 zones if they exist. The one or two layer-4 zones are the one(s) passed by the line connecting the center of the source zone and the center of the corresponding layer-3 zone.

d) A layer-4 zone forwards a MRREQ packet to one or two layer-5 zones if they exist. The one or two layer-5 zones are the one(s) passed by the line connecting the center of the source zone and the center of the corresponding layer-4 zone.

We then derive the following common rule, which can be proved by the symmetry properties of our partition method.

e) For n ($n > 2$), a layer-n zone forwards a MRREQ packet to one or two layer-(n+1) zones if they exist. The one or two layer-(n+1) zones are the one(s) passed by the line connecting the center of the source zone and the center of the corresponding layer-n zone.

In addition, the proposed strategy can guarantee that the MRREQ packets reach all the network nodes with minimum delay. This property is also illustrated in Figure 5.1. A MRREQ packet originated at the source zone (i.e., layer-1 zone) needs n-1 times of inter-zone forwarding before it reaches one layer-n zone, which is just the shortest distance at the zone level.

2) Intra-Zone Multicast Route Request Propagation

When a node receives a MRREQ packet, it firstly checks if this packet is destined to its own zone. If not, it does not process the packet but instead discard the packet. But for an appropriate (fresh and destined to its own zone) MRREQ packet, it compares the hop count recorded in the received packet to the hop count in this node's Node Table entry for the same source, group and *Broadcast_ID*. If the new hop count is less than that already recorded in the Node Table entry, this node updates the entry with the new hop count and sets the previous hop address in the entry to the node ID and zone ID of the previous hop from which it received the packet. In addition, if the node forwards the MRREQ packet, before doing so, it increments the *Hop_Cnt* field and copies its own ID and zone ID into the packet's previous hop address field.

If one node finds that it is the first one to receive a MRREQ packet in its zone, it will broadcast the MRREQ packet to its zone. Since all the nodes within the same zone are its 1-hop neighbors, the Wireless Multicast Advantage (WMA) [THA05] can be exploited for broadcast. WMA refers to that a single transmission can be received by all the nodes that are within the transmission range of a transmitting node. Upon receipt of the MRREQ packet after the zone broadcast, by the proposed inter-zone MRREQ propagation strategy, each node can determine which one(s) of the next layer zones that the MRREQ packet should be forwarded to.

Referring to the energy-efficient Route Request forwarding method used in [DU06], we propose the following strategy for intra-zone MRREQ forwarding. A node will forward the MRREQ packet to the specific next layer zone with a delay of $T_d = \alpha / Nbr + T_r$, where *Nbr* is the number of the node's 1-hop neighbors which reside in the specific next layer zone, α is a system parameter that can be adjusted, and T_r is a small random backoff time. By exploiting the wireless multicast advantage, if a node hears the forwarding to the specific next layer zone from some other node in the same zone, it knows that the MRREQ packet has already been forwarded to that zone. Then it will not forward the MRREQ packet again. This avoids duplicate queries to the same next layer zone, leading to the reduction in routing overhead. Due to the delay α / Nbr, only the nodes with more direct neighbors residing in the specific next layer zone would participate in the routing. Clearly, this helps improve the robustness of the route connecting these two neighboring zones. The small random backoff time T_r is used to avoid simultaneous forwarding of the same MRREQ packet by several nodes having almost the same value *Nbr*. The value of α is chosen to be large enough so that T_d is different for different *Nbr*, but α should not be too large; otherwise, it may cause a large routing delay. If the nodes find that they are staying in a border zone, they will not forward the MRREQ packet again.

3) Multicast Route Reply Propagation

As the MRREQ packet is broadcast across the network, nodes set up pointers to establish the reverse path. When a node **R** receives a MRREQ packet and it is an interested receiver, then **R** replies with a MRREP packet, to cause the necessary nodes along the path back to the source **s** to become forwarding nodes. **R** also creates a new entry in its Membership Table and sets the receiver flag. The MRREP packet then follows the path established by the forwarding of the received MRREQ packet, as recorded in the previous hop address field in each node's Node Table entry for this source **s**. When the MRREP packet enters a new zone, the new zone ID will also be recorded into the packet and thus a zone ID chain will be formed and gradually expanded. Each node that forwards the MRREP packet, if it does not already have a Membership Table entry for this group and source, creates a new entry setting the forwarder flag and recording all its child nodes from which it has received appropriate MRREP packets. The new entry also records the current zone ID chain carried by the MRREP packet. When the source receives an appropriate MRREP packet, it will record the node ID and zone ID of the child from which it has received this MRREP packet. It also records the current zone ID chain carried by the MRREP packet.

If multiple new receivers are staying in zones that have received the MRREQ packets forwarded from the same zone except the source zone, a few MRREP packets will traverse the same paths or subpaths on their way to the source **s**. However, in order to make each node along the common paths a forwarding node for **G** and **s**, it is enough for one MRREP packet to be received and forwarded by each such node. Therefore, GMZRP filters all but the first two of these multiple MRREP packets received by each of these nodes.

4) Example

Figure 5.2 shows the propagation of a multicast route request from a source **s** to a multicast group **G**={r1, r2, r3} and the reverse multicast route reply. The MRREQ packets reach all the three receivers along the shortest paths determined by the proposed inter-zone MRREQ propagation strategy. Once a receiver receives a MRREQ packet, it replies a MRREP packet back to the source along the reverse route, which is also the shortest path. Since the zones where r2 and r3 are staying will receive MRREQ packets forwarded from the same zone where f1 is staying, when f1 receives the same MRREP packets from both r2 and r3, it only forwards one MRREP packet to the source, thereby reducing overhead. For simplicity, Figure 5.2 illustrates the multicast route request and reply only at the zone-level layer.

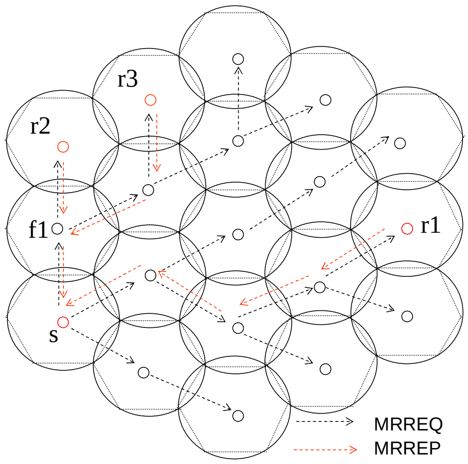


Fig. 5.2: Example of a multicast tree discovery

5.3.3 Location-guided Multicast Tree Optimization

Once **s** receives the MRREP packets from receivers, **s** can further optimize the multicast tree by geographic information. For example, in Figure 5.3, when **s** receives the MRREP packets from r2 and r3, **s** will find that they are staying in neighboring zones. Since the multicast packets from **s** to r2 will traverse inter-zone forwarding two times, **s** will ask r2 to join r3 for receiving multicast packets, thereby reducing one inter-zone forwarding. Once the route to a receiver is changed, a RouteUpdate packet will be reported to the source along the new routing path to update the multicast forwarding states kept by the forwarding nodes and the source. We can see that this is location-guided adaptive multicast tree optimization.

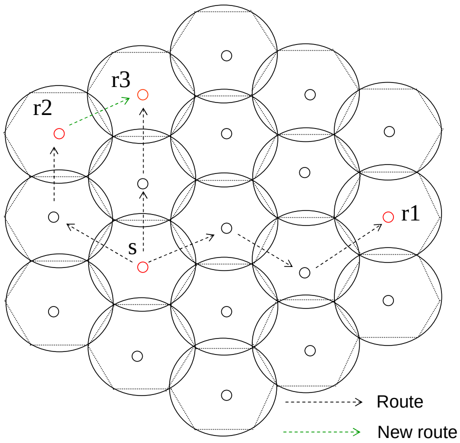


Fig. 5.3: Location-guided multicast tree optimization

5.3.4 Local Subtree Repair

We assume that nodes in the network may move at any time or fail due to battery exhaustion. Therefore, a link on the multicast forwarding tree may break due to node mobility or failure. When an intermediate node **C** detects that one of its child nodes on the multicast forwarding tree is unreachable, it initiates a local repair of the multicast forwarding tree. This child node may be the ancestor of one or more receivers. If all the downstream receivers reside in the same receiver zone, node **C** will send a RepairNotification packet to the original zone where the child resided, to query if there exist other ancestors of receivers in that receiver zone. If so, node **C** will establish a new link or path to the selected ancestor. Otherwise, the RepairNotification packet will be further forwarded to the next zone specified by the zone ID chain, which is kept by node **C** and ends at the receiver zone. The forwarding of the RepairNotification packet will continue until one ancestor is found or the receiver zone is reached.

If the downstream receivers of the unreachable child node belong to different receiver zones (i.e., the child node is a branch node), node **C** will generate multiple RepairNotification packets, each of which travels along the corresponding zone ID chain kept by node **C** to the corresponding receiver zone. The returned new paths will be merged if necessary. As shown above, GMZRP can easily recover the route breakage with the guidance of the zone ID chain. Furthermore, since the new routing path follows the same zone ID chain formed by the original route discovery, the existing part of the original multicast tree will be utilized as much as possible. Hence, the change made to the multicast forwarding tree is also minor.

5.3.5 Handling Special Zones

Clearly, the strategy for the inter-zone MRREQ packet propagation will work well in the networks where nodes are uniformly distributed with a reasonable density. In this kind of network, there is no empty zone and two neighboring zones can directly communicate with each other without going through the third one. However, it is still possible that some routing zone may be empty or unreachable from one of its neighboring zones, which we denote as a special zone. For an empty zone, since every node in the network needs to be queried, we must make sure it is really empty. Then its upper layer zone will take the responsibility for it to forward the MRREQ packets to its next layer zone(s). For a zone, which cannot be directly reached from one of its neighboring zones as required, we must establish a tunnel to connect them to guarantee the protocol's correct operation and consistency.

GMZRP handles the above problems by exploiting the advantage of geographic forwarding. For example, GPSR [KAR00], a widely-used geographic routing protocol, can route packets around the perimeter of an empty zone. When a MRREQ packet encounters a special zone, no matter empty zone or unreachable zone, the MRREQ packet will be routed along a perimeter formed by nodes surrounding the zone, as illustrated in Figure 5.4. Then two cases occur: one is that the packet returns back to the original zone where the packet entered the Perimeter mode; and the other is that the packet gets an entry to the zone during the perimeter traveling. For the former case, the empty zone's upper layer zone will send the MRREQ packets to the empty zone's next layer zone(s) by unicast tunnels. For the latter case, a unicast tunnel is also established along the path found by the MRREQ packet with Perimeter mode, to facilitate the next forwarding between these two neighboring zones.

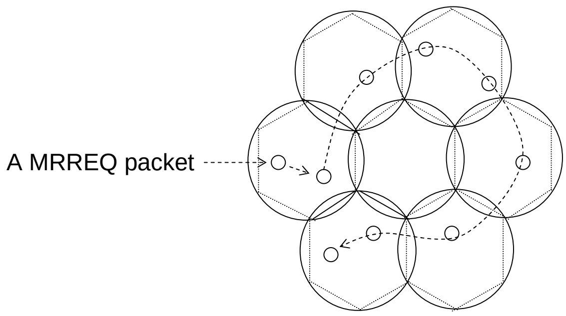


Fig. 5.4: Routing a MRREQ packet with Perimeter mode

5.4 Performance Evaluation

We evaluate the performance of GMZRP through detailed packet-level simulation in a variety of mobility and communication scenarios. In addition, we simulate the ODMRP, the best-studied on-demand multicast protocol for MANETs. ODMRP periodically floods the network with a control packet to re-create the multicast forwarding state. It allows redundant forwarding to each receiver, and hence increases the packet delivery ratio. We compare the obtained performance of GMZRP with that of ODMRP.

The cost of GPS devices becomes cheaper and cheaper. Aircraft, cars, handheld mobile terminals, and search-and-rescue teams all currently use GPS. MANETs are mainly used for disaster recovery, search-and-rescue, etc. In most of the application scenarios, we can give a reasonable assumption that each node has been equipped with a GPS receiver already. So GMZRP does not need to pay extra overhead for the use of GPS devices.

5.4.1 Simulation Environment

We implement and simulate GMZRP in Glomosim [ZEN98]. We use 802.11 MAC protocol with DCF and a transmission range of 250m. Geographic forwarding adopts GPSR with activated perimeter mode. Nodes follow the Random Waypoint mobility model, where each node moves at a constant speed chosen randomly from a predefined speed range. The speed range is different for different simulation scenario. In each simulation run, we simulate the behavior of 100 nodes in a 1.2km X 1.2km square, which can be partitioned into 19 full zones as shown in Figure 5.1. The simulation time is 100 seconds. The multicast sources in our simulations generate constant bit rate (CBR) traffic, with each source originating 5 128 bytes packets per second. Here, we only consider multicasting with single source. Each simulation scenario is repeated for 10 times and the average results are obtained.

Referring to the setting of network flood frequency in ADMR [JET01], in our simulations, the time between two consecutive MRREQ broadcasts is set as follows. After 5 seconds since the initial MRREQ packet propagation, GMZRP broadcasts the MRREQ packet the second time. After 10 additional seconds, GMZRP broadcasts the MRREQ packet the third time, as is one broadcast after each subsequent 30 seconds.

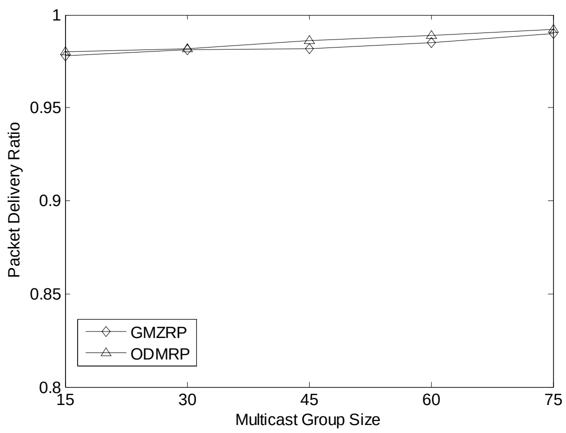
In these experiments, we assume two types of networks: quasi-static ad hoc networks and mobile ad hoc networks. In the mobility model followed by quasi-static ad hoc networks, the pause time is set to be 50 seconds and the node speed range is [$0ms^{-1}$, $5ms^{-1}$]. Quasi-static ad hoc networks simulate the scenario that nodes stay stationary or move slowly. In the mobility model followed by mobile ad hoc networks, the pause time is 0 seconds and the node speed range is set to be [$15ms^{-1}$, $20ms^{-1}$]. A pause time of 0 represents a network in which all nodes move continuously. We also vary the multicast group size in different experiments.

5.4.2 Simulation Results

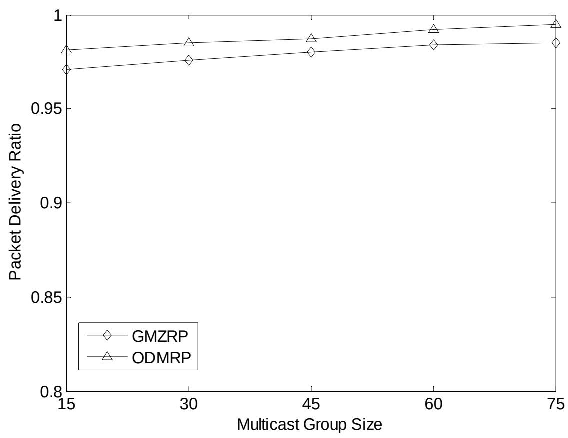
The commonly used performance metrics that we are also interested in are:

- *Packet delivery ratio*: The ratio of total number of packets received by all the receivers to the total number of packets originated by the source times the number of receivers.
- *Normalized packet overhead*: The total number of all control and data packets transmitted by any node in the network (either originated or forwarded), divided by the total number of data packets received across all multicast receivers. This metric represents the total packet overhead normalized by the total received packets.
- *Average path length*: The average number of hops traversed by each delivered data packet. This metric measures the performance of delivery latency.

Figure 5.5 shows the comparison results on packet delivery ratio under different multicast group sizes. In the quasi-static network, both GMZRP and ODMRP achieve the packet delivery ratio over 97.8% for all multicast group sizes and they have competing performance. In the mobile network, ODMRP outperforms GMZRP slightly by delivering within 1% of the multicast data packets due to redundant forwarding. However, ODMRP has contributed more than 3 times protocol overhead to this minor improvement compared to GMZRP. In addition, by comparing Figures 5.5(a) and 5.5(b), we can see that GMZRP works a little better in quasi-static network than in mobile network. It means that node mobility will affect the performance of GMZRP but very slightly.



(a)



(b)

Fig. 5.5: The comparison results in the packet delivery ratio between GMZRP and ODMRP in: (a) quasi-static networks, (b) mobile networks

In GMZRP, the primary control packet overhead comes from the propagation of MRREQ packets. With the increase in multicast group size, more receivers will reply a MRREP packet to the source. However, due to MRREP filtering at intermediate forwarding nodes, the increase in transmissions of MRREP packets is trivial. The number of other control packets is also negligible. Benefited from our strategy of MRREQ packet propagation, the number of MRREQ packet transmissions is independent of the multicast group size. Furthermore, the MRREQ packet propagation strategy is kind of broadcast technique instead of flooding. But in ODMRP, both control and data packets need to be flooded in the network periodically. Therefore, as shown in Figure 5.6, ODMRP generates much higher packet overhead than GMZRP.

In addition, with the increase in multicast group size, a larger fraction of the nodes have established forwarding state, and the density of forwarding nodes is higher. This will help GMZRP create a more efficient multicast tree, through which the number of packet transmissions shared by multiple receivers is increased. Hence, in GMZRP, the normalized packet overhead will be decreased as the multicast group size increases. A larger multicast group also helps create a natural redundancy which ODMRP exploits through the flood forwarding of the multicast data packets within the forwarding nodes. Since the packet overhead presented here is normalized to the total number of received packets, we can see that both protocols show lower overhead with more receivers, as shown in Figure 5.6.

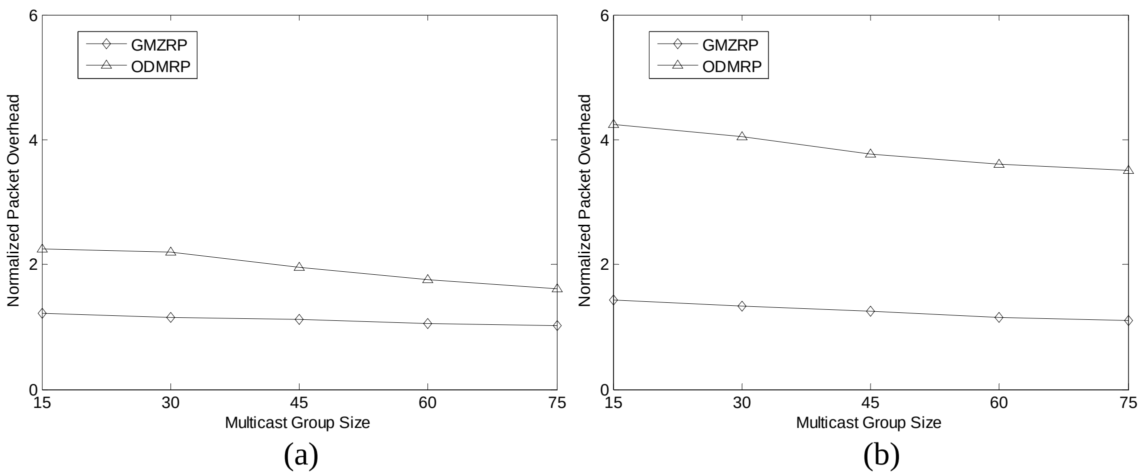


Fig. 5.6: The comparison results in the normalized packet overhead between GMZRP and ODMRP in: (a) quasi-static networks, (b) mobile networks

In addition, for both GMZRP and ODMRP, the overall overhead in lower mobility scenarios (quasi-static networks) is less than in the scenarios with high mobility (mobile networks). In GMZRP's case, this decrease is due to reduced broken links. In ODMRP's case, this is due to the creation of less redundant state.

We measure the average path length multicast packets have traversed. For a multicast forwarding structure (tree in GMZRP or mesh in ODMRP), we firstly count the path length to each receiver individually and then get the average result. Figure 5.7 shows that GMZRP has shorter average path length than ODMRP. The reason is that by the strategy of MRREQ packet propagation, GMZRP establishes a multicast tree, in which a path connecting the source to a receiver approximates the shortest path. In addition, as shown in Figure 5.7, the average path length in mobile networks is slightly higher than in quasi-static networks. It is because, due to broken links, a longer path will be recovered to replace the original shortest path.

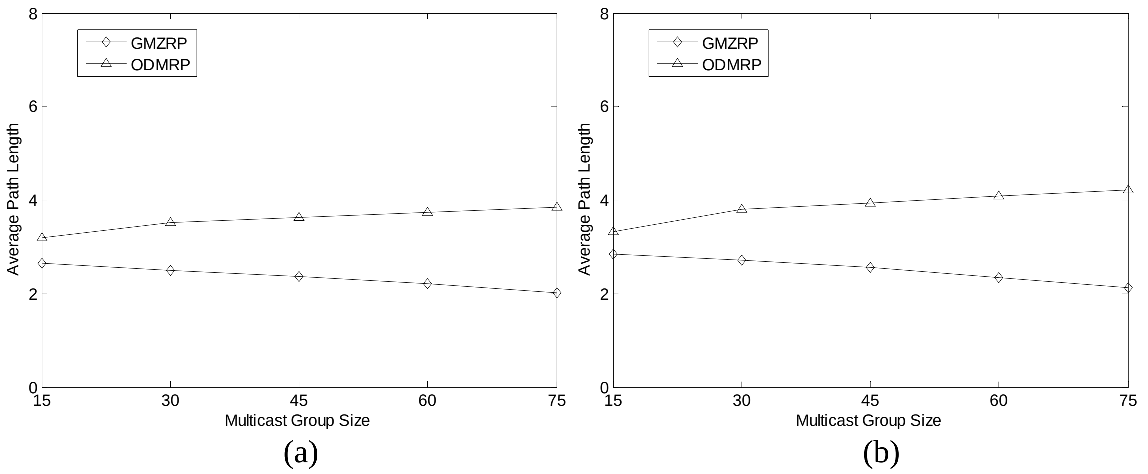


Fig. 5.7: The comparison results in the average path length between GMZRP and ODMRP in: (a) quasi-static networks, (b) mobile networks

5.5 Summary

In this chapter, we introduce the Geography-aided Multicast Zone Routing Protocol (GMZRP) for multicast routing in mobile ad hoc networks. It is kind of hybrid routing, combining the advantages of both topological routing and geographic routing. It uses an efficient MRREQ propagation strategy to discover the multicast forwarding tree with the lowest overhead and within the shortest time. In addition, each tree node also maintains a zone ID chain to each downstream receiver to benefit the recovery of broken links.

Simulation results show that GMZRP has decent performance in terms of packet delivery ratio, normalized packet overhead, and delivery latency.

Chapter 6

QoS-aware Multicast Tree Construction in the iMANET heterogeneous network

In this chapter, we describe the proposed two algorithms: *D*elay and *D*elay *V*ariation *M*ulticast *A*lgorithm (DDVMA) and the improved CBT+SP heuristic algorithm (CBT: *C*ore-*B*ased *T*ree, SP: *S*hortest *P*ath). Both of them are used to construct QoS-aware multicast trees in the iMANET heterogeneous network. The chapter is organized as follows. Firstly, a brief overview is given in Section 6.1. Section 6.2 describes the system model. Section 6.3 describes the benchmark algorithm *D*elay and *D*elay *V*ariation *C*onstraint *A*lgorithm (DDVCA). The detailed design, analysis and evaluation of DDVMA and the improved CBT+SP heuristic algorithm are presented in Section 6.3 and 6.4, respectively. Finally, Section 6.5 summarizes this chapter.

6.1 Overview

With the development of MANETs, more and more mobile nodes need to access the Internet. To benefit the integration of MANETs and the Internet, a new heterogeneous network architecture has been proposed [TSE03], which consists of several MANETs (e.g., working teams) attached to the backbone Internet through different gateway nodes. Figure 1.1 shows an iMANET heterogeneous network with five MANETs. In such a heterogeneous network, group communications occur when several team leaders need to work in a cooperative way. Furthermore, strict QoS requirements on delay and delay variation are required for some time-sensitive applications. In wireline network, such a problem is defined as Delay- and delay Variation-Bounded Multicast Tree (DVBMT) problem [GEO97] and proved to be NP-complete.

To solve the DVBMT problem in the iMANET heterogeneous network, we propose two efficient QoS multicast algorithms: DDVMA and the improved CBT+SP heuristic. Both of them take the wireless routing delay into account and can construct low delay variation multicast trees spanning all the gateway nodes involved in the group communications. Both of them are motivated from an existing heuristic, which is named as the CBT+SP heuristic [DIJ59, SHE02]. The idea of the CBT+SP heuristic is exactly employed in a well-known algorithm DDVCA [SHE02], which is known to be the best for the DVBMT problem. DDVMA improves the CBT+SP heuristic by further reducing the delay variation of the Shortest-Path Tree (SPT). The improved CBT+SP heuristic algorithm improves the original CBT+SP heuristic by eliminating its drawbacks. For these two algorithms, both theoretical analysis and simulation experiments were conducted, which show that they can really outperform DDVCA in terms of multicast delay variation under the same multicast end-to-end delay constraint.

6.2 System Model

As described in Section 1.2.2, we target at the following group communication scenarios in the iMANET heterogeneous network. Several MANETs (e.g., working team) attached to the backbone wired network need to coordinate their work, which involves one-to-many or multicast communication. Due to the limitation of radio transmission range of MHs, inter-MANET information transmission should go through the Internet. Each MANET will elect a team leader, which is responsible for information transmission between mobile nodes and the Internet gateway. In the multicast communication, one team needs to send packets to other teams. Without loss of generality, we assume only the team leader in each

MANET participates in the group communication. So there is a two-tier communication architecture: the lower tier is the communication between team leader and team members in the same team, the higher tier is the communication among team leaders and Internet gateways. The AODV routing protocol is used to discover routes between team leader and the Internet gateway.

A multicast message is sent from the source team leader. It is first forwarded to the source gateway $v_s \in V$, then arrives at a set of destination gateways $M \subseteq V - \{v_s\}$ through a multicast tree, and finally is forwarded to the destination team leaders by the destination gateways, separately. In the backbone network, a multicast routing tree *T* is used to transmit messages from the source gateway to each destination gateway in *M* concurrently. The mobile multicast tree for group communication in the heterogeneous network is constructed by concatenating *T* with the wireless routing paths between each team leader and its gateway. To guarantee the QoS of mobile group communication, the multicast end-to-end delay between the source team leader and each of the destination team leaders should not exceed the multicast end-to-end delay constraint Δ, and the multicast delay variation among the destination MHs should be minimized.

We construct the mobile multicast tree through two steps. First, AODV is used to find the routes between each team leader and its gateway and calculate the wireless routing delays. Second, using these delays, we construct the multicast tree *T* in the backbone network.

Let $P_T(v_s, v_w)$ denote the path from the source gateway v_s to a destination gateway $v_w \in M$ on *T*. Then the transmission delay between v_s and v_w on *T* is defined as $\sum_{l \in P_T(v_s, v_w)} D(l)$. For each gateway $g \in \{v_s\} \cup M$, we define a *gateway-delay* function $W : g \rightarrow r^+$. It assigns gateway *g* a nonnegative value *W*(*g*) representing the wireless routing delay between *g* and the team leader.

In this thesis, the problem is to determine an optimal multicast tree $T^*(V_{T^*}, E_{T^*})$, $\{v_s\} \cup M \subseteq V_{T^*}$, $E_{T^*} \subseteq E$, satisfying:

(1) $\Delta_{T^*} = W(v_s) + \max_{v_w \in M}\{ \sum_{l \in P_{T^*}(v_s, v_w)} D(l) + W(v_w)\} \leq \Delta$, and

(2) $\delta_{T^*} = \min_{T}\{ \max_{v_u, v_w \in M} \{ \left| (\sum_{l \in P_T(v_s, v_u)} D(l) + W(v_u)) - (\sum_{l \in P_T(v_s, v_w)} D(l) + W(v_w)) \right| \} \}$, where T denotes any multicast tree spanning v_s and M in $G(V, E)$.

If we assume $W(g)$=0 for each $g \in \{v_s\} \cup M$, our problem turns to be the DVBMT problem, which has been proved to be NP-complete [GEO97]. Our problem is also NP-complete because it contains, as a special case, the DVBMT problem. In this chapter, our work focuses on the DVBMT problem in the heterogeneous network, which consists of several MANETs attached to the backbone Internet. Recently, some works have been done on the DVBMT problem in optical network [XIN04]. Unfortunately, in [SHE99], it is proved that unless NP=P, it is impossible to find any ε_approximation algorithm for the DVBMT problem, where ε is a constant. Therefore, the most feasible solution now is to propose various heuristic algorithms and evaluate them by theoretical analysis and computer simulations.

In this chapter, we transform the problem of constructing a mobile multicast tree into the problem of constructing a fixed multicast tree in the backbone network. We let each gateway record the information about the wireless transmission path between it and the leader MH in its local MANET. Hence, when all the gateways have collected the information they need, heuristic multicast algorithms can be implemented in an IP routing protocol. This simplifies the problem and makes the implementation of the routing protocol easier.

6.3 Benchmark Algorithm

For the DVBMT problem, several heuristic algorithms have been proposed. Delay Variation Multicast Algorithm (DVMA) [GEO97] is a search algorithm which attempts to construct a multicast tree satisfying both the multicast end-to-end delay constraint and the multicast delay variation constraint. If the algorithm fails to discover such a feasible tree, it will return the tree with the smallest value of multicast delay variation among the trees considered. Although DVMA demonstrates good average case behavior in terms of the multicast delay variation, its time complexity is very high. DDVCA is an efficient algorithm, which is meant to search as much as possible for a multicast tree with a small multicast delay variation under the multicast end-to-end delay constraint. DDVCA claims to outperform DVMA slightly in the multicast delay variation. However, the time complexity of DDVCA is much lower than DVMA.

The fundamental strategy of DDVCA comes from CBT's Core Router concept and the minimum delay path algorithm [DIJ59] and we name it as the CBT+SP heuristic. Since DDVMA and the improved CBT+SP heuristic are both motivated by the original CBT+SP heuristic, we regard DDVCA as the benchmark algorithm. The basic idea in DDVCA is described as follows. For each network node, construct the SPT from it to all the destination nodes. The node whose SPT has the minimum multicast delay variation is selected as the central node. Then examine whether the addition of the minimum delay between the source node and the current central node and the maximum multicast end-to-end delay of the SPT rooted at the central node satisfies the multicast end-to-end delay constraint. If the central node violates the constraint, it will be abandoned. In this case, the algorithm will go on to pick the node whose SPT has the next minimum multicast delay variation as the next possible central node and apply the same checking process until a central node which satisfies the constraint is found.

6.4 DDVMA (Delay and Delay Variation Multicast Algorithm)

DDVMA can find a multicast tree satisfying the multicast end-to-end delay constraint and minimizing the multicast delay variation. Two concepts- the proprietary second shortest path and partially proprietary second shortest path are introduced, which can help DDVMA achieve better performance in terms of the multicast delay variation than DDVCA. The strategy employed by DDVMA is also applicable to handling the mobility of MHs in the iMANET heterogeneous network.

6.4.1 Notations and Definitions

In DDVCA, the minimum delay path algorithm and the SPT is used. A SPT is constructed by combining all the minimum delay paths from the source node to each destination node. The multicast delay variation of the SPT is the difference between the maximum and the minimum multicast end-to-end delays on the tree. In DDVCA, each node is checked to see whether it can be selected as the central node in increasing order of the multicast delay variation of the SPT rooted at it.

SPT has very good performance in terms of the multicast end-to-end delay. However, selecting the shortest (i.e., minimum delay) paths may lead to a violation of the delay variation constraint among nodes that are close to the source and nodes that are far away from it. Consequently, it may be necessary to replace the shortest paths by longer paths for some destination nodes in order to further reduce the multicast delay variation of the SPT. So if we introduce higher delay paths to replace the minimum delay paths from the source to some destinations on the SPT, intuitively more trees with small multicast delay variation can be searched compared to DDVCA.

For ease of description, we denote the central node being checked as v_c, denote the SPT rooted at v_c as $T(v_c)$, denote the minimum delay path between v_i and v_j as $P(v_i, v_j)$, denote the delay on path *P* as *Delay*(*P*).

The maximum multicast end-to-end delay of $T(v_c)$ should not exceed $\Delta' = \Delta - Delay(P(v_s, v_c))$, i.e., the difference between the multicast end-to-end delay constraint and the minimum delay between the source node and the current central node. Δ' stands for the given upper bound of all multicast end-to-end delays of the SPT.

Each step the replacement operation should conform to three criterions:

(1) The new path does not interfere with other multicast paths on the tree;

(2) The multicast delay variation of the multicast tree should not be increased;

(3) The end-to-end delay of the new path should not exceed Δ'.

In (2), the multicast delay variation is required to be not increased. It occurs very few that the multicast delay variation remains unchanged after the replacement. If it occurs, we still accept the new SPT to continue the search although the multicast delay variation has not been reduced. This strategy will make more potential multicast trees be searched.

We define the concepts of proprietary second shortest path and partially proprietary second shortest path. Both of them are generated according to different methods. The proprietary second shortest path and partially proprietary second shortest path will be used as the higher delay path.

For one destination node v_j, we define:

Proprietary links: links which are not shared by other destination nodes on $T(v_c)$.

Proprietary link set (*PS*): all the proprietary links of v_j.

Proprietary second shortest path: the second shortest path from v_c to v_j, which is obtained by computing the shortest path from v_c to v_j after deleting *l* from the network topology *G*, $l \in PS$. So the number of proprietary second shortest paths equals to the number of proprietary links for v_j. The proprietary second shortest path is actually the shortest path on the network topology $G-\{l\}$.

In Figure 6.1, assume Vc is the central node, V2, V4, V5, V6 are destination nodes. For V6, its proprietary links are (V2, V3) and (V3, V6). So its proprietary link set is {(V2, V3), (V3, V6)}.

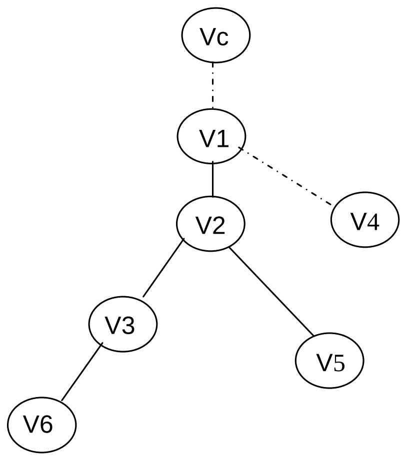


Fig. 6.1: Illustration of proprietary links and partially proprietary links

Partially proprietary links: links which are only shared by all its child destination nodes on $T(v_c)$.

Partially proprietary link set (*PPS*): all the partially proprietary links of v_j.

Partially proprietary second shortest path: the second shortest path from v_c to v_j or a child destination node of v_j, which is obtained by computing the shortest path from v_c to v_j or the child destination node of v_j after deleting *l* from the network topology *G*, $l \in PPS$. The partially proprietary second shortest path is actually the shortest path on the network topology $G-\{l\}$.

Also in Figure 6.1, for V2, its partial proprietary link is (V1, V2). So its partially proprietary link set is {(V1, V2)}.

For a multicast tree, it is easy to determine the proprietary link set or partially proprietary link set for a destination node. We can compute the proprietary second shortest paths or partially proprietary second shortest paths for a destination node using Dijkstra's algorithm conveniently and quickly.

The characteristics of proprietary second shortest paths and partially proprietary second shortest paths guarantee that adding them to the SPT will not create a cycle, which will be proved in Section 6.4.3. Thus other multicast paths on the SPT will not be interfered with.

6.4.2 Design of the Algorithm

DDVMA constructs a QoS multicast tree in the backbone network for mobile group communication in the heterogeneous network. The multicast tree is used to transmit multicast packets from the source gateway to all the destination gateways. The wireless transmission paths between each leader MH and its gateway are determined and optimized by MANET routing protocol. When DDVMA is executed, it is required that all the wireless transmission paths and their delays are known. In Section 6.4.5, we will discuss the mobility of the leader MHs and propose strategy to handle the effect of the changed wireless transmission paths on the multicast tree.

In DDVCA, for each node, the multicast delay variation of the SPT rooted at it is used to judge whether it can be selected as the central node. For example, there exist two SPTs rooted at node *A* and node *B*, respectively. For node *A* the multicast delay variations of its

SPT is 5, and for node B it is 4. Then DDVCA will select node *B* as the central node. However, in DDVMA we will check whether these two SPTs can be improved using higher delay paths to decrease their multicast delay variations. Assume the multicast delay variation of the SPT rooted at *A* can be further decreased to 3, and the SPT rooted at *B* cannot be improved. Thus DDVMA will select node *A* as the central node and find the multicast tree with smaller multicast delay variation than that of DDVCA. The improvement is realized by using the proprietary second shortest path or partially proprietary second shortest path to replace the multicast path with the minimum end-to-end delay on the SPT.

The improvement procedure can be seen as an optimization procedure, i.e., using a better path to optimize the QoS of the SPT. The optimization objective is to achieve smaller multicast delay variation under multicast end-to-end delay constraint.

The optimization procedure will terminate when one of the following two cases occurs:

(1) The multicast delay variation is decreased to a specified tolerance range or can not be decreased further, whichever occurs first.

(2) The maximum multicast end-to-end delay exceeds the given upper bound on the maximum multicast end-to-end delay of the SPT.

During the optimization procedure, the tree should always keep to be a SPT structure for the associated network topology. At the beginning, the associated network topology is just the network topology *G*. After each replacement, the associated network topology will exclude the selected proprietary link or partially proprietary link from itself.

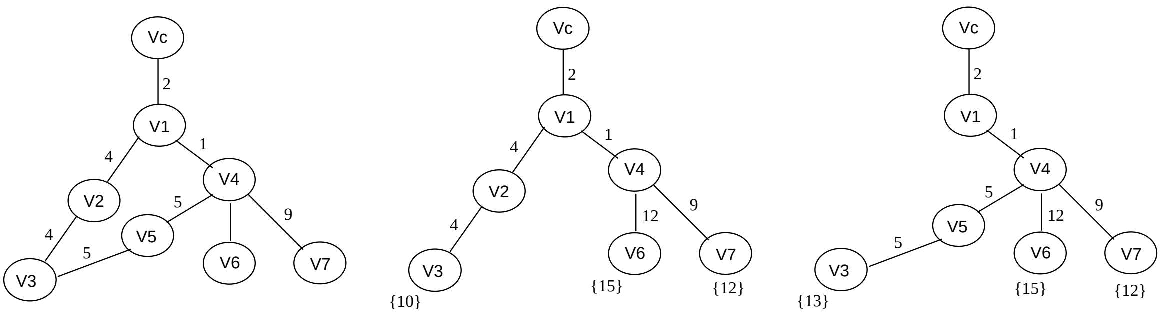
For one destination node on the SPT, if its proprietary link set is not NULL, its partially proprietary link set will be NULL, and vice versa. Assume we are dealing with the destination node with the minimum multicast end-to-end delay on the SPT, if its proprietary link set is not NULL which means it is a leaf node, we will check whether a proprietary second shortest path can be found to optimize the tree; if its partially proprietary link set is not NULL which means it is a non-leaf node, we will check whether partially proprietary second shortest paths can be found to optimize the tree. Two subalgorithms are proposed, one is to deal with the destination node with at least one proprietary link and the other is to deal with the destination node with at least one partially proprietary link. The former subalgorithm is named as algorithm P, the latter algorithm PP.

1) Algorithm P

An efficient algorithm for improving the multicast delay variation of an SPT using the proprietary second shortest path is listed in algorithm P. It starts out with an SPT and decreases the multicast delay variation by replacing the minimum delay multicast path with the appropriate proprietary second shortest path. If algorithm P returns False, it means the SPT remains unchanged.

For a SPT $T(v_c)$ rooted at v_c, v_j represents the destination node with the minimum multicast end-to-end delay on it. If v_j is a leaf node, obviously it has at least one proprietary link, i.e., $PS(v_j) \neq \phi$. Then algorithm P can be executed to improve the SPT. First we denote the second minimum multicast end-to-end delay of $T(v_c)$ as $sec\_min\_delay(T(v_c))$. For each proprietary link $l \in PS(v_j)$, the corresponding proprietary second shortest path $P'(v_c, v_j)$ will be computed. If the delay of one such path conforms to the following two conditions: (1) does not exceed Δ', the given upper bound on the maximum multicast end-to-end delay of $T(v_c)$; and (2) the difference between it and $sec\_min\_delay(T(v_c))$ does not exceed δ, this path can be used to replace the minimum delay path from v_c to v_j with the multicast delay variation of the SPT decreased. Thus the SPT will be improved with the QoS optimized.

The proprietary second shortest path will keep to be the shortest path and the improved SPT will keep to be the shortest path tree on the network topology $G' - \{l\}$. After $T(v_c)$ is modified, the network graph G' associated with it needs to exclude the selected proprietary link l (i.e., $G' = G' - \{l\}$) in order to keep the shortest path tree structure of $T(v_c)$. This is to say that the associated network topology will be updated with the modification of the SPT in algorithm P.



(a) (b) (c)

Fig. 6.2: An illustration of algorithm P: (a) a given network topology, (b) the SPT from Vc to V3, V6, and V7 with the multicast delay variation 5, (c) the improved SPT with the multicast delay variation 3

In the following, we illustrate algorithm P with an example. Figure 6.2(a) shows a given network topology G', where the number associated with every link represents the transmission delay on this link. The destination nodes are V3, V6, and V7. Figure 6.2(b) shows the SPT from Vc to all the destination nodes. The number in the bracket near each destination node represents the multicast end-to-end delay of the destination node. Clearly, the multicast delay variation of the SPT is 5 and V3 has the smallest multicast end-to-end delay. The proprietary link set of V3 is {(V1, V2), (V2, V3)}. We let $v_j = \text{V3}$ and run algorithm P to improve the SPT. The improved tree is showed in Figure 6.2(c) with the multicast delay variation decreased to 3. The selected proprietary link can be (V1, V2) or (V2, V3). It depends on which one is selected first.

Algorithm P The proprietary second shortest path algorithm

Input the SPT $T(v_c)$ from v_c to all destination nodes in *M*, the upper bound Δ' on the maximum multicast end-to-end delay of $T(v_c)$, the network topology graph G' associated with $T(v_c)$, the multicast delay variation δ of $T(v_c)$, a destination node v_j.

Output False or True with an improved SPT $T(v_c)$.

Step 0 Initialization

Let $sec\_min\_delay(T(v_c))$ be the second minimum multicast end-to-end delay of $T(v_c)$.

Step 1 Modify $T(v_c)$ to improve the multicast delay variation

while $PS(v_j) \neq \phi$ do begin

select $l \in PS(v_j)$ and compute the corresponding proprietary second shortest path $P'(v_c, v_j)$ from v_c to v_j.

if $Delay(P'(v_c, v_j)) \leq \min\{\Delta', \delta + sec\_min\_delay(T(v_c))\}$ then

$T(v_c) = T(v_c) - PS(v_j)$;

$T(v_c) = T(v_c) \cup P'(v_c, v_j)$;

$G' = G' - \{l\}$

return True. else

$PS(v_j) = PS(v_j) - \{l\}$.

end-if

end-while

return False.

2) Algorithm PP

Different from algorithm P, the partially proprietary second shortest paths are used in algorithm PP. If non-leaf node v_j is the destination node with the minimum multicast end-to-end delay on $T(v_c)$, some child nodes of v_j will also be destination nodes. Node v_j with all its child destination nodes forms a subset M' of *M*. $P(v_c, v_j)$ represents the multicast path from v_c to v_j on $T(v_c)$, and $T(v_j)$ represents the sub-SPT rooted at node v_j on $T(v_c)$. $P(v_c, v_j)$ is the common part of each multicast path $P(v_c, j)$ associated with node *j*, $j \in M' - \{v_j\}$. For each $j \in M' - \{v_j\}$, $P(v_c, j)$ will also be changed when $P(v_c, v_j)$ is replaced by the corresponding partially proprietary second shortest path $P'(v_c, v_j)$. We propose two strategies for the changes of $P(v_c, j)$ ($j \in M' - \{v_j\}$) caused by the change of $P(v_c, v_j)$: (1) compute the partially proprietary second shortest path $P'(v_c, j)$ as the new multicast path from v_c to *j* for each $j \in M' - \{v_j\}$; (2) use the corresponding path on $P'(v_c, v_j) \cup T(v_j)$ (i.e., the topology combining the partially proprietary second shortest path $P'(v_c, v_j)$ with the sub-SPT $T(v_j)$) as the new multicast path from v_c to *j* for each $j \in M' - \{v_j\}$. Algorithm PP adopts the former strategy because it can help improve the multicast delay variation between v_j and any other node in M'. We will prove the improvement of the former strategy on the multicast delay variation in Section 6.4.3. If algorithm PP returns False, it means that the SPT remains unchanged.

For non-leaf destination node v_j with the minimum multicast end-to-end delay on $T(v_c)$, obviously it has at least one partially proprietary link, i.e., $PPS(v_j) \neq \phi$. Then algorithm PP can be executed to improve the SPT. For each link $l \in PPS(v_j)$, compute the corresponding partially proprietary second shortest paths from v_c to each node in M', i.e., $P'(v_c, j)$ ($j \in M'$). Let *MinDelay* denote the minimum path delay among all the partially proprietary second shortest paths and the multicast paths from v_c to each destination node in $M - M'$. Among all the partially proprietary second shortest paths, find the maximum path delay. If the maximum path delay satisfies two conditions: (1) does not exceed the given upper bound Δ' of multicast end-to-end delay of $T(v_c)$; and (2) the difference between it and *MinDelay* does not exceed δ, these partially proprietary second shortest paths can be used to replace the corresponding multicast paths on $T(v_c)$. The replacement will decrease the multicast delay variation of the SPT. Algorithm PP terminates when all partially proprietary links are checked, or for some partially proprietary link, the corresponding partially proprietary second shortest paths satisfy the above conditions, whichever occurs first.

Similar to algorithm P, in order to keep the shortest path tree structure of $T(v_c)$, the associated network topology will be updated with the modification of the SPT in algorithm PP, i.e., $G' = G' - \{l\}$, assuming *l* is the selected partially proprietary link.

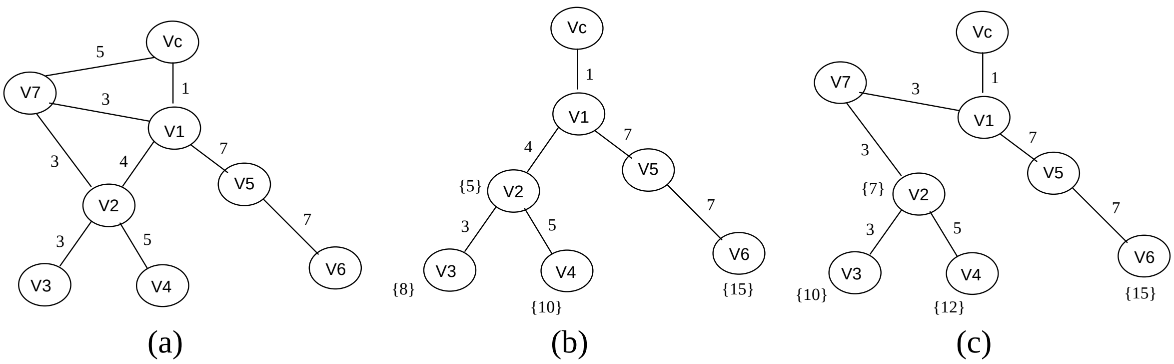


Fig. 6.3: An illustration of algorithm PP: (a) a given network topology, (b) the SPT from Vc to V2, V3, V4, and V6 with the multicast delay variation 10, (c) the improved SPT with the multicast delay variation 8

In the following, we illustrate algorithm PP with an example. Figure 6.3(a) shows the given network topology G'. The destination nodes are V2, V3, V4, and V6. Figure 6.3(b) shows the SPT from Vc to all the destination nodes. We can see the multicast delay variation of the SPT is 10 (i.e., 15-5), and V2 has the least multicast end-to-end delay. The partially proprietary link set of V2 is {(V1, V2)}. We let $v_j = \text{V2}$ and run algorithm PP to improve the SPT. The improved tree is shown in Figure 6.3(c) with the multicast delay variation decreased to 8 (i.e., 15-7). The selected partially proprietary link is (V1, V2).

Algorithm PP The partially proprietary second shortest path algorithm

Input the SPT $T(v_c)$ from v_c to all destination nodes in *M*, the upper bound Δ' on the maximum multicast end-to-end delay of $T(v_c)$, the network topology graph G' associated with $T(v_c)$, the multicast delay variation δ of $T(v_c)$, a destination node v_j.

Output False or True with the improved SPT $T(v_c)$.

Step 0 Initialization

Let $j_1, j_2, \cdots j_t$ represent the child destination nodes of v_j on $T(v_c)$, $T(v_j)$ represent the sub-SPT rooted at v_j on $T(v_c)$, and $M' = \{v_j, j_1, j_2, \cdots j_t\}$.

Step 1 Modify $T(v_c)$ to improve the multicast delay variation

while $PPS(v_j) \neq \phi$ do begin

Select $l \in PPS(v_j)$ and compute the corresponding partially proprietary second shortest path from v_c to each node in M', i.e.,

$P'(v_c, v_j)$, $P'(v_c, j_1)$, $P'(v_c, j_2)$, $\cdots$, $P'(v_c, j_t)$, respectively.

$MinDelay = \min\{\min_{v \in M - M'}\{Delay(P(v_c, v))\}, \min_{j \in M'}\{Delay(P'(v_c, j))\}\}$;

if $\{\max_{j \in M'}\{Delay(P'(v_c, j))\} \leq \min\{\Delta', MinDelay + \delta\}\}$ then

$T(v_c) = T(v_c) - PPS(v_j) - T(v_j)$;

$T(v_c) = T(v_c) \cup \bigcup_{j \in M'} P'(v_c, j)$;

$G' = G' - \{l\}$;

return True. else

$PPS(v_j) = PPS(v_j) - \{l\}$.

end-if

end-while

return False.

3) Formal Description of DDVMA

In the following, we present a formal description of DDVMA. Algorithm P and PP are integrated into it. Since DDVMA is designed for QoS mobile group communication scenario, the wireless transmission paths between each gateway and the leader MH in its local MANET should be considered. The multicast end-to-end delay is the delay from the leader MH in the source MANET to each leader MH in the destination MANETs. The multicast delay variation also means the delay variation among all the leader MHs in the destination MANETs. DDVMA is to construct a QoS multicast tree spanning the source gateway and all the destination gateways in the backbone network. A QoS mobile multicast tree will be generated by combining the multicast tree with the corresponding wireless transmission paths. The QoS mobile multicast tree should conform to the multicast end-to-end delay constraint and achieve the smallest multicast delay variation. So when constructing the multicast tree in the backbone network, the effects of the wireless transmission paths must be considered.

Each gateway participating in the group communication collects the information about the wireless transmission path between it and the leader MH in its local MANET. The wireless transmission delay is also used to compute the multicast end-to-end delay and the multicast delay variation in our algorithm. If a path ends at a destination gateway, the wireless transmission path delay recorded at the destination gateway needs to be added to the path delay. Thus if the constructed multicast tree in the backbone network by DDVMA can satisfy the QoS constraint, so can the mobile multicast tree.

In the beginning, DDVMA sets the necessary variables. Variable dv_{min} records the minimum multicast delay variation, T records the present constructed multicast tree, and v_c records the central node being checked. In DDVMA, when an SPT rooted at node v_c is constructed, we will try to decrease the multicast delay variation of the SPT further. It is realized by running algorithm P or PP. For variable *flag*, True means the SPT will be further considered for improvement, False just means the SPT cannot be further improved. If v_c is a destination gateway, it will be selected as the destination gateway with the minimum multicast end-to-end delay on the SPT rooted at v_c, which equals to 0. However, this will lead to that $PS(v_c) = \phi$ and $PPS(v_c) = \phi$, algorithm P and PP will not be executed.

The primary computation overhead in DDVCA comes from the calculation of the shortest paths, whose time complexity is $O(n^2)$. The primary computation overhead in DDVMA comes from the calculation of proprietary second shortest paths and partially proprietary second shortest paths. Their time complexity is also $O(n^2)$. The number of calculations of proprietary second shortest paths and partially proprietary second shortest paths in DDVMA is proportional to the number of destinations. Therefore, the time complexity of DDVMA is at the same level of DDVCA although the actual execution time of DDVMA is a bit longer.

Algorithm DDVMA

Input the network topology graph *G*, a set of destination gateways *M*, a source gateway v_s, an upper bound Δ on the maximum multicast end-to-end delay.

Output a multicast tree *T* satisfying the multicast end-to-end delay constraint and having a small multicast delay variation.

Step 0 Initialization

Let $dv_{min} = \infty$, $T = \phi$, $v_c = \phi$, and *flag*=True.

Step 1 for each $v_c \in V$ do begin

Let $P(v_s, v_c)$ represent the minimum delay path from v_s to v_c, $Delay(P)$ represent the delay on it, $\Delta' = \Delta - Delay(P(v_s, v_c))$, $G' = G$.

Compute the SPT $T(v_c)$ from v_c to all destination gateways in *M*.

while $flag == True$ do begin

Let v_j be the destination gateway with the minimum multicast path delay on $T(v_c)$ and δ be the multicast delay variation of $T(v_c)$.

if $PS(v_j) \neq \phi$ then *flag*=(run procedure P); else

if $PPS(v_j) \neq \phi$ then *flag*=(run procedure PP); else

flag=False;

end-if

end-if

end-while

Let $dv(v_c)$, $max\_delay(T(v_c))$ be the multicast delay variation, the maximum multicast end-to-end delay of $T(v_c)$, respectively.

if $\{dv_{min} > dv(v_c)\}$ and $\{Delay(P(v_s, v_c)) + max\_delay(T(v_c)) \le \Delta - W(v_s)\}$ then

$dv_{min} = dv(v_c)$;

$T = T(v_c) \cup P(v_s, v_c)$;

end-if

end-for

Step 2 if $\{T = \phi\}$ then print "no such multicast tree exists!"; else

return *T*;

end-if

In the following, we illustrate the operation of DDVMA with an example. We will compare it with DDVCA. The given network topology is shown in Figure 6.4. For a group communication scenario, we denote Vs as the source gateway, V2, V5 and V9 as the destination gateways, i.e., M={V2, V5, V9}. The number in the parentheses near gateway g (including the source gateway and all the destination gateways) represents the corresponding wireless transmission path delay $W(g)$. Suppose the multicast end-to-end delay constraint Δ is 60 for the QoS mobile multicast tree. Because the wireless transmission path delay between the source MH and the source gateway is 1, the multicast end-to-end delay constraint used in DDVMA will be 59 (i.e., 60-1). Table 6.1 shows the procedure of selecting a central node in DDVCA. Table 6.2 shows the corresponding procedure in DDVMA.

In Table 6.1, for each network node Vi, the minimum path delay between it and each destination gateway (i.e., the wireline transmission path delay in the backbone network, plus the corresponding wireless transmission path delay recorded at the destination gateway) is listed in each column. Then dv(Vi), the multicast delay variation of the SPT rooted at Vi, is computed and listed in the bottom line of each column. From Table 6.1, we know the multicast delay variation of the SPT rooted at Vs, V7 and V8 are all the minimum. Assume Vs is selected, we get the multicast tree which is illustrated in Figure 6.5(a). The multicast tree satisfies the multicast end-to-end delay constraint.

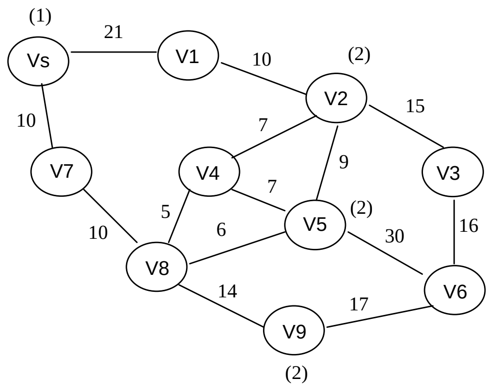


Fig. 6.4: A given network topology $G=(V, E)$

In Table 6.2, for each network node Vi, we also list the path delay between it and each destination gateway on the improved SPT rooted at Vi in each column. A * next to a delay value indicates that it is the delay of proprietary second shortest path or partially proprietary second shortest path. It means that the corresponding minimum delay paths on the SPT have been replaced by the proprietary second shortest paths or partially proprietary second shortest paths. Then *dv*(Vi), the multicast delay variation of the improved SPT, is computed and listed in the bottom line of each column. From Table 6.2, we know the multicast delay variation of the improved SPT rooted at V7 and V8 are both the minimum. Assume V7 is selected, we get the multicast tree illustrated in Figure 6.5(b). The multicast tree also satisfies the multicast end-to-end delay constraint.

From Figure 6.5, we can see DDVMA achieves the multicast tree with smaller multicast delay variation than that of DDVCA.

Table 6.1: The manner by which DDVCA selects a central node

Network node Vi The minimum delay	Destination node Vj	Vs	V1	V2	V3	V4	V5	V6	V7	V8	V9
The multicast path	V2	33	12	2	17	9	11	33	24	14	28
delay between Vi	V5	28	21	11	26	9	2	32	18	8	22
and Vj	V9	36	38	28	35	21	22	19	26	16	2
The maximum delay from Vi to each Vj		36	38	28	35	21	22	33	26	16	28
The minimum delay from Vi to each Vj		28	12	2	17	9	2	19	18	8	2
dv(Vi)		8	26	26	18	12	20	14	8	8	26

Table 6.2: The manner by which DDVMA selects a central node

Network node Vi The minimum delay	Destination node Vj	Vs	V1	V2	V3	V4	V5	V6	V7	V8	V9
The multicast path	V2	33	12	2	17	22*	11	33	24	14	28
delay between Vi	V5	34*	21	11	26	13*	2	32	24*	14*	22
and Vj	V9	36	38	28	35	21	22	19	26	16	2
The maximum delay from Vi to each Vj		36	38	28	35	22	22	33	26	16	28
The minimum delay from Vi to each Vj		33	12	2	17	13	2	19	24	14	2
dv(Vi)		3	26	26	18	9	20	14	2	2	26

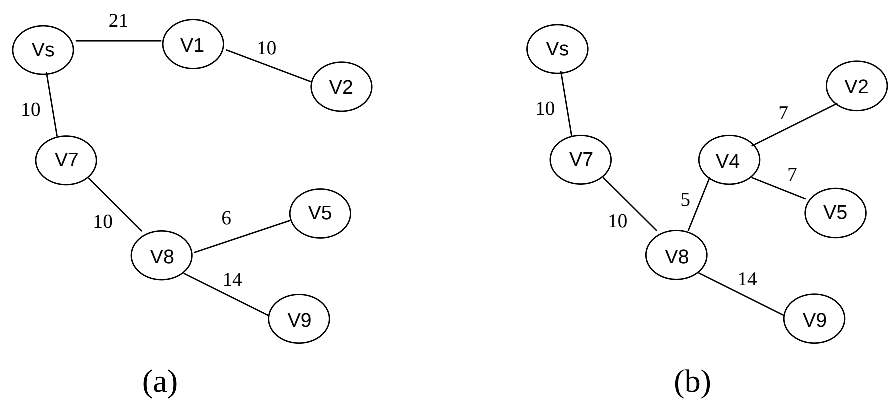


Fig. 6.5: Two multicast trees: (a) achieved by DDVCA with the multicast delay variation equal to 8, (b) achieved by DDVMA with the multicast delay variation equal to 2

6.4.3 Theoretical Analysis

In this section, theoretical analysis is made on DDVMA. We will prove its correctness. Also we will prove that DDVMA can achieve better efficiency in terms of the multicast delay variation than DDVCA.

Lemma 6.1: Let $T(v_c)$ be the SPT rooted at v_c. For any destination node $v_j \in M$ with $PS(v_j) \neq \phi$, use a proprietary second shortest path to replace the shortest path (i.e., the minimum delay path) will not create a cycle on $T(v_c)$.

Proof: Let $T^*(v_c)$ represent the SPT from v_c to all nodes in $M-\{v_j\}$. $T^*(v_c)$ is one part of $T(v_c)$. All links in $PS(v_j)$ are not shared by any destination node in $M-\{v_j\}$ on $T(v_c)$ according to the definition in Section 6.4.1. So no link in $PS(v_j)$ belongs to $T^*(v_c)$. The proprietary second shortest path is obtained by computing the shortest path after deleting the selected proprietary link from the associated network topology. All the shortest paths on $T^*(v_c)$ will still be the shortest paths for the updated network topology. Hence $T^*(v_c)$ is still one part of the improved SPT. The improved SPT will be constructed by combining the proprietary second shortest path with $T^*(v_c)$. If the replacement creates a cycle, the proprietary second shortest path must pass at least one node belonging to $T^*(v_c)$ except v_c. But for the proprietary second shortest path, the subpath from v_c to any node belonging to $T^*(v_c)$ is still the shortest path which coincides with the corresponding path on $T^*(v_c)$. It contradicts with the assumption of a cycle being created. Hence the replacement of a proprietary second shortest path will not create a cycle. ■

Lemma 6.2: Let $T(v_c)$ be the SPT rooted at v_c. For any destination node $v_j \in M$ with $PPS(v_j) \neq \phi$, let $j_1, j_2, \cdots j_t$ represent the child destination nodes of v_j on $T(v_c)$ and $M' = \{v_j, j_1, j_2, \cdots j_t\}$. Use partially proprietary second shortest paths to replace the corresponding shortest paths from v_c to each node in M' will not create a cycle on $T(v_c)$.

Proof: It is similar to *Lemma 6.1*. Let $T^*(v_c)$ represent the SPT from v_c to all nodes in $M - M'$. All links in $PPS(v_j)$ are not shared by any destination node in $M - M'$ on $T(v_c)$ according to the definition in Section 6.4.1. So no link in $PPS(v_j)$ belongs to $T^*(v_c)$. The updated network topology associated with the improved SPT is obtained by deleting the selected partially proprietary link from the network topology associated with $T(v_c)$. For the updated network topology, all the shortest paths on $T^*(v_c)$ will still be the shortest paths, and the partially proprietary second shortest paths from v_c to each node in M' are also the shortest paths. So combining $T^*(v_c)$ with all the partially proprietary second shortest paths will form the improved SPT and no cycle is created. ■

Theorem 6.1: Correctness of algorithm P: Algorithm P keeps the SPT unchanged or returns an improved SPT. The improved SPT satisfies the multicast end-to-end delay constraint and does not increase the multicast delay variation.

Proof: In algorithm P, the replacement of a proprietary second shortest path occurs or not. From the algorithm description, we can see if $Delay(P'(v_c, v_j)) \le \min\{\Delta', \delta + sec\_min\_delay(T(v_c))\}$, the replacement will occur. We need to prove the expression can guarantee that the new SPT will satisfy the multicast end-to-end delay constraint and does not increase the multicast delay variation.

From the above expression, we get $Delay(P'(v_c, v_j)) \le \Delta'$. It means the delay of the proprietary second shortest path satisfies the multicast end-to-end delay constraint. All the other multicast paths remain unchanged after the replacement. So the new SPT will satisfy the multicast end-to-end delay constraint.

From the definition of the proprietary second shortest path, we know $Delay(P'(v_c, v_j))$ is not less than the minimum multicast end-to-end delay of the old SPT. If it is less than or equal to the maximum multicast end-to-end delay of the old SPT, the multicast delay variation of the new SPT will not be increased obviously. If it is greater than the maximum multicast end-to-end delay of the old SPT, it will become the maximum multicast end-to-end delay of the new SPT and $sec\_min\_delay(T(v_c))$ also becomes the minimum multicast end-to-end delay of the new SPT. From the above expression, we get $Delay(P'(v_c, v_j)) \le \delta + sec\_min\_delay(T(v_c))$, i.e., $Delay(P'(v_c, v_j)) - sec\_min\_delay(T(v_c)) \le \delta$. Hence the multicast delay variation of the new SPT will not be greater than the old SPT. So the new SPT does not increase the multicast delay variation. ■

Lemma 6.3: Let v_j be a non-leaf destination node with the minimum multicast end-to-end delay on $T(v_c)$ (the SPT rooted at v_c) and $PPS(v_j) \ne \phi$. Let $j_1, j_2, \cdots j_t$ represent the child destination nodes of v_j on the SPT and $M' = \{v_j, j_1, j_2, \cdots j_t\}$. The former one of the two strategies mentioned in Section 6.4.2- 2 can improve the multicast delay variation between v_j and any other node in M'.

Proof: If the former one of the two strategies mentioned in Section 6.4.2- 2 is adopted, algorithm PP will compute the partially proprietary second shortest path $P'(v_c, j)$ for each $j \in M'$ and use them to replace the corresponding shortest paths on $T(v_c)$.

Let $P'(v_c, v_j)$ represent the partially proprietary second shortest path from v_c to v_j. For each $j \in M' - \{v_j\}$, let $P(v_j, j)$ represent the shortest path between v_j and j, $P'(v_c, j)$ represent the partially proprietary second shortest path between v_c and j. For the updated network topology (i.e., deleting the selected partially proprietary link from the network topology associated with the old SPT), $P'(v_c, v_j)$, $P'(v_c, j)$ and $P(v_j, j)$ are all the shortest paths. So we have: $Delay(P'(v_c, v_j)) \le Delay(P'(v_c, j)) + Delay(P(v_j, j))$ and $Delay(P'(v_c, v_j)) + Delay(P(v_j, j)) \ge Delay(P'(v_c, j))$. From the two expressions, we get $\left| Delay(P'(v_c, j)) - Delay(P'(v_c, v_j)) \right| \le Delay(P(v_j, j))$. Because $Delay(P(v_j, j)) \le \delta$, we get $\left| Delay(P'(v_c, j)) - Delay(P'(v_c, v_j)) \right| \le \delta$.

We can see this strategy can help improve the multicast delay variation between v_j and each $j \in M' - \{v_j\}$. ■

Theorem 6.2: Correctness of algorithm PP: Algorithm PP keeps the SPT unchanged or returns an improved SPT. The improved SPT satisfies the multicast end-to-end delay constraint and does not increase the multicast delay variation.

Proof: In algorithm PP, the replacement of the partially proprietary second shortest paths occurs or not. From the algorithm description, we can see if $\max_{j \in M'} \{Delay(P'(v_c, j))\} \le \min\{\Delta', MinDelay + \delta\}$, the replacement will occur. Similar to

Theorem 6.1, we need to prove the expression can guarantee that the new SPT will satisfy the multicast end-to-end delay constraint and does not increase the multicast delay variation.

From the above expression, we get $\max_{j \in M'} \{Delay(P'(v_c, j))\} \le \Delta'$. It means all the partially proprietary second shortest paths satisfy the multicast end-to-end delay constraint. The multicast path from v_c to each node in $M - M'$ will remain unchanged after the replacement. So the new SPT will satisfy the multicast end-to-end delay constraint.

As is known that *MinDelay* is the minimum multicast end-to-end delay of the new SPT. From the definition of the partially proprietary second shortest path, we know *MinDelay* is not less than the minimum multicast end-to-end delay of the old SPT. If $\max_{j \in M'} \{Delay(P'(v_c, j))\}$ is less than or equal to the maximum multicast end-to-end delay of the old SPT, the multicast delay variation of the new SPT will not be increased obviously; if it is greater than the maximum multicast end-to-end delay of the old SPT, it will become the maximum multicast end-to-end delay of the new SPT. Also from the above expression, we get $\max_{j \in M'} \{Delay(P'(v_c, j))\} \le MinDelay + \delta$, i.e., $\max_{j \in M'} \{Delay(P'(v_c, j))\} - MinDelay \le \delta$. Hence the multicast delay variation of the new SPT will be less than or equal to the old SPT. So the new SPT does not increase the multicast delay variation. ■

Theorem 6.3: DDVMA can (1) find a multicast tree satisfying the multicast end-to-end delay constraint as long as such a tree exists in the network; and (2) achieve better efficiency in terms of the multicast delay variation than DDVCA.

Proof: In step 1 of DDVMA, since each network node is checked, the source gateway v_s is also likely to be selected as the central node. The multicast tree will be constructed by connecting v_s to each destination gateway through the minimum delay path. Such a multicast tree is the SPT from v_s to all destination gateways. If it does not satisfy the multicast end-to-end delay constraint, obviously there does not exist any multicast tree, which will satisfy the multicast end-to-end delay constraint regulated by the input. This characteristic was also stated in [SHE02].

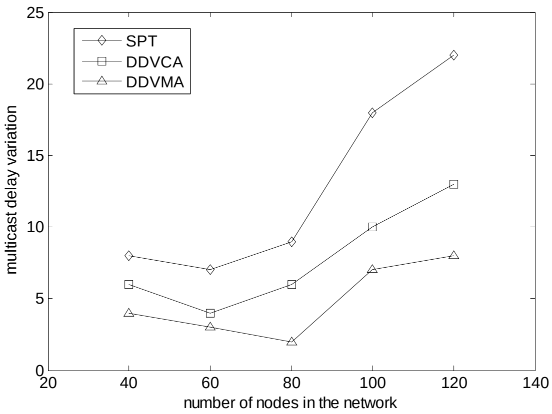
For each network node being checked, when the SPT is constructed, the multicast delay variation of the SPT will be used in DDVCA. However, in DDVMA, we will execute algorithm P and PP to further reduce the multicast delay variation of the SPT. From Theorem 6.1 and Theorem 6.2, we know algorithm P and PP can keep the SPT unchanged or return an improved SPT with smaller multicast delay variation. Algorithm P or PP will be called repeatedly until the SPT cannot be improved. So DDVMA will search more possible multicast trees and achieve better efficiency in terms of the multicast delay variation than DDVCA. ■

6.4.4 Performance Evaluation

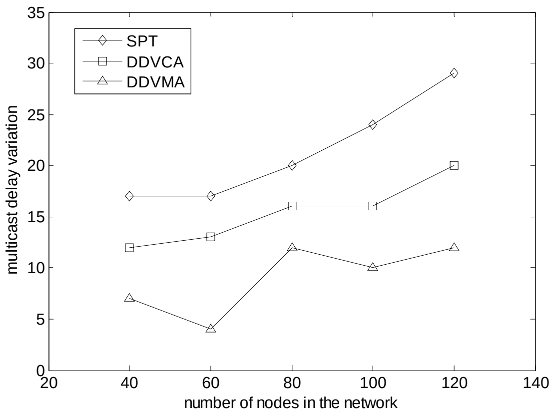
In this section, simulation experiments are conducted to examine the efficiency of DDVMA. Given two integers *n* and *m* ($n-1 \le m \le n(n-1)/2$), an interval [*LD*, *UD*], and an integer *d*, our random graph generator will generate a connected network topology graph with *n* nodes and *m* links. The delay on each link are an integer value in [*LD*, *UD*], which is in direct proportion to the length of the link. The degree of each node does not exceed *d*. The random graph generator first generates the *n* nodes. It then picks out two different nodes randomly. For the two nodes, if no direct link connects them and both of their node degrees are less than *d*, a new link between them will be added to the graph. This process is continued until *m* links are added to the graph. Similar random graph generation approach is introduced in [WAX88]. In our simulation experiments, we generate five different network topology graphs. The sizes of them begin from ones with 40, 60, 80, and up to 120 nodes. The delay on each link is drawn from the interval [1, 10]. Referring to [SHE02], for a specified multicast group, the upper bound on the maximum multicast end-to-end delay, Δ, is set to be 1.5 times the minimum delay between the source node and the farthest destination node.

In the simulation, we compare the proposed algorithm DDVMA with DDVCA and the SPT Algorithm produced from Dijkstra's Algorithm. We will evaluate the multicast delay variations and multicast end-to-end delays of the three algorithms. For each network, we investigate two cases, one is that the destination nodes in the multicast group will occupy 5% of the total nodes on the network and the other is 20%. For each case, we generate twenty different multicast groups randomly. Then twenty multicast trees will be obtained by each algorithm. We calculate the average over the multicast delay variations of the twenty multicast trees for each algorithm. The average value will be used to evaluate the efficiency of the algorithm in terms of the multicast delay variation.

Figure 6.6 shows the simulation results of multicast delay variations. Figure 6.6(a) corresponds to the multicast groups of sizes equal to 5% of the number of network nodes. It can be regarded as the scenario that multicast nodes are distributed sparsely in the network. Figure 6.6(b) corresponds to the multicast groups of sizes equal to 20% of the number of network nodes. It represents the scenario that multicast nodes are distributed densely in the network. We observe that the trees constructed by DDVMA have an average multicast delay variation that is always smaller than that of the SPT and DDVCA trees. With the ratio of the multicast group size to the number of network nodes increasing from 5% to 20%, it is apparent that the multicast delay variation of DDVMA performs much better than that of DDVCA. The performance of the SPT Algorithm is the worst in terms of the multicast delay variation among the three algorithms.



(a)



(b)

Fig. 6.6: A comparison on the multicast delay variations of the three different algorithms: (a) multicast group sizes equal to 5% of the number of network nodes, (b) multicast group sizes equal to 20% of the number of network nodes

To evaluate the performance of each algorithm, we also calculate the average over the maximum multicast end-to-end delays of the obtained multicast trees for each algorithm. Figure 6.7 shows the simulation results on the multicast end-to-end delays of different algorithms. It corresponds to the case in which the destination nodes in a multicast group occupy 5% of the total network nodes. The simulation result of the 20% case is similar. We only present and discuss the 5% case. We observe that the multicast end-to-end delay of DDVCA performs better than that of DDVMA, but not much. It can be explained by the design of DDVMA. In DDVMA, we improve the multicast delay variations of the SPTs by introducing higher delay paths. If the delay of the accepted new path exceeds the maximum multicast end-to-end delay of the SPT, the maximum multicast end-to-end delay of the multicast tree will increase. However, if the delay of the new path is so high that the multicast delay variation of the SPT is increased, the path will not be accepted. So in average, we can see that DDVMA and DDVCA have competing performance on multicast end-to-end delays. The SPT Algorithm has the best performance in terms of the multicast end-to-end delay inherently.

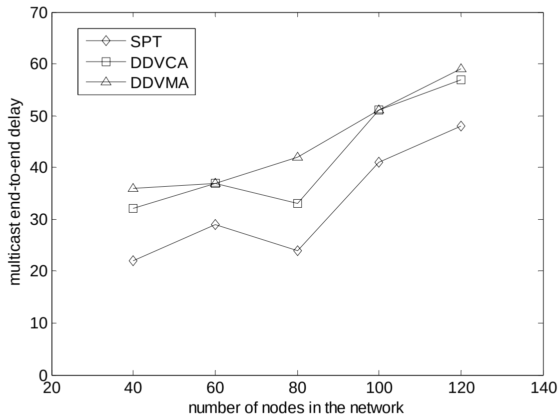


Fig. 6.7: A comparison on the multicast end-to-end delays of the three algorithms

6.4.5 Handling Mobility

For mobile communications, handling mobility of the MHs is an important issue. For the QoS mobile group communications in the iMANET heterogeneous network, the multicast tree needs to support host mobility by reconstructing the multicast path to the MH's new location adaptively. We confine the movement of an MH within its local MANET. If one MH moves to a new location, the AODV routing protocol will discover the new wireless route between the MH and its gateway. The delay of the new wireless route will be collected again. The delay of the new wireless route may decrease or increase. This will lead to the decrease or increase of the multicast end-to-end delay between the source MH and the destination MH because the delay of the new wireless route is part of the multicast end-to-end delay. If the decrease or increase of the multicast end-to-end delay does not make the multicast delay variation of the mobile multicast tree intolerable, the multicast tree in the backbone network can still be used; otherwise, it needs to be reconstructed locally.

As we know, the multicast tree obtained by DDVMA is the combination of the shortest path between the source gateway and a central node and a tree from the central node to all the destination gateways. The tree is an SPT based on the associated network topology. To handle mobility, the associated network topology needs to be recorded after the multicast tree is determined in the backbone network. The following operation can be conducted based on the associated network topology.

For the case that the movement of a destination MH leads to the decrease of the multicast end-to-end delay, we first compute the proprietary second shortest paths or partially proprietary second shortest paths between the central node and the corresponding destination gateway. Then, we select the one whose replacement on the SPT will improve the multicast delay variation the most under the multicast end-to-end delay constraint and use it as the new multicast path.

For the case that the movement of a destination MH leads to the increase of the multicast end-to-end delay, the increased multicast end-to-end delay will become the maximum multicast end-to-end delay of the multicast tree. Then for the destination gateway with the minimum multicast end-to-end delay on the SPT, we compute the proprietary second shortest paths or partially proprietary second shortest paths for it. We also select the one, whose replacement will improve the multicast delay variation the most under the multicast end-to-end delay constraint, to be the new multicast path. For the updated SPT, repeat this process until the multicast delay variation can not be improved. Finally, use the new SPT to replace the old SPT on the multicast tree.

6.5 The Improved CBT+SP Heuristic Algorithm

In this section, we analyze the drawbacks of the CBT+SP heuristic in depth. Since DDVCA exactly employs the CBT+SP heuristic, DDVCA also has these drawbacks. In the following, we will use the CBT+SP heuristic to represent DDVCA. To eliminate these drawbacks, we propose the improved CBT+SP heuristic algorithm to construct a multicast tree with small multicast delay variation under the multicast end-to-end delay constraint. The improved heuristic algorithm really improves the CBT+SP heuristic, as shown by theoretical proof. Simulation experiments also verify the theoretical analysis.

6.5.1 Analysis of the CBT+SP Heuristic Algorithm

The original CBT+SP heuristic has drawbacks because it does not consider the following two cases:

(a) possible existence of one destination node, which is an intermediate node of the shortest path from the source node to the central node;

(b) possible existence of one link, which is shared by the shortest path from the source node to the central node and the shortest path from the central node to one destination node.

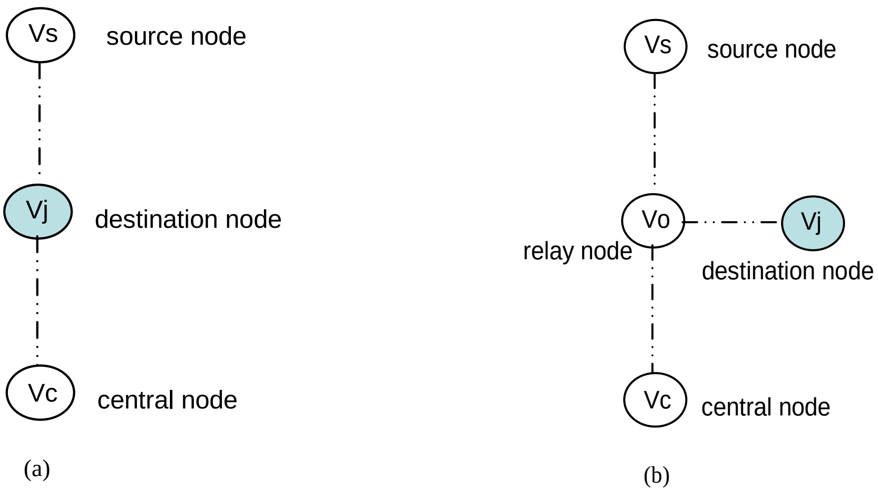


Fig. 6.8: The illustration of the drawbacks in the CBT+SP heuristic

These two cases are illustrated in Figure 6.8(a) and (b), respectively. For both cases, when the shortest path from the source node to the central node is connected with the SPT rooted at the central node, there will be some overlapping links. So the structure of the final multicast tree is not the expected structure. Therefore, its delay variation is not yet the delay variation of the SPT rooted at the central node. To avoid this, the algorithm should be able to detect these two cases when calculating the SPT rooted at a network node.

Kim *et al*. [KIM04a] pointed out the problem related to case (a) and proposed an improvement algorithm [KIM04b] to solve it but not correct. The improvement algorithm only checks whether any destination node is visited on the path from the source node to the node being checked. But for case (b), there is no any destination node on the path from the source node to the node being checked, hence the problem still exists.

In the following, we explain why the improvement algorithm does not work. The algorithm first calculates the delay variation of the SPT connecting each network node to all the destination nodes. If the SPT rooted at some network node has the minimum delay variation and the final multicast tree constructed by using this node as the central node satisfies the multicast end-to-end delay constraint, the node will be added into the candidate set of central node. The above procedure is the same as the CBT+SP heuristic. The difference is that the algorithm records $pass(s, v_i, m_k) = Dij(s, m_k)$ when any destination node m_k is visited on the path from the source node s to a network node c_i. $Dij(s, m_k)$ is the minimum delay from s to m_k. For the nodes in the candidate set, the algorithm chooses the node with $min\{\phi_D(P(s,c_i)) - min\{pass(s,c_i,m_j)\}\}$ as the central node. $\phi_D(P(s,c_i))$ is just $Dij(s,c_i)$, the minimum delay from s to c_i. $pass(s,c_i,m_j)$ is $Dij(s,m_j)$, the minimum delay from s to m_j. Actually, $\phi_D(P(s,c_i)) - min\{pass(s,c_i,m_j)\} = max\{Dij(m_j,c_i)\}$. So the algorithm chooses the node with $min\{max\{Dij(m_j,c_i)\}\}$. This means that for a node c_i in the candidate set, the algorithm calculates $max\{Dij(m_j,c_i)\}$, the minimum delay from it to the farthest destination node m_j, if there exists at least one destination node on the shortest path from s to c_i. Then the node with the minimum $max\{Dij(m_j,c_i)\}$ is chosen as the central node. Figure 6 in [KIM04b] illustrates the selection method of central node.

The improvement algorithm has the following drawbacks:

(1) It only considers the problem in case (a).

(2) Even for case (a), it constructs the candidate set of central nodes in terms of the delay variation of the SPT connecting a network node to all the destination nodes. As we have shown that, such a delay variation is not always equal to the delay variation of the final multicast tree. It is the same problem in the CBT+SP heuristic. Furthermore, only the nodes whose SPTs have the same minimum delay variations will be added into the candidate set. This will lead to a very small candidate set. When the size of the candidate set is 1, the algorithm just turns to be the CBT+SP heuristic. So compared to the CBT+SP heuristic, the improvement made by the algorithm highly depends on the size of the candidate set of central nodes.

(3) Even for the selection of central node among the candidate set, the solution proposed by the algorithm is wrong. For Figure 6 in [KIM04b], if we change the delay values of some links as shown in Figure 6.9, the algorithm will still choose V8 as the central node because $Dij(\mathrm{V5}, \mathrm{V8}) = 3 < Dij(\mathrm{V5}, \mathrm{V4}) = 4$. But the delay variation of the final multicast tree shown in Figure 6.9(b) is 8, which is higher than the multicast tree shown in Figure 6.9(a). So V4 should be chosen as the central node. The algorithm makes wrong decision at this time.

In addition, both the CBT+SP heuristic and the mentioned improvement algorithm have another drawback in the method of checking the multicast end-to-end delay constraint. In both of them, when checking a multicast tree constructed by using a network node as the central node, the addition of the minimum delay between the node and the source node and the maximum of the minimum delays between the node and each destination node is regarded as the maximum multicast end-to-end delay. But if case (a) or (b) occurs, the final multicast tree is not the expected structure, which leads to that not only the multicast delay variation but also the maximum multicast end-to-end delay of the final multicast tree is not the expected value. Hence both of the algorithms can not always correctly judge if the final multicast tree satisfies the multicast end-to-end delay constraint.

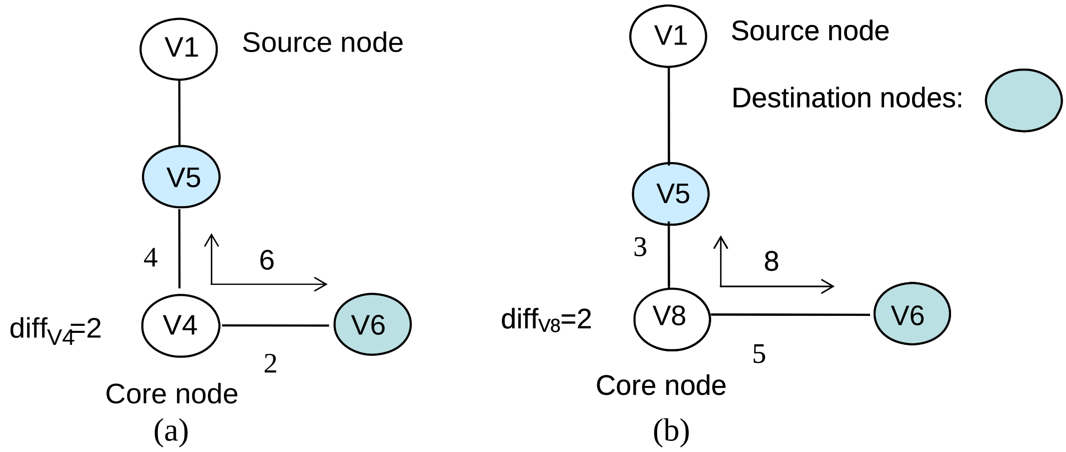


Fig. 6.9: The illustration of how the previous proposed improvement algorithm [KIM04b] selects wrong central node

In the improved CBT+SP heuristic algorithm, both case (a) and (b) are considered and the problems caused by them are solved completely and efficiently. The time complexity still remains the same as the original CBT+SP heuristic. Moreover, the improved CBT+SP heuristic algorithm uses the wireless routing information collected by gateway nodes using AODV protocol and takes the wireless routing delay into the calculation. Thus it can support the QoS mobile group communications in the iMANET heterogeneous network.

6.5.2 A Formal Description of the Algorithm

Both the CBT+SP heuristic and the improved heuristic algorithm check all the network nodes and select one of them as the central node. The difference between them is in the central node selection method. The CBT+SP heuristic selects the one whose SPT to all the destinations has the smallest delay variation. Then connect it with the source node and all the destination nodes to construct the final multicast tree. However, it is possible that the multicast delay variation of the final multicast tree is not equal to the delay variation of the

SPT rooted at the selected central node. So, although the SPT rooted at the selected central node has smaller delay variation than the SPTs rooted at other network nodes, the multicast delay variation of the final multicast tree may be larger than other multicast trees with different central nodes. As analyzed below, the CBT+SP heuristic may not work correctly due to this selection method. Different from the CBT+SP heuristic, the proposed algorithm has integrated the multicast tree construction into the selection of central node. When one node is checked as a candidate central node, the proposed algorithm will evaluate it by the multicast delay variation of the multicast tree, which is constructed by connecting the candidate central node with the source node and all the destination nodes, instead of the delay variation of the SPT rooted at it. Hence, the central node selected by the proposed algorithm can help construct the multicast tree with the smallest multicast delay variation among all the explored multicast trees.

In the following, we present a formal description of our algorithm. In the beginning, the algorithm sets the necessary variables. Variable dv_{min} records the minimum multicast delay variation, *T* records the present constructed multicast tree, v_c records the stand-by central node. For each network node $v_i \in V$, Line 4 calculates the shortest path *P* from the source node to it. $ppd(v_1, v_2)$ represents the minimum delay between v_1 and v_2, which is obtained by Dijkstra's Algorithm. For each node p_j on *P*, $UpDelay(p_j)$ denotes the upper bound on the end-to-end delay from p_j to its any child destination node on the multicast tree, $Max(p_j)$ and $Min(p_j)$ record the present maximum and minimum one among the delays of the paths from p_j to all its child destination nodes on the multicast tree, respectively. The variable *Flag* defined in Line 5 is used to indicate whether the multicast end-to-end delay constraint is satisfied if v_i is chosen as the central node. In Line 6-17, v_i calculates the shortest path to each v_w, denote p_k as the node which is on *P* and is also the nearest node to v_w on the shortest path. Then the algorithm will check whether the sum of the path delay from p_k to v_w and the wireless routing delay from

v_w to the team leader in the MANET does not exceed $UpDelay(p_k)$. If the delay constraint is not satisfied, *Flag* will be set as False. Otherwise, $Max(p_k)$ and $Min(p_k)$ will be updated. The update rule is as follows: if the delay sum is greater than $Max(p_k)$, $Max(p_k)$ will be replaced by the delay sum; otherwise the algorithm compares the delay sum with $Min(p_k)$, if the former is less, $Min(p_k)$ will be replaced by the delay sum.

Step 1 Using AODV protocol to get the delay of the wireless routing path between each team leader and its gateway

Step 2 Multicast tree construction in the backbone wired network

begin
$dv_{min} = \infty$, $T = \phi$, $v_c = \phi$;
for each $v_i \in V$ do
using Dijkstra's Algorithm to calculate P, the shortest path from v_s to v_i. For each node p_j on P, record $ppd(v_s, p_j)$ and initialize three variables: $UpDelay(p_j) = \Delta - ppd(v_s, p_j)$, $Max(p_j) = 0$, $Min(p_j) = \infty$;
Flag=true;
for each $v_w \in M$ do
calculate the shortest path from v_i to v_w, denote p_k as the node which is on P and is also the nearest node to v_w on the shortest path;
if $\{ppd(v_w, p_k) + W(v_w) > UpDelay(p_k)\}$
then *Flag*=false; break;
else if $\{ppd(v_w, p_k) + W(v_w) < Min[p_k]\}$
then $Min[p_k] = ppd(v_w, p_k) + W(v_w)$;
else if $\{ppd(v_w, p_k) + W(v_w) > Max[p_k]\}$
then $Max[p_k] = ppd(v_w, p_k) + W(v_w)$;
end if
end if
end if
end of for $v_w \in M$ loop
if (*Flag* = = true)
then $dv(v_i) = \max_{p_j \in P}\{ppd(v_s, p_j) + Max(p_j)\} - \min_{p_j \in P}\{ppd(v_s, p_j) + Min(p_j)\}$;
if $dv_{min} > dv(v_i)$ then $dv_{min} = dv(v_i)$; $v_c = v_i$;
end if
end if
end of for each $v_i \in V$ loop
if $v_c = \phi$ then print "no such multicast tree exists!";
for each $v_w \in M$ do
$T = T \cup \{l \mid l \in$ the minimum delay path from v_w to $v_c\}$;
end of for each $v_w \in M$ loop
$T = T \cup \{l \mid l \in$ the minimum delay path from v_s to $v_c\}$;
return *T*;
end of the algorithm

For v_i, after all the destination nodes have been checked, the algorithm checks the variable *Flag*. If *Flag* remains True, v_i can be chosen as the central node. Then the algorithm calculates the delay variation $dv(v_i)$, which is equal to $\max_{p_j \in P}\{ppd(v_s, p_j) + Max(p_j)\} - \min_{p_j \in P}\{ppd(v_s, p_j) + Min(p_j)\}$. $\max_{p_j \in P}\{ppd(v_s, p_j) + Max(p_j)\}$ and $\min_{p_j \in P}\{ppd(v_s, p_j) + Min(p_j)\}$ are the maximum and minimum multicast end-to-end delay from the source node to destination nodes, respectively. If $dv(v_i)$ is less than dv_{min}, v_i will become the current stand-by central node v_c. Finally, when all the network nodes have been checked, v_c is the selected central node. Line 24 deals with the case that v_c is NULL. In Line 25-29, if v_c is not NULL, the final multicast tree is constructed by connecting the source node to v_c and connecting v_c to all the destination nodes, all using the shortest path.

From the description of the algorithm, we can see that the advantage of our algorithm is that the actual multicast end-to-end delays are recorded in the nodes on the path from the source node to the node being checked. So the improved CBT+SP heuristic algorithm can achieve the actual multicast delay variation of the multicast tree, which is constructed using the checked node as the central node. Also, the multicast end-to-end delay constraint is checked for each destination node when the actual multicast end-to-end delay is recorded. Clearly, the improved heuristic algorithm overcomes the drawbacks of the CBT+SP heuristic.

6.5.3 Theoretical Analysis

As shown above, both the CBT+SP heuristic and the improved CBT+SP heuristic algorithm have adopted the same fundamental strategy of combining CBT and the minimum delay path algorithm. Since each network node needs to be checked before the final multicast tree is constructed, this strategy will guide the algorithms to explore *n* multicast trees, where *n* is the total number of network nodes. Hence, both the CBT+SP heuristic and our algorithm will select the final multicast tree from the same set of multicast trees. So we need to prove that the multicast tree selected by our algorithm can achieve better performance than the one selected by the CBT+SP heuristic in terms of multicast delay variation under the same multicast end-to-end delay constraint. We also need to prove that our algorithm's time complexity is not worse than that of the CBT+SP heuristic.

Lemma 6.4: Under the same multicast end-to-end delay constraint, if the CBT+SP heuristic finds that one multicast tree satisfies the constraint, our algorithm can also do.

Proof: Assume the multicast end-to-end delay constraint is Δ. For each candidate central node v_c, we denote the node with the maximum end-to-end delay on $T(v_c)$ as v_i. If $P(v_s, v_c) \cap T(v_c) = \{v_c\}$, the shortest path from v_s to v_c and the SPT rooted at v_c intersects only at v_c. Hence, the CBT+SP heuristic will calculate the maximum multicast end-to-end delay as $D_{P(v_s,v_c)} + D_{P(v_c,v_i)}$ and our algorithm will calculate it as $D_{P(v_s,v_c)+P(v_c,v_i)}$. Since $P(v_s, v_c) \cap T(v_c) = \{v_c\}$ and $P(v_c, v_i) \subseteq T(v_c)$, we get $P(v_s, v_c) \cap P(v_c, v_i) = \{v_c\}$, and hence we have $D_{P(v_s,v_c)+P(v_c,v_i)} = D_{P(v_s,v_c)} + D_{P(v_c,v_i)}$. If $D_{P(v_s,v_c)} + D_{P(v_c,v_i)} \le \Delta$, then $D_{P(v_s,v_c)+P(v_c,v_i)} \le \Delta$.

If $P(v_s, v_c) \cap T(v_c) \supset \{v_c\}$, the shortest path from v_s to v_c and the SPT rooted at v_c also intersects at other nodes or links in addition to v_c. However, the CBT+SP heuristic also calculates the maximum multicast end-to-end delay as $D_{P(v_s,v_c)} + D_{P(v_c,v_i)}$. Since $P(v_c, v_i) \subseteq T(v_c)$, if $P(v_s, v_c) \cap P(v_c, v_i) = \{v_c\}$, it is the same as the above case. Hence we only need to consider that $P(v_s, v_c) \cap P(v_c, v_i) \supset \{v_c\}$, i.e., v_i is on $P(v_s, v_c)$ or $P(v_s, v_c)$ shares at least one link with $P(v_s, v_i)$.

(1) If v_i is on $P(v_s,v_c)$, our algorithm will calculate the maximum multicast end-to-end delay as $D_{P(v_s,v_c)} + Max\{D_{P(v_c,v_k)} \mid v_k \in T(v_c) \wedge v_k \notin P(v_s,v_c)\}$. Since $Max\{D_{P(v_c,v_k)} \mid v_k \in T(v_c) \wedge v_k \notin P(v_s,v_c)\} < D_{P(v_c,v_i)}$, we have $D_{P(v_s,v_c)} + Max\{D_{P(v_c,v_k)} \mid v_k \in T(v_c) \wedge v_k \notin P(v_s,v_c)\} < D_{P(v_s,v_c)} + D_{P(v_c,v_i)}$. Therefore, if $D_{P(v_s,v_c)} + D_{P(v_c,v_i)} \le \Delta$, then $D_{P(v_s,v_c)} + Max\{D_{P(v_c,v_k)} \mid v_k \in T(v_c) \wedge v_k \notin P(v_s,v_c)\} \le \Delta$.

(2) If $P(v_s,v_c)$ shares at least one link with $P(v_s,v_i)$, assume that $v_{i'}$ is the node that, among all the destinations, has the maximum multicast end-to-end delay from the source v_s. Assume v_o ($v_o \in P(v_s,v_c)$) has the shortest delay to $v_{i'}$ on $P(v_s,v_c)$. Then our algorithm will calculate the maximum multicast end-to-end delay as $D_{P(v_s,v_{i'})} = D_{p(v_s,v_o)} + D_{P(v_o,v_{i'})}$. Since $D_{P(v_s,v_o)} \le D_{P(v_s,v_c)}$ and $D_{P(v_o,v_{i'})} \le D_{P(v_c,v_i)}$, we have $D_{P(v_s,v_{i'})} = D_{p(v_s,v_o)} + D_{P(v_o,v_{i'})} \le D_{P(v_s,v_c)} + D_{P(v_c,v_i)}$. Therefore, if $D_{P(v_s,v_c)} + D_{P(v_c,v_i)} \le \Delta$, then $D_{P(v_s,v_{i'})} \le \Delta$.

As a result, for the set of multicast trees, which are considered to satisfy the multicast end-to-end delay constraint by the CBT+SP heuristic, our algorithm also considers them satisfying the same constraint. ■

Theorem 6.4: Under the same multicast end-to-end delay constraint, our algorithm can achieve better performance than the CBT+SP heuristic in terms of multicast delay variation.

Proof: We use T_1 to denote the set of multicast trees, which are considered to satisfy the multicast end-to-end delay constraint by the CBT+SP heuristic. We use T_2 to denote the set of multicast trees, which are considered by our algorithm to satisfy the multicast end-to-end delay constraint. the CBT+SP heuristic will select the final multicast tree from T_1 and our algorithm will select the final multicast tree from T_2. By *Lemma 6.4*, we have $T_1 \subseteq T_2$. Clearly, both T_1 and T_2 are the subsets of the *n* multicast trees explored by the fundamental strategy. Without loss of generality, we assume that the CBT+SP heuristic will select t' from T_1 and our algorithm will select t^* from T_2. Since our algorithm selects the final multicast tree according to the multicast delay variation of each tree in T_2, clearly t^* is the tree with the smallest multicast delay variation $dv(t^*)$ in T_2.

If $T_1 = T_2$, then the CBT+SP heuristic and our algorithm will select the final multicast tree from the same set of multicast trees denoted by T ($T = T_1 = T_2$). Since $t' \in T$, we have $dv(t^*) \leq dv(t')$.

If $T_1 \subset T_2$, we set $T_0 = T_2 - T_1$. If $t^* \in T_2 - T_0$, the proof is the same as in the above case of $T_1 = T_2$. If $t^* \in T_0$, we assume t_1^* is the one with the smallest multicast delay variation in $T_2 - T_0$ (i.e., T_1). Clearly, we have $dv(t_1^*) \leq dv(t')$. Since t^* has the smallest multicast delay variation in T_2, we have $dv(t^*) \leq dv(t_1^*)$. Then we get $dv(t^*) \leq dv(t')$.

As a result, under the same multicast end-to-end delay constraint, our algorithm can achieve better performance in terms of multicast delay variation. ■

Theorem 6.5: Our algorithm has the same time complexity as the CBT+SP heuristic.

Proof: Just like the CBT+SP heuristic, the computation overhead of our algorithm is mainly on the calculation of the shortest delay paths. It is known that the time complexity of calculating the shortest path by Dijkstra's Algorithm is $O(n^2)$, where n is the number of network nodes. For each network node, our algorithm firstly calculates the shortest path from the source node to it, and then calculates the shortest path from it to each destination node. For each destination node, the CBT+SP heuristic also calculates the shortest paths from it to all the network nodes. So the number of shortest path calculation in our algorithm is the same as that in the CBT+SP heuristic. Hence our algorithm has the same time complexity as the CBT+SP heuristic.

6.5.4 An Illustrative Example of the Algorithm

In the following, we will illustrate the operation of our algorithm using an example. We will compare it with the CBT+SP heuristic. We use the given network topology as shown in Figure 6.10. For a group communication scenario, we denote Vs as the source gateway, V2, V5 and V8 as the destination gateways, i.e. M={V2, V5, V8}. The number in the parentheses near gateway g (including the source gateway and all the destination gateways) represents the corresponding wireless routing delay $W(g)$. Suppose the multicast end-to-end delay constraint Δ is 60. Because the wireless routing delay between the source leader MH and the source gateway is 1, the multicast end-to-end delay constraint used in our algorithm will be 59 (i.e., 60-1).

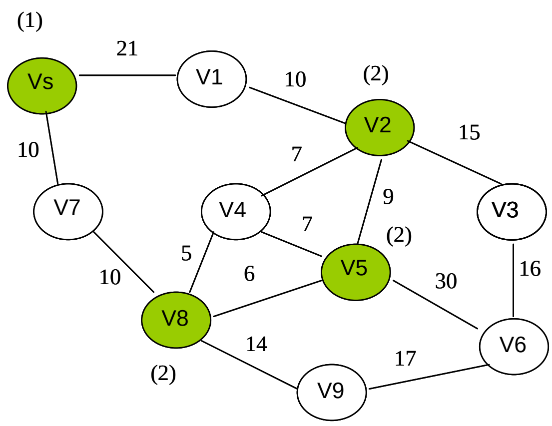


Fig. 6.10: The simulation network topology

If the CBT+SP heuristic is adopted, when all the network nodes have been checked, V4 is selected as the central node. The delay variation calculated by the CBT+SP heuristic is 2. But the actual delay variation of the final multicast tree is 12. The mistake is made due to the drawbacks of the CBT+SP heuristic. Because the CBT+SP heuristic is only used in wireline network, we ignore the effect of wireless routing delays to make our algorithm comparable to the CBT+SP heuristic. In the following discussion, we do not count the wireless routing delay into the end-to-end delay.

In our algorithm, when V4 is checked, the shortest path *P* from Vs to V4 is first calculated and two variables *Max*[Vi] and *Min*[Vi] are initialized in each node on *P*, as shown in Figure 6.11(a). Then V4 calculates the shortest delay path to V8. The nearest node to V8 on *P* is V8 itself. $ppd(\text{V8}, \text{V8}) = 0$. Hence *Min*[V8] is updated to be 0, as shown in Figure 6.11(b). V4 continues to calculate the shortest delay path to V2. The nearest node to V2 on *P* is V4. $ppd(\text{V4}, \text{V2}) = 7$. Because $ppd(\text{V4}, \text{V2}) > Max[\text{V4}]$ and $ppd(\text{V4}, \text{V2}) < Min[\text{V4}]$, the algorithm updates both *Max*[V4] and *Min*[V4] to be 7, as shown in Figure 6.11(c). Finally, V4 calculates the shortest delay path to V5. The nearest node to V5 on *P* is also V4. $ppd(\text{V4}, \text{V5}) = 6$. Because $ppd(\text{V4}, \text{V5}) < Min[\text{V4}]$, $Min[\text{V4}]$ is updated to be 6, as shown in Figure 6.11(d).

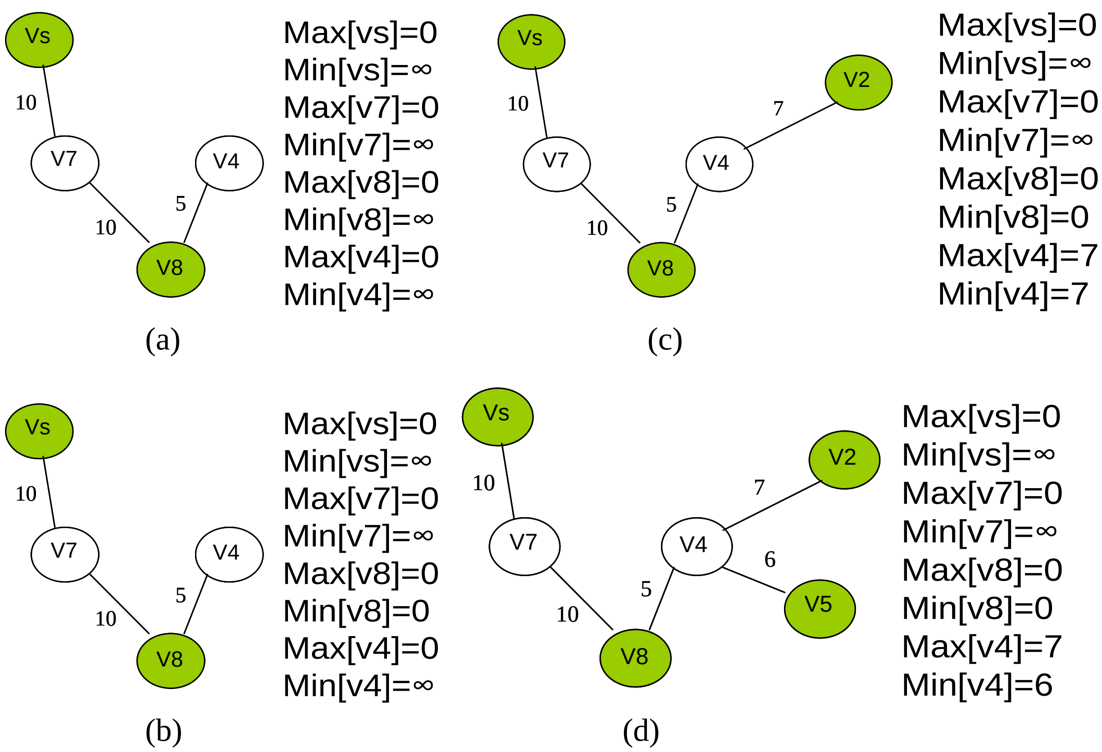


Fig. 6.11: The update procedure of delay variables in the improved heuristic when V4 is the candidate central node

By comparison, we can see that the multicast end-to-end delay constraint is satisfied, so the algorithm continues to calculate the delay variation. For each node on *P*, the maximum value calculated by $\max_{p_j \in P}\{ppd(v_s, p_j) + Max(p_j)\}$ is $ppd(\text{Vs}, \text{V4}) + Max(\text{V4}) = 32$, the minimum value calculated by $\min_{p_j \in P}\{ppd(v_s, p_j) + Min(p_j)\}$ is $ppd(\text{Vs}, \text{V8}) + Min(\text{V8}) = 20$. So the multicast delay variation $dv(\text{V4}) = 32 - 20 = 12$. Thus our algorithm can achieve the actual delay variation.

The CBT+SP heuristic also calculates wrong delay variation value 9 when V5 is checked. Our algorithm calculates the correct value 15. In our algorithm, when Vs is checked, we get the shortest path tree from Vs to all the destination nodes with delay variation equal to 11. The shortest path tree is also the optimal multicast tree. For V6, the CBT+SP heuristic can judge that the multicast end-to-end delay constraint is not satisfied after calculating the delay variation. But our algorithm detects it as early as possible before the delay variation calculation. Thus the further calculation can be stopped to avoid unnecessary overhead.

The illustrated example shows that our algorithm can achieve the optimal solution, but the CBT+SP heuristic cannot. Our algorithm outperforms the CBT+SP heuristic. In next section, we will use extensive experiments to show the performance improvement of our algorithm over the CBT+SP heuristic.

6.5.5 Performance Evaluation

In Section 6.5.1, we have described the drawbacks of both the CBT+SP heuristic and one previously proposed improvement algorithm. We point out that the improvement algorithm does not work. In Section 6.5.4, an example is given to show that our algorithm really improves and outperforms the original CBT+SP heuristic. To verify the performance of our algorithm, simulation experiments are conducted. We generate the connected network topology with *n* nodes and *m* links ($n-1 \le m \le n(n-1)/2$). The degree of each node does not exceed *d*. The delay on each link is an integer value, which falls into the range [*LD*, *UD*] and is in direct proportion to the length of the link. For a multicast group, Δ, the upper bound of the maximum multicast end-to-end delay, is set to be 1.5 times the minimum delay between the source node and the farthest destination node.

Five wireline network topologies are generated with *n*=20, 40, 60, 80, 100. Other parameters are set as *d*=6, *LD*=1, *UD*=10. In this simulation, we will compare our algorithm with the CBT+SP heuristic. To compare with the CBT+SP heuristic, we ignore the wireless routing delay. Because we only add the wireless routing delay to the delay of the wireline path in the algorithm, the performance of the algorithm will not be affected although we do not take the wireless routing delay into account. We will evaluate the multicast delay variations and multicast end-to-end delays of the two algorithms. For each network, we investigate two cases, one is that the destination nodes in the multicast group occupy 10% of the total nodes in the network and the other is 30%. For each case, we generate twenty different multicast groups randomly on each network topology. Then twenty multicast trees will be obtained by each algorithm. We calculate the average over the multicast delay variations of the twenty multicast trees for each algorithm on each topology. The average value will be used to evaluate the efficiency of the algorithm in terms of the multicast delay variation.

Figure 6.12 shows the simulation results of multicast delay variations. Figure 6.12(a) corresponds to the multicast groups of sizes equal to 10% of the number of network nodes. It can be regarded as the scenario that multicast nodes are distributed sparsely in the network. Figure 6.12(b) corresponds to the multicast groups of sizes equal to 30% of the number of network nodes. It represents the scenario that multicast nodes are distributed densely in the network. We observe that the multicast trees constructed by our algorithm have an average multicast delay variation that is smaller than that of the trees generated by the CBT+SP heuristic. In Figure 6.12(b), it is apparent that the multicast delay variation of our algorithm performs much better than that of the CBT+SP heuristic with the ratio of the multicast group size to the number of network nodes increasing from 10% to 30%. But this phenomenon is not shown in Figure 6.12(a). It is because the two cases shown in Figure 6.8 appear more frequently in large multicast groups.

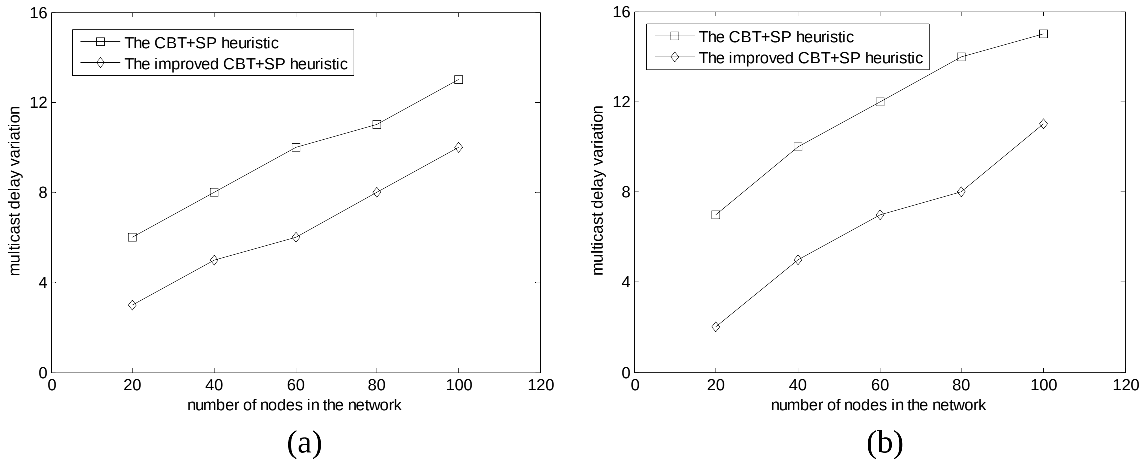


Fig. 6.12: A comparison on the multicast delay variations of both algorithms: (a) multicast group sizes equal to 10% of the number of network nodes, (b) multicast group sizes equal to 30% of the number of network nodes

To evaluate the performance of each algorithm, we also calculate the average over the maximum multicast end-to-end delays of the obtained multicast trees for each algorithm. Figure 6.13 shows the simulation results on the multicast end-to-end delays of both algorithms. It corresponds to the case in which the destination nodes in a multicast group occupy 10% of the total network nodes. The simulation result of the 30% case is similar. We only present and discuss the 10% case. It is observed that both algorithms achieve almost the same performance in terms of multicast end-to-end delay. This is because both algorithms check the multicast end-to-end constraint although our algorithm checks it earlier than the CBT+SP heuristic. The checking sequence does not affect the algorithm performance on multicast end-to-end delay.

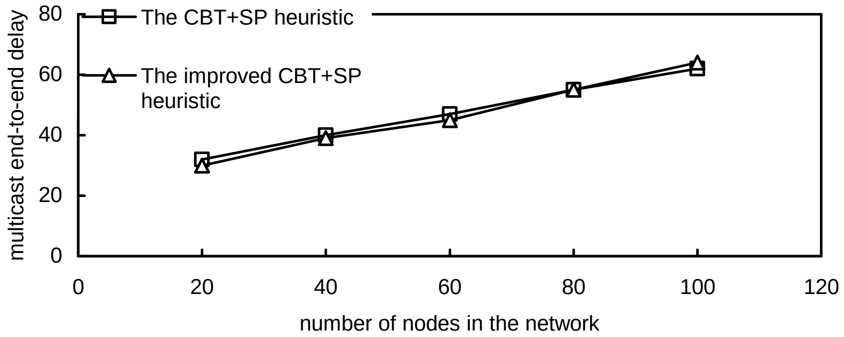


Fig. 6.13: A comparison on the multicast end-to-end delays of both algorithms

6.6 Summary

In this chapter, we develop two algorithms to support QoS mobile group communications in the iMANET heterogeneous network: DDVMA and the improved CBT+SP heuristic algorithm.

In DDVMA, two subalgorithms- algorithm P and algorithm PP are firstly proposed to calculate higher delay paths which are used to replace the corresponding shortest paths aiming to reduce the multicast delay variation of the SPT. Then a multicast tree is constructed from the source gateway to all the destination gateways in the iMANET heterogeneous network. Algorithm P and PP are integrated into DDVMA to further reduce the multicast delay variation of the SPT rooted at the stand-by central node. The QoS mobile multicast tree constructed can satisfy the multicast end-to-end delay constraint and achieve smaller multicast delay variation than the multicast tree obtained by DDVCA known to be the best algorithm for the DVBMT problem. Theoretical analysis is made on DDVMA and proves its correctness. Simulations also compare DDVMA with DDVCA and the SPT Algorithm in terms of multicast delay variations and multicast end-to-end delays. The simulation results show that DDVMA can achieve the smallest multicast delay variation with a litter higher multicast end-to-end delay than DDVCA.

The improved CBT+SP heuristic algorithm is developed based on a simple heuristic named as the CBT+SP heuristic. The CBT+SP heuristic has inherent drawbacks, which bring severe problems to the algorithms based on it. These problems are firstly analyzed. Inspired by the analysis, we propose the improved CBT+SP heuristic algorithm. Both theoretical analysis and simulations shows that our algorithm has better performance than the original heuristic in terms of multicast delay variation under the same multicast end-to-end delay constraint.

Chapter 7

Conclusions and Future Work

In this chapter, we briefly summarize our work and outline directions for future research.

7.1 Conclusions

In this thesis, we develop group-oriented protocols and algorithms for two types of wireless mobile networks: MANET and its Internet extension iMANET. The iMANET heterogeneous network has a backbone wired network and provides gateways to serve MANETs. We design a research framework on group-oriented communication. In this framework, we classify the groups into two types. One is named as aggregated group, in which nodes stay closely and move following a certain group mobility pattern. The other is named as distributed group, in which group members are distributed to different locations in the network.

Group mobility reflects the movement behavior of the aggregated group. In MANETs, group mobility becomes increasingly important due to its vast range of potential applications. In this thesis, we explore the effect of group mobility on various issues in MANETs. We first introduce four group mobility models, and then simply review the results reported on network partition prediction and replica allocation in MANETs with group mobility. We observe two properties from group mobility. One is the relative stability of distance between two neighboring group members and the other is motion affinity. Accordingly, based on these two properties, we present a clustering algorithm SMoC and a location service protocol GrLS, respectively.

SMoC is a stability-aware multi-objective clustering algorithm. When most of the nodes conform to group mobility pattern in an MANET, one node can determine a few relatively stable neighbors if they move together within a certain period of time. Then a relatively stable network topology can be derived. By running SMoC on the relatively stable topology, stability-aware cluster structure can be obtained. Additionally, SMoC considers multi-metric clustering by adopting a promising multi-objective evolutionary algorithm, SPEA2. Simulation experiments show that it outperforms the previous Weighted Clustering Algorithm.

GrLS stands for Group-based Location Service. It also applies to the network scenario in which group mobility dominates the mobility pattern. Since a group of nodes has similar mobility trajectory and shows motion affinity, in GrLS only group leaders recruit location servers and send location update messages to the location servers. Thus, GrLS produces less network traffic than prior schemes. GrLS divides the network coverage into equal circular-shaped regions and the division originates from the estimated network center. This can help reduce the average location update/query path length. In addition, a home region is further divided into several subregions and each subregion holds a location server for each node which has selected this home region. Thus, the load of acting as location servers can be evenly distributed in the whole network.

For the distributed group, group members need to perform efficient group communication to coordinate their work. Group communication needs the support of multicast protocols. We develop a hybrid multicast routing protocol GMZRP for MANETs. It is an on-demand source-based multicast routing protocol. It exploits geographic partition of the network to reduce the duplicate multicast route request packets. To make the multicast tree robust to node mobility, it maintains the routing information at two levels: node level and geographic zone level. Performance evaluation shows that it outperforms another prevalent on-demand multicast routing protocol, ODMRP.

In the iMANET heterogeneous network, we consider the group communication scenario that group members are located in different MANETs attached to different Internet gateways. Group communication goes through the backbone wired network. Hence, to guarantee QoS and construct efficient multicast tree is a challenging issue. We design two algorithms to support QoS group communication in the iMANET heterogeneous network: DDVMA and the improved CBT+SP heuristic algorithm. Both of them are designed for solving the delay- and delay-variation bounded multicast tree problem, which has been proved to be NP-complete.

DDVMA can find a multicast tree satisfying the multicast end-to-end delay constraint and minimizing the multicast delay variation. Two concepts are proposed, namely, 1) the proprietary second shortest path and 2) the partially proprietary second shortest path. They can help DDVMA achieve better performance in terms of multicast delay variation than DDVCA algorithm that is known to be the most efficient so far. Theoretical analysis is given to show the correctness of DDVMA, and simulations are conducted to demonstrate the performance improvement of DDVMA in terms of multicast delay variation. The strategy employed by DDVMA is also applicable to handling the mobility of mobile hosts in the iMANET heterogeneous network.

DDVCA exactly employs the CBT+SP heuristic, which uses the Core Based Tree with the shortest path to construct multicast tree. However, this heuristic has several drawbacks. Although one attempt was made to improve it, the drawbacks have not been overcome completely yet. By analyzing the drawbacks of the heuristic in depth, we explain why the previous attempt made by other researchers does not work. Then we propose the improved

CBT+SP heuristic algorithm which removes its drawbacks completely. Theoretical analysis proves that the improved heuristic algorithm has better performance than the original heuristic. Experimental results also verify the theoretical results.

7.2 Future Work

In real-life scenarios, collaborative applications become more and more important since few tasks can be accomplished by single person. In the network, many applications also require lots of nodes to work cooperatively, e.g., computer-supported cooperative work (CSCW). So far, applications of mobile ad hoc networks are envisioned mainly for crisis situations (e.g., in the battlefield or in rescue operations). In these scenarios, one mission is often conducted by a group of nodes.

In this thesis, we design a clustering algorithm and a location service protocol by exploiting group mobility. We think more issues can be reconsidered when group mobility is presented to MANETs. As the future work, we plan to investigate the following issues.

1) Group Structure

Depending on the applications, a group may have leader or not. Different structures will have different requirements on protocols and algorithms.

2) Location Service

In GrLS, it is assumed that there is a group leader in each group. However, it is possible that there is no leader in a group. Then the location service protocol should be revised.

In addition, the performance of GrLS can be improved by the following optimization techniques:

- *Location cache can be used. When forwarding a location update message, a node adds the location information it learns from the message to its location cache. The node associates a relatively short timeout value with the cached location information.*
- *When a node relaying a location query message finds that it is just the destination that the location query is for, it directly sends location reply to the source and drops the location query message.*
- *Location maintenance is used between two communication partners. When data transmission is conducted between a pair of nodes, their location information is periodically piggybacked to the data packets destined to the other end. Thus, they can know the accurate location information of each other without querying them again.*

3) Routing in Mobile Ad hoc Access Networks

In the iMANET heterogeneous network, each Internet gateway serves an MANET. Mobile ad hoc access network just refers to a group of mobile nodes attached to a fixed Internet gateway. In our proposed two QoS multicast algorithms, DDVMA and the improved CBT+SP heuristic, we just simply let the gateway collect the wireless routing information in the MANET it serves. However, to establish an efficient route in mobile ad hoc access network is also an interesting and challenging problem. In this type of network, most of the traffic goes through the gateway. Hence, the network architecture is gateway-centric. Intuitively, nodes closer to the gateway bear higher load. Hence, efficient load-aware routing protocols should be designed to support mobile ad hoc wireless access networks.

References

[AFI03] H. Afifi and D. Zeghlache ed., *Applications & Services in Wireless Networks*, London: Kogan Page Science, 2003.

[AHA98] E. Aharoni and R. Cohen, "Restricted Dynamic Steiner Trees for Scalable Multicast in Datagram Networks", *IEEE/ACM Transactions on Networking*, Vol. 6, No. 3, pp. 286 - 297, 1998.

[AN01] B. An and S. Papavassiliou, "A Mobility-based Clustering Approach to Support Mobility Management and Multicast Routing in Mobile Ad-hoc Wireless Networks", *International Journal of Network Management*, Vol. 11, No. 6, pp. 387–395, 2001.

[ANA01] G. Anastasi, A. Bartoli, and F. Spadoni, "A Reliable Multicast Protocol for Distributed Mobile Systems: Design and Evaluation", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 12, No. 10, pp. 1009-1022, 2001.

[BAS99] S. Basagni, "Distributed Clustering for Ad Hoc Networks", *Proc. the 4th International Symposium on Parallel Architectures, Algorithms and Networks*, 1999.

[BAT94] R.J. Bates, *Wireless Networked Communications: Concepts, Technology and Implementation*, New York: McGraw-Hill, 1994.

[BET03] C. Bettstetter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 3, pp. 257-269, 2003.

[BIR93] K. Birman, "The Process Group Approach to Reliable Distributed Computing", *Communications of the ACM*, Vol. 36, No. 12, pp. 37-53, 1993.

[BLA05] L. Blazevic, J.Y.L. Boudec, and S. Giordano, "A Location-based Routing Method for Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 2, pp. 97-110, 2005.

[CAN04] C.A.V. Campos, D.C. Otero, and L.F.M. de Moraes, "Realistic Individual Mobility Markovian Models for Mobile Ad hoc Networks", *Proc. WCNC*, 2004.

[CAO06] J.N. Cao, L.F. Zhang, G.J. Wang, and H. Cheng, "SSR: Segment-by-Segment Routing in Large-Scale Mobile Ad Hoc Networks", *Proc. 3rd IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (MASS), 2006.

[CHA02] M. Chatterjee, S.K. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks", *Cluster Computing*, Vol. 5, No. 2, pp. 193-204, 2002.

[CHA03] R.-S. Chang, W.-Y. Chen, and Y.-F. Wen, “Hybrid Wireless Network Protocols”, *IEEE Transactions on Vehicular Technology*, Vol. 52, No. 4, pp. 1099-1109, 2003.

[CHA04] M. Charikar, J. Naor, and B. Schieber, “Resource Optimization in QoS Multicast Routing of Real-time Multimedia”, *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, pp. 340-348, 2004.

[CHE02] K. Chen and K. Nahrstedt, “Effective Location-guided Tree Construction Algorithms for Small Group Multicast in MANET”, *Proc. INFOCOM*, 2002.

[CHE03] W.-T. Chen and P.-Y. Chen, “Group Mobility Management in Wireless Ad Hoc Networks”, *Proc. IEEE Semiannual Vehicular Technology Conference*, October 2003.

[CHE04] X.Z. Cheng, X. Huang, and D.-Z. Du, ed., *Ad Hoc Wireless Networking*, Kluwer Academic Publishers, 2004.

[CHE06] H. Cheng, J.N. Cao, and X.W. Wang, “Constructing Delay-bounded Multicast Tree with Optimal Delay Variation”, *Proc. ICC*, 2006.

[CHI98] C.-C. Chiang, M. Gerla, and L.X. Zhang, “Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks”, *Cluster Computing*, Vol. 1, No. 2, pp. 187-196, 1998.

[CHR03] Å. Christer and Z. Arkady, “Extending Global IP Connectivity for Ad Hoc Networks”, *Telecommunication Systems*, Vol. 24, No. 2, pp. 221-250, 2003.

[COR03] C.M. Cordeiro, H. Gossain, and D.P.Agrawal, “Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions”, *IEEE Network*, Vol. 17, No. 1, pp. 52-59, 2003.

[CUI03] Y. Cui, K. Xu, J.P. Wu, Z.C. Yu, and Y.J. Zhao, “Multi-constrained Routing based on Simulated Annealing”, *Proc. ICC*, 2003.

[DAS05] S.M. Das, H. Pucha, and Y.C. Hu, “Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing”, *Proc. INFOCOM*, 2005.

[DEB02] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithms: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.

[DEV01] V. Devarapalli and D. Sidhu, “MZR: A Multicast Protocol for Mobile Ad Hoc Networks”, *Proc. ICC*, 2001.

[DIJ59] E.W. Dijkstra, “A Note on Two Problems in Connexion with Graphs”, *Numerische Mathematik*, Vol. 1, pp. 269-271, 1959.

[DU06] X. Du and D. Wu, "Adaptive Cell-Relay Routing Protocol for Mobile ad hoc Networks", *IEEE Transactions on Vehicular Technology*, Vol. 55, No. 1, pp. 278-285, 2006.

[ER04] I.I. Er and W.K.G. Seah, "Mobility-based D-hop Clustering Algorithm for Mobile Ad Hoc Networks", *Proc. WCNC*, 2004.

[ESB95] H. Esbensen, "Compute Near-optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm", *Networks*, Vol. 26, pp. 173-185, 1995.

[GAO05] J. Gao, *et al*., "Geometric Spanners for Routing in Mobile Networks", *IEEE Journal of Selected Areas in Communications*, Vol. 23, No. 1, pp. 174-185, 2005.

[GAO06] J. Gao and L. Zhang, "Load-Balanced Short-Path Routing in Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, No. 4, pp. 377-388, 2006.

[GAL04] L. Galluccio, G. Morabito, and S. Palazzo, "Spontaneous Group Management in Mobile Ad Hoc Networks", *Wireless Networks*, Vol. 10, No. 4, pp. 423-438, 2004.

[GAR99] J. Garcia-Luna-Aceves and E. Madruga, "A Multicast Routing Protocol for Ad Hoc Networks", *Proc. INFOCOM*, 1999.

[GAR02] V.K. Garg, *Wireless Network Evolution: 2G to 3G*, NJ: Prentice Hall, 2002.

[GEN00] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, New York: Wiley, 2000.

[GEO97] N.R. George and B. Ilia, "Multicast Routing with End-to-end Delay and Delay Variation Constraints", *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 346-356, 1997.

[GER95] M. Gerla and J.T.C. Tsai, "Multicluster, Mobile, Multimedia Radio Network", *Wireless Networks*, Vol. 1, No. 3, pp. 255-265, 1995.

[GER03] M. Gerla, K.X. Xu, and X.Y. Hong, "Exploiting Mobility in Large Scale Ad Hoc Wireless Networks", *Proc. IEEE Computer Communication Workshop*, 2003.

[GIO03] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position-based Routing Algorithms for Ad Hoc Networks: A Taxonomy," in *Ad Hoc Wireless Networking*, Norwell, MA: Kluwer, 2003.

[GLO89] F. Glover, "Tabu search: Part I", *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206, 1989.

[GRE01] V. C. Gregory, K. Idit, and V. Roman, "Group Communication Specification: A Comprehensive Study", *ACM Computing Surveys*, Vol. 33, No. 4, pp. 427-469, 2001.

[HAB02] Y. Habib, A.-M. Abdulaziz, M.S. Sadiq, and A.T. Muhammad, “QoS-driven Multicast Tree Generation using Tabu Search”, *Computer Communications*, Vol. 25, No. 11-12, pp. 1140-1149, 2002.

[HAS99] Z.J. Haas and B. Liang, “Ad Hoc Mobility Management with Uniform Quorum Systems”, *IEEE/ACM Transactions on Networking*, Vol. 7, No. 2, pp. 228-240, 1999.

[HAS01] Z.J. Hass and M.R. Pearlman, “The Performance of Query Control Schemes for the Zone Routing Protocol”, *IEEE/ACM Transactions on Networking*, Vol. 9, No. 4, pp. 427-438, 2001.

[HEL00] C.S. Helvig, G. Robins, and A. Zelikovsky, “An Improved Approximation Scheme for the Group Steiner Problem”, *Networks*, Vol. 37, No. 1, pp. 8-20, 2000.

[HIG01] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing”, *IEEE Computer*, Vol. 34, No. 8, pp. 57-66, 2001.

[HON99] X.Y. Hong, M. Gerla, G.Y. Pei, and C.-C. Chiang, “A Group Mobility Model for Ad Hoc Wireless Networks”, *Proc. 2nd ACM MSWiM*, 1999.

[HSI04] P.-H. Hsiao and H.T. Kung, “Gravity Routing in Ad Hoc Networks: Integrating Geographical and Topology-based Routing”, *Proc. 7th International Symposium on Parallel Architectures, Algorithms and Networks*, 2004.

[HUA06] J.-L. Huang and M.-S. Chen, “On the Effect of Group Mobility to Data Replication in Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 5, pp. 492-507, 2006.

[HWA92] F.K. Hwang and D.S. Richards, “Steiner Tree Problems”, *Networks*, Vol. 22, No. 1. pp. 55-89, 1992.

[JAI01] R. Jain, A. Puri, and R. Sengupta, “Geographical Routing Using Partial Information for Wireless Ad Hoc Networks”, *Proc. INFOCOM*, 2001.

[JET01] J.G. Jetcheva and D.B. Johnson, “Adaptive Demand-Driven Multicast Routing in Multi-hop Wireless Ad Hoc Networks”, *Proc. MobiHoc*, 2001.

[JIA97] X.H. Jia, N. Pissinou, and K. Makki, “A Real-time Multicast Routing Algorithm for Multimedia Applications”, *Computer Communications*, Vol. 20, No. 12, pp. 1098-1106, 1997.

[JIA98] X.H. Jia, “A Distributed Algorithm of Delay-bounded Multicast Routing for Multimedia Applications in Wide Area Networks”, *IEEE/ACM Trans. Networking*, Vol. 6, No. 6, pp. 828-837, 1998.

[JOH96] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad-hoc Wireless Networks", in T. Imielinski and H. Korth, ed., *Mobile Computing*, pp. 153-181, Kluwer Academic Publishers, 1996.

[KAP93] A. Kapsalis, V.J. Rayward-Smith, and G.D. Smith, "Solving the Graphical Steiner Tree Problem Using Genetic Algorithm", *Journal of the Operational Research Society*, Vol. 44, No. 4, pp. 397-406, 1993.

[KAR00] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proc. MobiCom*, 2000.

[KHU95] S. Khuller, B. Raghavachari, and N. Young, "Balancing Minimum Spanning and Shortest Path Trees", *Algorithmica*, Vol. 14, No. 4, pp. 305-21, 1995.

[KIM04a] M.S. Kim, Y.-C. Bang, and H.S. Choo, "Advanced Multicasting for DVBMT Solution", *Proc. the 4th International Conference on Computational Science*, 2004.

[KIM04b] M.S. Kim, Y.-C. Bang, and H.S. Choo, "On Core Selection Algorithm for Reducing Delay Variation of Many-to-many Multicasts with Delay-bounds", *Proc. IFIP Networking*, 2004.

[KNO00] J.D. Knowles and D.W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy", *Evolutionary Computation*, Vol. 8, No. 2, pp. 149-172, 2000.

[KO98] Y.-B. Ko and N.H. Vaidya, "Location-Aided Routing in Mobile Ad-Hoc Networks", *Proc. MobiCom*, 1998.

[KO00] Y.-B. Ko and N.H. Vaidya, "GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks", *Proc. IC NP*, 2000.

[KOM05] O. Komolafe and J. Sventek, "RSVP Performance Evaluation using Multi-Objective Evolutionary Optimisation", *Proc. INFOCOM*, 2005.

[KUI02] F. Kuipers and P.V. Mieghem, "MAMCRA: A Constrained-based Multicast Routing Algorithm", *Computer Communications*, Vol. 25, No. 8, pp. 802-811, 2002.

[LAU01] M. Laumanns, http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2_c_source.html, 2001.

[LAW05] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "Fireworks: An Adaptive Group Communications Protocol for Mobile Ad Hoc Networks", *Proc. IFIP Networking*, 2005.

[LEE00] S. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols", *Proc. INFOCOM*, 2000.

[LEE05] Y.-J. Lee and M. Atiquzzaman, "Least Cost Heuristic for the Delay-constrained Capacitated Minimum Spanning Tree Problem", *Computer Communications*, Vol. 28, No. 11, pp. 1371-1379, 2005.

[LEU98] Y. Leung, G. Li, and Z.B. Xu, "A Genetic Algorithm for the Multiple Destination Routing Problems", *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 4, pp. 150-161, 1998.

[LEV04] T.V. Levanova and M.A. Loresh, "Algorithms of Ant System and Simulated Annealing for the p-median Problem", *Discrete Optimization*, Vol. 65, No. 3, pp. 431-438, 2004.

[LI00] J.Y. Li, J. Jannotti, D.S.D Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", *Proc. MobiCom*, 2000.

[LI02] B.C. Li, "On Increasing Service Accessibility and Efficiency in Wireless Ad-Hoc Networks with Group Mobility", *Wireless Personal Communications*, Vol. 21, No. 1, pp. 105-123, 2002.

[LI03] B.C. Li and K.H. Wang, "NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility", *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, pp. 1627-1641, 2003.

[LI04] L. Li and C. Li, "A QoS-guaranteed Multicast Routing Protocol", *Computer Communications*, Vol. 27, No. 1, pp. 59-69, 2004.

[LIM04] S.H. Lim, W.-C. Lee, G.H. Cao, and C.R. Das, "Performance Comparison of Cache Invalidation Strategies for Internet-based Mobile Ad Hoc Networks", *Proc. 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (MASS), 2004.

[LIM05] M. Lim *et al.*, "Landmark Guided Forwarding", *Proc. ICNP*, 2005.

[LIM06] S.H. Lim, W.-C. Lee, G.H. Cao, and C.R. Das, "A Novel Caching Scheme for Improving Internet-based Mobile Ad Hoc Networks Performance", *Ad Hoc Networks*, Vol. 4, No. 2, pp. 225-239, 2006.

[LIM07] S.H. Lim, W.-C. Lee, G.H. Cao, and C.R. Das, "Cache Invalidation Strategies for Internet-based Mobile Ad Hoc Networks", *Computer Communications*, Vol. 30, No. 8, pp. 1854-1869, 2007.

[LOW00] C.P. Low and Y.J. Lee, "Distributed Multicast Routing with End-to-end Delay and Delay Variation Constraints", *Computer Communications*, Vol. 23, No. 9, pp. 848-862, 2000.

[MAR99] J.-N. Mario and I-T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1415-1424, 1999.

[MAU01] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-based Routing in Mobile Ad Hoc Networks", *IEEE Networks*, Vol. 15, No. 6, pp. 30-39, 2001.

[MIC99] G. Michel, L. Jean-Francois, and S. Brunilde, "A Tabu Search Heuristic for the Steiner Tree Problem", *Networks*, Vol. 34, No. 2, pp. 162-172, 1999.

[MIS04] A.R. Mishra, *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G*, NJ: John Wiley & Sons, 2004.

[MOB04] P. Mohapatra, C. Gui, and J Li, "Group Communications in Mobile Ad hoc Networks", *IEEE Computer*, Vol. 37, No. 2, pp. 52-59, 2004.

[MUR01] F. Muratore ed., *UMTS: Mobile Communications for the Future*, NY: John Wiley, 2001.

[MUR04] C.S.R. Murthy and B.S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice Hall PTR, 2004.

[NAR00] P. Narvaez, K.-Y. Siu, and H.-Y. Tzeng, "New Dynamic Algorithms for Shortest Path Tree Computation", *IEEE/ACM Transactions on Networking*, Vol. 8, No. 6, pp. 734-746, 2000.

[ORD05] A. Orda and A. Sprintson, "A Scalable Approach to the Partition of QoS Requirements in Unicast and Multicast", *IEEE/ACM Transactions on Networking*, Vol. 13, No. 5, pp. 1146 - 1159, 2005.

[PAH03] A. Pahwa, S. Chavali, and S. Das, "Intelligent Computational Methods for Power Systems Optimization Problems", *Proc. IEEE Power Engineering Society General Meeting*, 2003.

[PAP98] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, NY: Dover Publications Inc., 1998.

[PAR98] M. Parsa, Q. Zhu, and J.J. Garcia-Luna-Aceves, "An Iterative Algorithm for Delay-constrained Minimum-cost Multicasting", *IEEE/ACM Trans. Networking*, Vol. 6, No. 4, pp. 461-474, 1998.

[PEI00] G.Y. Pei, M. Gerla, and X.Y. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility", *Proc. MobiHoc*, 2000.

[PER94] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proc. ACM SIGCOMM*, 1994.

[PER99] C.E. Perkins and E.M. Royer, "Ad hoc On-Demand Distance Vector Routing", *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[PER01] C.E. Perkins, *Ad Hoc Networking*, Addison Wesley, 2001.

[PRA98] R. Prakash and R. Baldoni, "Architecture for Group Communication in Mobile Systems", *Proc. 17th IEEE SRDS*, 1998.

[RAO03] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information", *Proc. Mobicom*, 2003.

[ROB00] G. Robins and A. Zelikovsky, "Improved Steiner Tree Approximation in Graphs", *Proc. ACM/SIAM Symposium on Discrete Algorithms*, 2000

[ROB04] K. Robert, *Wireless Network Coexistence*, New York: McGraw-Hill, 2004.

[ROY99] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol", *Proc. MobiCom*, 1999.

[ROY04] A. Roy and S.K. Das, "QM^2RP: A QoS-based Mobile Multicast Routing Protocol Using Multi-objective Genetic Algorithm", *Wireless Networks*, Vol. 10, No. 3, pp. 271-286, 2004.

[SAM04] P. Samar, M.R. Pearlman, and Z.J. Haas, "Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 4, pp. 595-608, 2004.

[SHE99] P.R. Sheu and S.T. Chen, "On the Hardness of Approximating the Delay Variation Constraint Multicast Trees", *Proc. the Taiwan Computer Symposium*, 1999.

[SHE02] P.-R. Sheu and S.-T. Chen, "A Fast and Efficient Heuristic Algorithm for the Delay- and Delay Variation-bounded Multicast Tree Problem", *Computer Communications*, Vol. 25, No. 8, pp. 825-833, 2002.

[SIA05] S. Siachalou and L. Georgiadis, "Algorithms for Precomputing Constrained Widest Paths and Multicast Trees", *IEEE/ACM Transactions on Networking*, Vol. 13, No. 5, pp. 1174 - 1187, 2005.

[SIV05] S. Sivavakeesar and G. Pavlou, "Scalable Location Services for Hierarchically Organized Mobile Ad hoc Networks", *Proc. MobiHoc*, 2005.

[SRI95] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, Vol. 2, No. 3, pp. 221-248, 1995.

[STO99] I. Stojmenovic, "A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks", *SITE, University of Ottawa, Tech. Rep. TR-99-09*, 1999.

[STO01] I. Stojmenovic and L. Xu, "Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 10, pp. 1023-1032, 2001.

[THA05] M.T. Thai, Y.S. Li, and D.-Z. Du, "A Combination of Wireless Multicast Advantage and Hitch-hiking", *IEEE Communications Letters*, Vol. 9, No. 12, pp. 1037-1039, 2005.

[TOH02] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall PTR, 2002.

[TOJ04] T. Tojo, T. Enokido, and M. Takizawa, "Notification-based QoS Control Protocol for Multimedia Group Communication in High-Speed Networks", *Proc. ICDCS*, 2004.

[TSE03] Y.-C. Tseng, C.-C. Shen, and W.-T. Chen, "Integrating Mobile IP with Ad Hoc Networks", *IEEE Computer*, Vol. 36, No. 5, pp. 48-55, 2003.

[TUR02] D. Turgut, S. K. Das, R. Elmasri, B. Turgut, "Optimizing Clustering Algorithm in Mobile Ad Hoc Networks Using Genetic Algorithmic Approach", *Proc. GlobeCom*, 2002.

[TUR03] D. Turgut, B. Turgut, R. Elmasri, T.V. Le, "Optimizing Clustering Algorithm in Mobile Ad Hoc Networks Using Simulated Annealing", *Proc. WCNC*, 2003.

[TYA03] H.-Y. Tyan, J.C. Hou, and B. Wang, "Many-to-many Multicast Routing with Temporal Quality of Service Guarantees", *IEEE Transactions on Computers*, Vol. 52, No. 6, pp. 826-832, 2003.

[VAL05] A.C. Valera, W.K.G. Seah, and S.V. Rao, "Improving Protocol Robustness in Ad Hoc Networks through Cooperative Packet Caching and Shortest Multipath Routing", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, pp. 443-457, 2005.

[VIS06] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring Mesh and Tree-Based Multicast Routing Protocols for MANETs", *IEEE Transactions on Mobile Computing*, Vol. 5, No. 1, pp. 28-42, 2006.

[WAN00] B. Wang and J.C. Hou, "A Survey on Multicast Routing and Its QoS Extensions: Problems, Algorithms, and Protocols", *IEEE Network*, Vol. 14, No. 1, pp. 22-36, 2000.

[WAN02] K.H. Wang and B.c. Li, "Group Mobility and Partition Prediction in Wireless Ad-Hoc Networks", *Proc. ICC*, 2002.

[WAN06] X.W. Wang, J.N. Cao, H. Cheng, and M. Huang, "QoS Multicast Routing for Multimedia Group Communications Using Intelligent Computational Methods", *Computer Communications*, Vol. 29, No. 12, pp. 2217-2229, 2006.

[WAX88] B.M. Waxman, "Routing of Multipoint Connections", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, pp. 1617-1622, 1988.

[WOL04] K. Wolfgang, F. Holger, and W. Jorg, "Hierarchical Location Service for Mobile Ad-Hoc Networks", *ACM Mobile Computing and Communications Review*, Vol. 8, No. 4, pp. 47-58, 2004.

[WU05a] X.X. Wu, "VPDS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks", *Proc. ICDCS*, 2005.

[WU05b] X.X. Wu and B. Bhargava, "AO2P: Ad hoc On-demand Position-based Private Routing Protocol", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 4, pp. 335-348, 2005.

[WU06] S.B. Wu, K.S. Candan, "GMP: Distributed Geographic Multicast Routing in Wireless Sensor Networks", *Proc. ICDCS*, 2006.

[XIA99] X. Xiao and L.M. Ni, "Internet QoS: A Big Picture", *IEEE Network*, Vol. 13, No. 2, pp. 8-18, 1999.

[XIA06] X.J. Xiang, X. Wang, and Z.H. Zhou, "An Efficient Geographic Multicast Protocol for Mobile Ad Hoc Networks", *Proc. 7th International Symposium on World of Wireless, Mobile and Multimedia Networks*, 2006.

[XIN04] Y.F. Xin and G.N. Rouskas, "Multicast Routing under Optical Layer Constraints", *Proc. INFOCOM*, 2004.

[XUE01] Y. Xue and B.C. Li, "A Scalable Location Management Scheme in Mobile Ad-hoc Networks", *Proc. IEEE Local Computer Networks*, 2001.

[XUE03] G.L. Xue, "Minimum-cost QoS Multicast and Unicast Routing in Communication Networks", *IEEE Transactions on Communications*, Vol. 51, No. 5, pp. 817-824, 2003.

[YAN01] D.-N. Yang, W.J. Liao, and Y.-T. Lin, "MQ: An Integrated Mechanism for Multimedia Multicasting", *IEEE Transactions on Multimedia*, Vol. 3, No.1, pp. 82-97, 2001.

[YAN05] S. Yang and J. Wu, "New Technologies of Multicasting in MANET", in *Design and Analysis of Wireless Networks*, Nova Science Publishers, 2005.

[YU05] Y. Yu and H.J. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", *IEEE Communications Surveys & Tutorials*, Vol. 7, No. 1, pp. 32-48, 2005.

[ZAP04] D. Zappala, "Alternate Path Routing for Multicast", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 1, pp. 30-43, 2004.

[ZEN98] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks", *Proc. the 12th Workshop on Parallel and Distributed Simulations*, 1998.

[ZHA99] Q. Zhang and Y.-W. Leung, "An Orthogonal Genetic Algorithm for Multimedia Multicast Routing", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 1, pp. 53-62, 1999.

[ZHA04] X. Zhang and L. Jacob, "MZRP: An Extension of the Zone Routing Protocol for Multicasting in MANETs", *Journal of Information Science and Engineering*, Vol. 20, No. 3, pp. 535-551, 2004.

[ZHO05] B. Zhou, *et al*., "GeoLANMAR: Geo Assisted Landmark Routing for Scalable, Group Motion Wireless Ad Hoc Networks", *Proc. 61st IEEE Semiannual Vehicular Technology Conference* (VTC2005-Spring), 2005.

[ZIT02] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", in *Evolutionary Method for Design, Optimisation and Control*, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.), 2002.