Scalable routing in mobile ad hoc networks

Yang, Peng

2005

Yang, P. (2005). Scalable routing in mobile ad hoc networks. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2547>

<https://doi.org/10.32657/10356/2547>

Nanyang Technological University

Downloaded on 26 Apr 2023 19:45:26 SGT

Scalable Routing in Mobile Ad Hoc Networks

Yang Peng

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2005

TK
5105.543
Y22
2005

Abstract

With the technology advances in wireless network, a large number of mobile devices for communication are now publicly available. However, the traditional wireless networks are based on infrastructure modes with some administration in the system. Deployment of such networks takes time and cannot be set up in time of emergency. Therefore, mobile multi-hop radio networks, also called mobile ad hoc networks(MANETs), play a critical role in places where infrastructure is neither available nor economical to build. However MANETs are usually limited in terms of bandwidth, energy, processing capabilities, and security. These problems result in some interesting issues. Scalability issue is an attractive one among them. In this thesis, we focus on the scalability study on the routing in MANETS. We analyze the existing routing approaches in MANETs, we conclude that clustering and hierarchical routing are promising ways to improve the scalability of the routing based on the analysis. We propose an adaptive clustering scheme (RMACP: adaptive&cached cluster-based routing protocol) to reduce the maintenance overhead for clustering schemes. Simulation results show that without any routing performance degradation the RMACP can achieve much lower control overhead compared with other existing clustering techniques.

In addition we describe a novel hierarchical approach for MANETs. We provide a thorough and detailed view of the hierarchical maintenance for hierarchical routing. Over the previous hierarchical routing approaches, our scheme has the following advantages.

- Forwarding pointers are used in the maintenance of our hierarchical system to reduce maintenance overhead.
- In our hierarchical system a random clustering based hierarchical routing algorithm(RCHR) is proposed to overcome the convergence problems found by traditional hierarchical approaches.

- Instead of broadcasting in multiple hops for traditional hierarchical approaches, broadcasting in one hop and unicasting over multiple hops are used in our hierarchical system to reduce the maintenance overhead.
- Our hierarchical scheme can increase hierarchies along with the expansion of network size adaptively.

We use simulation to compare our hierarchical approach with other categories of routing algorithms proposed in literature. The results show that our hierarchical approach scale better than others when the network size increases. On the other hand, our approach achieves lower overhead without imposing delivery ratio for large network.

In summary, this thesis provides some insight into possible new designs of scalable routing approaches for MANETs.

Acknowledgments

I would like to express my profound gratitude to my supervisor Dr. Yow Kin Choong for his support since the beginning, and for his patience and guidance. I am especially grateful to Dr. Sukumar Nandi for his supervision, guidance, friendship, and moral support. Without their continuous encouragement and advice, I could never complete this thesis.

I would like to thank all my friends and colleagues in the Center for Multimedia Networking Technology(CeMNet), especially Wang Huiqing, Zhou Zhi, Ye Minghua, Xie Jiawei and Li Zhifei, with whom I have worked together, for their helpful discussions and friendship. I would like to thank my parents for their love, understanding and support.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Objective	3
1.4 Contributions	4
1.4.1 Passive Cached Routing and Adaptive Clustering	4
1.4.2 A random clustering based hierarchical routing algorithm	5
2 Literature Review	6
2.1 Routing Issues in MANETs	6
2.2 Classification of Routing Protocols in MANETs	7
2.3 Problems of Scalable Routing	8
2.4 Scalability of Existing Routing Protocols	9
2.5 Overview of Hierarchical Routing Protocols: Clustering and Maintenance	11
2.6 Problems of Hierarchical Routing Protocols	16
3 Improving Cluster-based Routing on Caching & Adaptivity	18
3.1 Existing Clustering Techniques	18
3.1.1 Highest-degree Algorithm(HID)	19
3.1.2 Lowest-id Algorithm(LID)	20

3.1.3	Least Clustering Change Algorithm(LCC)	20
3.1.4	Random Competition Based Algorithm(RCC)	21
3.1.5	Mobility Based Algorithm(MOBIC)	22
3.1.6	Design Principles of Clustering Techniques	23
3.2	Passive Cached Cluster-based Routing for MANETs	24
3.2.1	Route caching	25
3.2.2	Passive cached Method	25
3.2.3	Simulation results and discussion	29
3.3	Adaptive Clustering for MANETs	32
3.3.1	Adaptive System	32
3.3.2	An adaptive system for clustering routing	33
3.3.3	Simulation Results and Discussion	35
3.4	An Improved Cluster-based Routing Algorithm based on Combination of RCC & MOBIC & Caching & Adaptivity	39
3.4.1	Simulation and Comparison	41
3.5	Summary	44
4	A Random Clustering Based Hierarchical Routing Algorithm	45
4.1	Clustering technique for hierarchical strategy	45
4.1.1	Level status in hierarchical structure	46
4.1.2	Cluster tables	46
4.1.3	One-level hierarchical clusters	49
4.1.4	Multi level hierarchical clusters	52
4.2	Route Discovery	58
4.2.1	Route Requests	59
4.2.2	Route Replies	62
4.3	Data Delivery	63
4.4	Maintenance	64
4.4.1	Maintaining hierarchical structure with forwarding pointers	64
4.4.2	Route Error Messages	66
4.5	Performance Evaluations	67

- 4.6 Comparison with other routing protocols 72
 - 4.6.1 Comparison with Flat Routing Protocols 72
 - 4.6.2 Comparison with Hierarchical Routing Protocols 78
 - 4.6.3 Comparison with Location Based Routing Protocols 81
- 4.7 Summary 82
- 5 Conclusion and Future Work 83**
 - 5.1 Conclusion&Contribution 83
 - 5.2 Future Work 84
 - 5.2.1 Mobility Management and Clustering 85
 - 5.2.2 Combination of Position-based Schemes with Clustering Techniques 86
 - 5.2.3 Load Balance of Hierarchical Protocols 87
 - 5.2.4 Using Different Power Level at Different Levels 89
 - 5.2.5 Simulation with Different Network Size and Different Mobility Models 89
- References 91**

List of Figures

2.1	CGSR Routing	13
2.2	HSR: multi-level structure	14
3.1	Decision after a route request is received	27
3.2	Control Overhead Vs Network Size	31
3.3	Delivery Ratio Vs Network Size	31
3.4	End-to-end Delay Vs Network Size	32
3.5	Illustration of the adaptive algorithm	34
3.6	Control overhead Vs. Network Size	37
3.7	Delivery ratio Vs. Network Size	37
3.8	End-to-end delay Vs. Network Size	38
3.9	Control Overhead vs. Mobility	42
3.10	Delay vs. Mobility	42
3.11	Delivery ratio vs. Mobility	43
4.1	Hierarchical structure from level-0 to level-3	48
4.2	Update procedure upon receiving a Hello message	53
4.3	Level-0 hello message of node 5	55
4.4	Level-1 hello message of node 5	55
4.5	Hierarchical structure and hierarchical tree	57
4.6	Convergence time taken to establish a hierarchical system	68
4.7	Number of nodes at each level	69
4.8	Convergence time of three clustering algorithms	71
4.9	Number of Employed Forwarding Pointer	72
4.10	Delivery Ratio vs. Network Size (Low Mobility=2m/s)	73

4.11 Control Packets vs. Network Size (Low Mobility=2m/s) 74

4.12 Delay vs. Network Size (Low Mobility=2m/s) 74

4.13 Delivery Ratio vs. Network Size (High Mobility=10m/s) 76

4.14 Control Packets vs. Network Size (High Mobility=10m/s) 76

4.15 Delay vs. Network Size (High Mobility=10m/s) 78

4.16 Control Overhead vs. Network Size (Mobility=10m/s) 79

4.17 Delivery Ratio vs. Network Size (Mobility=10m/s) 79

4.18 Delay vs. Network Size (Mobility=10m/s) 80

5.1 Location based clustering approach 88

List of Tables

3.1	Simulation Parameters	29
3.2	Simulation Parameters	35
3.3	Simulation parameters	43
4.1	level-0 neighbor table for node 2	48
4.2	level-0 neighboring cluster table for node 2	48
4.3	level-1 neighbor table for node 2	48
4.4	level-1 neighboring cluster table for node 2	48
4.5	level-2 neighbor table for node 2	48
4.6	Simulation parameters	73

Chapter 1

Introduction

This thesis discusses the scalability issues of routing protocols in ad hoc networks and details the specification, design and implementation of hierarchical routing protocols.

In this chapter, the background and challenges of ad hoc networks are first described in section 1.1; the research motivation is then presented in section 1.2; and finally, our contributions are highlighted in section 1.4.

1.1 Background

Wireless networks have existed for many decades. With the development of technology in both wireless devices and applications mobile wireless networking has become popular recently. A large number of mobile devices [EBS03] for communication are now publicly available. The traditional wireless networks are based on infrastructure modes with some administration in the system, implying that networks are fixed and constrained to a geographical area with predefined boundary. However, deployment of such networks takes time and cannot be set up in times of emergency. Therefore, mobile multi-hop radio networks, also called mobile ad hoc networks (MANETs), play a critical role in places where infrastructure is neither available nor economical to be built.

A wireless ad hoc network is composed of autonomous nodes or terminals that communicate with each other by maintaining multihop radio connection in a decentralized

CHAPTER 1. INTRODUCTION

manner. Each node in a wireless ad hoc network functions as both a host and a router, and the control of the network is distributed among the nodes. Since the nodes communicate over wireless links, they have to tolerate the effects of radio communication such as noise, fading, and interference. In addition, the wireless links typically have less bandwidth than in a wired network. The network topology is also dynamic, where nodes move randomly and the topology changes unexpectedly. Since the connectivity among the nodes may vary with time due to node mobility, the transmission will be affected due to the broken links. Some of aforementioned features characterize ad hoc networks and introduce open problems for research in relative areas.

There are many challenges in the creation of an ad hoc network [BC04]: routing challenges, wireless mediums access challenges [BC98] [LTG98] [Bia00], security challenges [PH02] [PH03] and portability challenges(e.g., lower power [KK03] [LPJW01] [JSAC01] and smaller storage capacity). In this thesis, we focus on the routing issue in MANETs.

1.2 Motivation

Mobile wireless networks have many unique characteristics that make traditional routing protocols inapplicable to them. The topology of a mobile network is often highly dynamic due to the mobile nature of the nodes. Hence a broken link in a wireless network is likely to occur more frequently than in a wired network. Additional characteristics of mobile wireless networks include limited resource (power and bandwidth), and high error rates due to the wireless device. These features limit the communication between nodes and make routing in MANETs extremely difficult. Also routing performance is constrained by limited resources and unreliable environment. Furthermore, routing may incur exhaustion of these resources when the offered load increases and the network size expands. Therefore, more efficient routing protocols must be designed to save the limited resources and enhance the routing scalability for MANETs.

Our research goals is aimed at improving the scalability and flexibility of routing in MANETs. More specifically, it is necessary to embed various adaptive mechanisms to make these systems self-adaptive and more efficient. The utilization of limited resources should also be optimized. The performance of routing protocols should not degrade much when the parameters (e.g. mobility, network size, etc) of the network change. This work should enable MANET routing protocols to be more flexible and more scalable.

1.3 Objective

With the characteristics of limited bandwidth, frequent unpredictable changing of topology and non-centralized structure, it is very difficult to manage the entire framework of MANETs. As wireless communication channels are highly affected by unpredictable factors such as interference, propagation loss and multipath fading, it is necessary to embed mechanisms to make these systems self-adaptive and more efficient. Our objective is to optimize bandwidth resource utilization in large networks because control messages have to be transmitted globally throughout the network.

The basic goals of a routing system to achieve scalability are as follows [Yu00]:

- Stability [GWFM02]:

The performance of the routing system will not degrade upon the expansion of the network with limited resources;

- Redundancy and robustness:

Redundancy imposes the need for as many alternative paths as possible for each route, which increases route processing and storage burdens. Redundancy also increases adjacencies to the network topology; robustness is improved as a result.

- Reasonable convergency time:

Change of the routing states in the node will be propagated into the whole network within reasonable time.

CHAPTER 1. INTRODUCTION

- Routing information integrity:

Encapsulation of routing policy and service requirement are preferred to make the routing information more compact and easy to be processed.

- Sensible and manageable routing policy:

Routing Policy refers to the administrative criteria for handling routing information, commonly in the form of routing path selection and route filtering. The routing policy should be intelligent and easy to be managed.

1.4 Contributions

In this thesis, we have made the following contributions:

1.4.1 Passive Cached Routing and Adaptive Clustering

We propose a passive caching mechanism used for cluster-based routing protocols in mobile ad hoc networks (MANETs). It utilizes the backbone [KKRM01] [LH00] of the network to gather and reuse discovered routing information more efficiently and effectively. We achieve three goals through this algorithm:

- utilize the backbone to centralize the routing information of the MANET, which makes the routing information more easily manageable and be accessed more efficiently
- obtain routing information without incurring any additional overhead
- reuse the pre-discovered routes in the cache and reduce routing overhead caused by route discoveries

In addition we propose an adaptive system based on existing clustering routing algorithms. Instead of using fixed broadcast interval for clustering maintenance we employ the

node mobility as feedback to tune the broadcasting interval of each node. Consequently, hello messages broadcasted for clustering maintenance are generated more efficiently. As a result, the control overhead is reduced and the bandwidth is reused more effectively.

1.4.2 A random clustering based hierarchical routing algorithm

A hierarchical system is a good solution to provide scalability in large networking systems. In this thesis we present a random clustering based hierarchical routing algorithm(RCHR) and describe the clustering and the maintenance of the hierarchical algorithm in detail. With RCHR the problems faced by traditional hierarchical approaches are overcome. In order to achieve good convergence of clustering, we make use of random processing to partition the network into clusters. Multiple hierarchies are set up through broadcasting in one hop and unicasting within multiple hops. Forwarding pointers are used in maintenance in order to distribute communication tasks in the hierarchical system and reduce communication overhead. Employment of forwarding pointers also stabilize the hierarchical structure and reduces the high cost required to update hierarchical addresses from low levels to high levels. As a result, routing tasks are distributed more evenly and communication overhead of critical nodes in high levels is reduced. Furthermore, the hierarchical scheme can increase the number of levels with the expansion of the network size adaptively. The new algorithm also provides some insight into new possible designs of hierarchical routing protocols.

The work above has been accepted by two conference for oral presentation: ICICS(the Fourth International Conference on Information, Communications & Signal Processing)on December 15, 2003 and ICTTA (the First International Conference on Information & Communication Technologies: from Theory to Applications) on April 19, 2004.

Chapter 2

Literature Review

2.1 Routing Issues in MANETs

An ad hoc network is a collection of wireless mobile nodes forming a temporary network without any centralized administration. Due to the limited propagation range of each node, a node is not able to connect all the nodes within the network. Establishing a multihop connection is required for routing in MANETs. Some previous routing schemes have been developed for wired networks, however, these schemes are proposed for routing in single hop. Some routing schemes are required to deal with multihop communication in MANETs. Thus, routing schemes in MANETs are concerned. The growing interest in Mobile Ad Hoc Network techniques has resulted in many routing issues. Currently there are many attractive issues about MANET routing protocols, the following is a list of them [Yu00].

- Loop-freedom [GLA93]: Not necessary but desirable.
- Multicast support [TDLR02]: Construct paths along which data packets from a source are distributed to reach many, but not all, destination in a network.
- Security: Sufficient security protection to prohibit disruption of modification or protocol operation is desired.

CHAPTER 2. LITERATURE REVIEW

- Unidirectional link support: If only unidirectional links exist, protocols should be able to use them.
- QoS routing [TCG97] [MCE98] [CML99]: Effectively control the traffic to provide the QoS requirements for each admitted connection.
- Scalability [BJKM04]: The routing overhead increases rapidly when the network becomes larger. Since the bandwidth is scarce and the resource is limited in MANET nodes, some schemes should be considered to make protocols more scalable.

2.2 Classification of Routing Protocols in MANETs

Currently routing protocols for MANETs fall into three categories according to their control patterns [HXG02]: flat routing protocols, hierarchical routing protocols and location based routing protocols.

- Flat routing protocols are characterized by having the same functionality for each node within the network. Flat routing protocols include proactive routing protocols (e.g., DSDV [PB94], GSR [ICP⁺99], TBRPF [BOT01]), reactive routing protocols (e.g., DSR [BJM03], AODV [PR99]) and hybrid routing protocols (e.g., ZRP [Haa97]). In proactive protocols each node maintains consistent and up-to-date routing information to all other nodes and routes can be obtained immediately when needed. On the other hand, in reactive protocols, routes are discovered only on demand and maintained only as needed. Hybrid protocols are the combination of proactive protocols and reactive protocols.
- In hierarchical routing protocols, nodes with different status have different functionalities. Nodes are grouped into clusters locally. A hierarchical system is built based on clusters for the network and the nodes at high hierarchy manage tasks of the

CHAPTER 2. LITERATURE REVIEW

nodes at low hierarchy. Some hierarchical routing protocols have been proposed, such as CGSR [CG97], HSR [ICP⁺99] and FSR [ICP⁺99].

- In location-based routing protocols, location information is used for directional routing. Geographic forwarding is applied to improve the routing performance. A lot of location-based schemes are developed for MANETs, such as LAR [KV98], DREAM [BCSW98], GPSR and [KK00].

2.3 Problems of Scalable Routing

Due to the fact that nodes in MANETs share the same access wireless channel, they cooperate to engage in multiple-hop forwarding. Each node functions not only as a host but also as a router that maintains routes to forward data packets for other nodes in the network that may not be within direct transmission range. Routing in MANETs faces extreme challenges in terms of node mobility/dynamics, potentially very large number of nodes, and limited communication resources(e.g.,bandwidth and energy). Thus, not only is maintaining and acquiring routing information in MANETs difficult to achieve, it is also difficult to achieve this in a manner that scales well with the increasing network size. The routing protocols for MANETs have to adapt quickly to frequent and unpredictable topology changes and must be parsimonious of communications and processing resources [HXG02].

Scalable characteristic for a networking system indicates that the performance(e.g., throughput, delivery ratio, transmission delay, storage overhead) of the system will not degrade much when network size increases.

Currently there are three main problems for scalable routing in ad hoc networks:

- As the bandwidth is scarce in MANETs and that the population in a MANET is small, as compared to the wired Internet, the scalability issue for wireless multi-hop routing protocols is mostly concerned with excessive routing message overhead

CHAPTER 2. LITERATURE REVIEW

caused by the increase of network population and mobility [HXG02]. In large networks, the transmission of routing information will ultimately consume most of the bandwidth and consequently block applications. Thus, reducing routing control overhead becomes a key issue in achieving routing scalability.

- The size of the routing table is also a concern in MANETs because large routing tables imply large control overhead generally using either distance-vector or link-state routing algorithms [Kes97].
- The scalability is more challenging in the presence of both large numbers and mobility. If nodes are stationary, the large population can be effectively handled with conventional hierarchical routing. In contrast, when nodes move, the hierarchical partitioning must be continuously updated. The scheme is concerned with the ability to make the partition more stable and flexible [HXG02].

These problems impose a direct challenge to the routing scalability of the MANET. The routing design for a large-scale network should achieve the basic goals of accuracy, stability, redundancy and convergence as described in section 1.2 so as to achieve them in a scalable fashion.

With the characteristics of limited bandwidth, frequent unpredictable changing of topology and a non-centralized structure, it is very difficult to manage the entire framework of MANETs. Our objective is to achieve scalable routing in MANETs by reusing resources and reducing communication overhead. The bandwidth resource should be utilized in an optimal way.

2.4 Scalability of Existing Routing Protocols

In this section, the scalable problems of different type of routing protocols are addressed.

CHAPTER 2. LITERATURE REVIEW

In proactive protocols each node maintains routing tables containing entries for all the nodes in the network. To keep these routing entries fresh in every node, each node periodically broadcast its own view of the network using flooding [LTG99]. This mechanism is acceptable if the population of the network is small. However, storage overhead and control overhead caused by table broadcasting and updating will increase rapidly when network size increases. Thus, proactive protocols do not scale well for large networks.

In reactive protocols each route requires the flooding of route requests over the entire network. Flooding is a reliable way to send packets when the location of the destination is not known. However, flooding covers the entire network, and every node and every channel will contribute a large amount of resource to it. Hence, flooding incurs exhaustion of these resources when the offered load increases and the network size expands. Therefore, reactive protocols also do not scale well for networks with large population and heavy traffic load.

Since hybrid protocols are the combination of proactive protocols and reactive protocols. They face scalability problems of both proactive protocols and reactive protocols.

The advances in the development of Global Positioning System benefits location-based routing in ad hoc networks by guiding packet forwarding depending on the location information. Geographic forwarding is provided to reduce processing overhead caused by flooding. Research [KK00] [RT99] [KSU99] [BCS99] [TCN02] has shown that location information can improve routing performance in ad hoc networks. However, considering the mechanism in a mobile environment, if location information is inaccurate by the time the information is used, the network can not be controlled well. On the other hand, when network size increases, more cost will be consumed to maintain global location information.

Typically, when the network size increases, current flat routing schemes can not deal well with large networks because of the storage and processing overhead. One way to

CHAPTER 2. LITERATURE REVIEW

solve this problem and to produce scalable and efficient solution is hierarchical routing. In hierarchical routing schemes nodes are grouped into clusters based on their relative proximity to one another. In turn, clusters are grouped into super-clusters, and so on. A hierarchical system is established based on clustering algorithms. A specific node is chosen as the clusterhead for each cluster. Clusterheads are responsible for the management of clusters, while other nodes only have a scope within clusters. In this way the network is organized hierarchically and routes are recorded hierarchically. When routing across clusters, a hierarchical structure increases robustness and reuse of routes. It can reuse much resources and reduce excessive overhead when network size increases. The performance of the network will not degrade much as a result. Furthermore, it is more flexible and easy for a large network to manage neighboring nodes within a hierarchy instead of over all the nodes.

2.5 Overview of Hierarchical Routing Protocols: Clustering and Maintenance

In this section we review some existing hierarchical schemes and address their problems.

Traditionally hierarchical schemes mainly include two parts: clustering and hierarchy maintenance.

Clustering focuses on organizing nodes into groups. Nodes are divided into clusters on their relative proximity to one another. Each cluster elects a node acting as clusterhead to rule its cluster. Automatically, clusterheads at a low level become members of the next higher level. These new members in turn organize themselves in super clusters, and so on. Consequently, a hierarchical structure is established and the partition of the network in each level of the hierarchical structure is completed by clustering.

When the formation of hierarchical structure is completed, the hierarchy maintenance deals with the communication and integration of different levels in the hierarchical structure. Due to the node movement, the topology of the network changes frequently, which

CHAPTER 2. LITERATURE REVIEW

incurs the reformation of clusters and reorganization of hierarchical structure. When a node moves from a cluster to another cluster, the location information of the mobile node is propagated to higher levels of the hierarchical structure until all the clusters that contain this node have the updated information.

Currently some typical hierarchical routing protocols have been proposed for MANETs, for example CGSR [CG97], HSR [KF97]. We shall review these two hierarchical routing protocols in the following description.

CGSR (Clusterhead-Gateway Switch Routing)

Clusterhead-Gateway Switch Routing (CGSR) [CG97] is a hierarchical routing protocol using two levels. A stable partition algorithm known as Least Clusterhead Change (LCC) [CG97] is used to partition the network into clusters and one clusterhead is elected in each cluster. After partitioning every node in the network will belong to at least one cluster. A node belonging to two or more clusters is a gateway which connects two neighboring clusters together. A node belonging only to one cluster is called an internal node. Data packets are routed alternatively between clusterheads and gateways. The procedure is demonstrated in Figure 2.1. A packet will be routed via $C_1, G_1, C_2, G_2, C_3 \dots$, where C_i are clusterheads and G_i are gateways, and finally reaches its destination's clusterhead.

CGSR modify the DSDV scheme by exploiting the clusterheads. Actually, two levels are established and hierarchical routing is used to route packets. Each node maintains two tables: a cluster member table and a routing table. The cluster member table records the destination clusterhead for each node, and broadcasts it periodically. A node will update its cluster member table when it receives a new member table from its neighbors. Cluster member table is used to map destination address to the destination clusterhead address. The routing table maintains one entry for each cluster and records the path to the corresponding clusterhead.

Figure 2.1: CGSR Routing

To find a path to the destination, a current node looks up the clusterhead of the destination from the cluster member table. Then, it consults its routing table to find the next hop to the destination's clusterhead. The destination clusterhead will finally route the packet to the destination node.

The major advantage of CGSR is that it can greatly reduce the routing table size compared to proactive protocols [PB94]. Only one identity (corresponding clusterhead) is needed for all the nodes in the same cluster. On the other hand, the robustness and utilization of routes increases by routing alternatively between clusterheads and gateways. These features make CGSR more scalable to large network size compared with proactive protocols. The drawback of CGSR is that each node still needs to store global information for each node on its corresponding clusterhead and routing information for all clusters in the network. The number of clusters increases when the network size increases, which results in producing more storage overhead for tables and more communication overhead for propagating and updating member tables. In addition, the LCC algorithm introduces additional overhead and complexity in the formation and maintenance of clusters in a mobile environment.

Figure 2.2: HSR: multi-level structure

HSR (Hierarchical State Routing)

Hierarchical State Routing (HSR) [ICP⁺99] is a multi-level, clustering based link state routing protocol. It maintains a hierarchical topology, where elected cluster heads at the lowest level become members of the next higher level. These new members in turn organize themselves in clusters and so on. The goal of clustering is to reduce routing overhead at each level. An example of a three level hierarchical structure is demonstrated in Figure 2.2.

Generally there are three kinds of nodes in a cluster, clusterheads(e.g., node 1, 2, 3, 4), gateways(e.g., node 6, 7, 8 and 11), and internal nodes(e.g., node 5, 9, and 10). A clusterhead acts as a local coordinator for transmissions within the cluster. At the first level each node monitors the state of link to each neighbor and broadcasts it within

CHAPTER 2. LITERATURE REVIEW

the cluster. The clusterhead summarizes link state information within its cluster and propagates it to the neighboring clusterheads. The information between neighboring clusterheads leads to the formation of level-2 clusters. For example, as in figure ??, neighboring clusterheads 3 and 4 become members of the level-2 cluster C_{2-3} . Link state entries at level-2 nodes contain the virtual links contained by the level-1 path. The virtual links can be viewed as a tunnel implemented through lower level nodes. For instance nodes 1 and 2 in C_{2-1} consists of path $\langle 1, 6, 2 \rangle$ from clusterhead 1 to clusterhead 2 through gateway node $\langle 6 \rangle$. After obtaining the link state information at one level, each virtual node floods it down to the nodes of the lower level clusters. As a result, each physical node has a hierarchical topology information through the hierarchical address of each node as opposed to a full topology view as in flat link state schemes.

In HSR, the Hierarchical ID (HID) of a node is defined as the sequence of the MAC addresses of the nodes in the path from the top hierarchy to the node itself. For example, in Figure 2.2 the hierarchical address of node 6, is $\langle 3, 2, 6 \rangle$. In this example, node 3 is a member of the top hierarchical cluster (level-2), and it is also the clusterhead of C_{1-3} and C_{2-3} . Node 2 is a member of C_{2-3} and is the clusterhead of C_{1-2} . Node 6 is a member of C_{1-2} and can be reached directly from node 2. The advantage of this hierarchical address scheme is that each node can dynamically and locally update its own HID upon receiving the routing updates from the nodes higher up in the hierarchy.

The hierarchical address is sufficient to deliver a packet to its destination from anywhere in the network. Referring to Figure 2.2, consider for example the delivery of a packet from node 5 to node 10. Note that $HID(5) = \langle 1, 1, 5 \rangle$ and $HID(10) = \langle 3, 3, 10 \rangle$. The packet is forwarded upwards to the top hierarchy by node 5. Node 1 delivers the packet to node 3, which is the top hierarchy node for destination 10. Node 1 has a virtual link, i.e. a tunnel, to node 3, namely, the path $(1, 6, 2, 8, 3)$. It thus delivers the packet to node 3 along this path. Finally, node 3 delivers the packet to node 10 along the downwards hierarchical path, which in this case reduces to a single hop.

CHAPTER 2. LITERATURE REVIEW

Under the logical partition and location management approach of HSR [ICP⁺99], each cluster is a subset and each subset is associated with a home agent. When a node moves in this subset it will register with its new home agent. The home agent HIDs are appended to the top level routing tables. All home agents advertise their HIDs to the top hierarchy. The top hierarchy has the global HIDs of all the nodes in the network. When a source wants to send packet to a destination, it first extracts the subnet address from its top hierarchy and obtains the hierarchical address of the corresponding home agent. It then sends the packet to the home agent using the hierarchical address.

The major advantage of HSR is that the hierarchical address is employed to make management and routing process of networks more scalable than flat networks. However, the assumption of home agent containing HIDs of all the nodes is not good since it is difficult to gain global information in an infrastructureless system. Moreover, in a mobile environment due to the movement of the nodes, nodes at a high level may frequently be troubled in order to maintain and update the hierarchical addresses for the mobile nodes at low levels. This causes the creation of critical nodes at high levels, which are fragile and inclined to be deadlocked by communications with nodes at low levels. Furthermore, communication between low levels and high levels consumes much bandwidth resources when mobility is high.

2.6 Problems of Hierarchical Routing Protocols

Having reviewed two typical hierarchical routing protocols, we can find some problems of hierarchical routing protocols. Three problems are emphasized here: convergence, stability and distribution.

- Convergence indicates the ability of the hierarchical structure to get organized in a short time interval. This is important because hierarchical routing protocols can

CHAPTER 2. LITERATURE REVIEW

not work without a hierarchical structure. In other words, the time spent to initiate the hierarchical structure should occupy only a trivial ratio of the whole routing process.

- Stability implies that in a mobile environment the hierarchical structure is not likely to be affected by movement of nodes, and the different levels of hierarchical structure are quite independent. This independency means updates generated by one level will seldom influence other levels, i.e. the update generated from low levels will not be propagated to high levels.
- Distribution means that the mechanism will not create critical nodes for the network. Each node should deal with a fair amount of tasks in the routing processing and most of the work should not be centralized to particular nodes.

Among these three problems convergence and stability are related to clustering technique; stability is also related to maintenance in hierarchical routing schemes.

Chapter 3

Improving Cluster-based Routing on Caching & Adaptivity

In this chapter we focus on cluster-based routing algorithms. Existing clustering techniques are reviewed and design principles of the clustering techniques are given. After that, we propose two methods to improve cluster-based routing performance: (1) We present a passive caching mechanism used on cluster-based routing protocols to improve performance of clustering algorithms; (2) We propose an adaptive system to reduce control overhead caused by periodic hello messages in cluster maintenance.

3.1 Existing Clustering Techniques

Clustering is a method by which nodes are organized based on their relative proximity to one another. The network is divided into clusters using clustering techniques. One node in each cluster assumes the role of a clusterhead, and plays a more active role in keeping track of the topology and feasible routing paths. The role of the clusterhead is a temporal and it changes dynamically as the topology or other factors affecting it change. There are some existing mechanisms for cluster formation, e.g. node-id based, degree based, mobility based, etc; and each scheme focuses on a particular characteristic of the network.

Usually clustering schemes include two phrases:

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

- Marking: organize the network into clusters and let each node belong to at least one cluster
- Competition: when two clusterhead move into transmission with each other, one remains as clusterhead and the other relinquishes its role of clusterhead.

Compared to other routing protocols, the cluster-based routing approaches incur the lower overhead because of local topology updates and also have quicker convergence [KVCP97]. The effectiveness of this approach also lies in the fact that existing routing protocols can be easily extended to find routes between different clusters simply by replacing the nodes with clusters.

In MANETs the topology changes constantly and it is difficult for each mobile node to have knowledge about the global topology, so it is not wise for a clusterhead to maintain a cluster diameter with long distance. Therefore, clustering techniques with long cluster diameter are not appropriate for MANETs. Currently most clustering algorithms in MANETs are focused on short distance clustering, especially one hop clusters. In the following sections, we review some representative ones.

3.1.1 Highest-degree Algorithm(HID)

The highest-degree algorithm, also as known as connectivity-based clustering, was originally proposed by Gerla and Parekh [GT95] [Par94]. Each node broadcasts beacons and receives acknowledgements to obtain the number of neighbors. The node with the maximum number of neighbors is chosen as a clusterhead. When two neighboring nodes have the same number of neighbors, the node with higher ID is selected as the clusterhead. The neighbors of a clusterhead become the members of that cluster and can no longer participate in the clusterhead election process. Since no clusterheads are direct neighbors, only one clusterhead exists per cluster. Any two nodes in a cluster are at

most two-hops away since the clusterhead is one hop away from each of its neighbors in the cluster. A node is called a gateway if it lies within the transmission range of two or more clusterheads. Gateway nodes are generally used for routing between clusters. Only gateway nodes belong to different clusters.

3.1.2 Lowest-id Algorithm(LID)

The lowest-id algorithm is one of the ID based clustering algorithms and it was originally proposed by Baker and Ephremides [BE81a] [BE81b] [GT95]. This algorithm assigns a unique ID to each node and chooses the node with the smallest ID as the clusterhead. Thus, the IDs of all the neighbors of the clusterhead will be higher than that of the clusterhead. However, the clusterhead will relegate its responsibility if a new node with a lower ID moves into its cluster.

3.1.3 Least Clustering Change Algorithm(LCC)

LCC [CG97] defines a stable competition rule for clustering techniques. In LCC, a lowest-id cluster algorithm or a highest-degree cluster algorithm is used to create initial clusters. When a non-clusterhead node in cluster i moves into a cluster j , the clusterheads in cluster i and j will not change(only cluster members will change). When a non-clusterhead node moves out of its cluster and doesn't enter into any existing cluster, it becomes a new clusterhead and forms a new cluster. When clusterhead $C(i)$ from cluster i moves into the cluster j , it challenges the corresponding clusterhead $C(j)$. One of them, $C(i)$ or $C(j)$, will give up its clusterhead position according to lowest-id or highest-degree metric(or some other predetermined priority metric).

LCC is more stable than HID and LID since only two cases cause reformation of the cluster.

- When two clusterheads move into transmission range of each other, they will compete and reform the clusters.

- When a node does not belong to any cluster, it will declare itself as clusterhead and include its neighbors as its cluster members.

3.1.4 Random Competition Based Algorithm(RCC)

Stability is a serious problem in cluster-based routing algorithms, especially when clustering is used to support routing. For a hierarchical structure, the stability of backbone nodes and local subnets are highly preferred.

The main idea of random competition based algorithm(RCC) [XG02] is that any node, which does not belong to any cluster, can initiate a cluster formation by broadcasting a packet to declare itself as a clusterhead. The first node, which broadcasts such a packet, will be declared as the clusterhead of this cluster. Clusterheads have to periodically broadcast a clusterhead claim packet to maintain their clusters. Since there is a delay from the time one node broadcasts its clusterhead claim packet to the time this packet is heard by its neighbors, several neighbor nodes may broadcast during this time period. To reduce such concurrent broadcasts, a random timer is introduced. Each node defers a random time before its clusterhead claim. If a node hears a clusterhead claim during this random time, it then gives up this broadcast.

In this scheme the random timer is able to revolve some of the conflicts when two nodes which within transmission range of each other try to broadcast simultaneously. Of course, the random timer cannot completely solve the concurrent broadcast problem. When such concurrent broadcasts happen, the node ID is used to resolve the conflict and the node with lower ID will become the clusterhead. Simplicity when two clusterheads move into transmission range of each other, the one with lower ID will remain as clusterhead.

3.1.5 Mobility Based Algorithm(MOBIC)

In mobility based algorithm(MOBIC) [BKL01], a mobility metric is proposed to capture and quantify the motion of nodes; this proposed metric is geometric in the sense that the speed of a node is measured relative to the other nodes. The mobility measure between any pair of nodes is defined as their absolute relative speed taken as an average over time. In order to model mobility, the power level (received signal strength) detected at the receiving node is used. R_{rx} indicates the received power. T_{rx} indicates the transmit power. In the ideal situation, Friis' free space propagation model[Rpp96] is used, which uses an inverse-square dependence of the ratio of received and transmit power on the physical distance between the transmitter and the receiver, i.e. $\frac{R_{rx}}{T_{rx}} \propto \frac{1}{d^2}$ [BKL01]. But in real life, an exact calculation of the distance between a transmitter and a receiver may not be possible from the measured signal strengths owing to the complexities involved in accurate channel modeling. However, from the signal strength between two successive packet transmissions from a neighboring node, a good indication of the relative mobility between the two nodes can be obtained. The relative mobility of a node is obtained by calculating the variance of the relative mobility with all neighbors. This variance is then used as mobility metric for each clusterhead. When two clusterheads come into transmission range of each other, the one with lower mobility metric will become the new clusterhead. The relative mobility, $M_Y^{rel}(X)$, at a node Y with respect to X, is defined in the manner of equation Eq. 3.1:

$$M_Y^{rel}(x) = 10 \log_{10} \frac{P_{rx}X \rightarrow Y^{new}}{P_{rx}X \rightarrow Y^{old}}. \quad (\text{Eq. 3.1})$$

The $P_{rx}X \rightarrow Y^{new}$ indicates the new received power from node X to node Y. The $P_{rx}X \rightarrow Y^{old}$ indicates the old received power from node X to node Y. The mobility metric is defined in this manner: $M_Y = \text{var}_0(M_Y^{rel}(X_1), M_Y^{rel}(X_2), \dots, M_Y^{rel}(X_m)) = E[(M_Y^{rel})^2]$.

Here, var_0 denotes the variance with respect to zero (and not the mean of the sample) and is equal to $E[(M_Y^{rel})^2]$ which is the expected value of the m relative mobility samples from Y 's neighbors.

3.1.6 Design Principles of Clustering Techniques

In this section we summarize the advantages and the disadvantages of the above clustering techniques.

The lowest-id clustering approach is a node-id based clustering scheme. It provides a simple way to form clusters. However, this clustering approach is biased towards certain nodes based on their identities. Hence, the node-id based clustering approach results in non-uniform load distribution. One solution to distribute the load uniformly is to use a random number for deciding clusterheads [BE81a]. But this will result in unnecessary changes in clusters, thus resulting in a less stable system.

The highest-degree clustering mechanism aims at reducing the average number of clusterheads by favoring the nodes with higher connectivity to become clusterheads. Again, this results in an unbalanced load distribution. As the connectivity of a node may change rapidly as it moves, the degree-based clustering mechanism also tends to be less stable.

The LCC intends to make clusters more stable in case of frequent topology changes. In LCC only two situations will cause reformation of the cluster, and this results in less changes within clusters.

The RCC proposes a random process clustering technique specifically to help build a stable mechanism. A random timer is introduced to obtain good convergence as the contention is avoided. When contention is reduced, the control overhead also decreases.

The MOBIC is based on using a mobility metric for the selection of clusterheads to achieve more stable cluster formation and less competition. The number of cluster changes is also reduced.

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

In cluster-based routing algorithms, since the clusterhead deals with more tasks (within and outside clusters) than its cluster members, it consumes resources faster than the other nodes which are not clusterheads. Hence, if a node behaves as a clusterhead for a long duration, its energy will be exhausted. In contrast, nodes which are not clusterheads may consume only a little amount of energy. In order to avoid the exhaustion of particular nodes in the network, load balancing is also an importance consideration for cluster-based routing algorithms. It is necessary to propose some algorithms to give each node a fair chance to behave as a clusterhead.

Thus, from the above discussion we realize the following design principles to guide the development of clustering techniques:

- Small number of clusters, i.e. the average number of clusterheads to the total number of nodes.
- High stability, i.e. the average time a node remained in the same cluster.
- Short convergence, i.e. the initial time cost to finish a cluster formation.
- Fair clusterhead load distribution, i.e. the amount of work performed by each node.
- Low clustering overhead, i.e. the control overhead used to form and maintain clusters.

3.2 Passive Cached Cluster-based Routing for MANETs

In this section we present a passive caching mechanism used on clustering routing protocols for mobile ad hoc networks (MANETs). It utilizes the backbone [KKRM01] of the network to gather and reuse discovered routing information more efficiently and effectively. We propose a caching mechanism based on existing clustering algorithms, we build databases of routing information on nodes along the network backbone organized

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

by the clustering algorithms. Thus, instead of generating routing packets to discover routes, on-demand routes can be found by accessing databases of routing information distributed on the backbone. Therefore, routing overhead caused by route discovery is reduced.

3.2.1 Route caching

In a reactive routing protocol, a newly discovered route can be cached, so that it may be reused the next time that the same route is requested. There are two ways to gather cached information: passive way and active way. In the passive way, a node works in overhearing mode and records the routing information carried by routing packets it has received. In the active way, a node broadcasts messages in a limited scope and caches routing information carried by replies. The advantage of passive mode is that it gathers useful information without producing any extra routing overhead; while the advantage of active mode is that through broadcasting routing packets and receiving feedback, cached routing information can be obtained more purposefully and in time than passive mode at the expense of control overhead. In our algorithm we choose the passive way to gather routing information since we may improve the performance of routing without producing any additional control overhead.

3.2.2 Passive cached Method

In this section, we describe our passive cached mechanism and discuss how it can be used in cluster-based routing algorithms. Existing routing protocols such as proactive or reactive protocols can be used in cluster-based routing algorithms to forward packets from one cluster to another cluster. Since reactive protocols are preferred for networks with high mobility and large size than proactive protocols, we use reactive protocols as forwarding mechanisms among clusters. For large network when the network size

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

increases, the number of clusters will also increase, and in such case flooding over all clusters will incur much routing overhead and the scarce resources of the MANET will be consumed. Therefore, in order to reduce flooding overhead among clusters during route discovery, cache is configured for some nodes to improve the reuse of the discovered routes. In clustering schemes all route discoveries and route replies go through gateways along the network backbone, gateways are configured to load the cache functionality.

The basic idea of our approach is that gateways will cache routing information by overhearing requests and replies in the passive way. When routes are required, the source will check the nearest gateway for useful information. Once the required routing information is hit in the cache of the gateway, routing overhead caused by routes discoveries can be reduced.

Our mechanism is presented as follows. At initiation the existing clustering scheme LCC [CHWM97] is used to partition the network. After the partition, some nodes become clusterheads, some nodes become gateways, and the backbone of the network system is established. When a route is required, the source node will send a route request message to neighboring gateways. If a gateway has the required routing information in its cache, it will reply the source node immediately; otherwise, it does nothing. If the source node does not hear any reply after a pre-fixed consultant time, it will flood route request messages along the backbone of the network to find the route. In the scheme each gateways works in promiscuous mode and keep refreshing its cache with overheard routing information.

The detailed procedure after a node receives a route request is illustrated in the flowchart of Figure 3.1. When a clusterhead receives a route request, it checks whether the destination lies in its cluster. If it finds the destination within its cluster, it will return a route reply along the reverse path of the forwarding route to inform the source about the list of intermediate nodes to the destination; otherwise, it forwards this route request to all un-traveled gateways in its cluster. When a gateway receives a route request,

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Figure 3.1: Decision after a route request is received

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

it checks its cache to see whether there is available routing information for reaching the destination. If it has, the cached route will be replied to the source; otherwise, it forwards the route request to clusterheads that are not visited by the route request. Any node, which receives the route request packet, will append its address in the routing packet before it re-forwards the packet to other nodes. Therefore, routing packets can record the list of nodes along the forwarding route. A gateway always works in overhearing mode and will cache the routing information contained in the routing packets it has received. When the route request arrives at a node, which has the route information to reach the destination, the node will return a route reply along the reverse of the forwarding route to inform the source about the list of intermediate nodes to the destination.

In DSR [BJM03], a similar caching method is also used to contribute to route discovery. But there is a difference between its use in DSR and our clustering approach. In DSR because each node can cache, the cached routing information is distributed over all the nodes in the network. However, in our clustering approach, routing information is gathered and cached more purposefully along the backbone of the network, hence, it is more efficient and much is saved for the overall network. On the other hand, since the overall routing information is centralized along the backbone, for our clustering approach the request routes are more likely to be hit in the cache than DSR. The advantages of our algorithms are highlighted below:

- It utilizes the backbone to centralize the routing information of the MANET, which makes the routing information be managed more easily and be accessed more efficiently;
- It obtains routing information without incurring any additional overhead;
- Since it reuses the pre-discovered routes in cache, as a result routing overhead caused by route discoveries is reduced.

3.2.3 Simulation results and discussion

3.2.3.1 Simulation environment and parameters

Table 3.1: Simulation Parameters

Simulator	NS-2
MAC	802.11DCF
Transmission Range	250m
Bandwidth	2Mb/s
Simulation area	500x500, 700x700, 870x870, 1000x1000
Number of nodes	50, 100,150,200
Movement model	Random Waypoint
Max Speed	4 m/sec
Pause Time	10 sec
Simulation Time	600 sec
Traffic	UDP/CBR

The simulations were performed using the ns-2 network simulator with wireless extensions [Pro]. Table 3.1 shows the parameters of our simulation. The message exchange uses a MAC layer that realizes the default characteristics of the distributed coordination function (DCF) of IEEE 802.11. The channel capacity and transmission range are 2 Mbits/sec and 250m respectively. We varied the network size to be 50, 100, 150 and 200 nodes. We fixed the density of the network and keep the simulation area proportional to the number of nodes. The scenarios were randomly generated with input parameters such as maximum speed, pause time, number of nodes, area etc. The mobility model used was the Random Waypoint Model with pause time fixed to 10 seconds, while the max speed of mobility is 4m/sec. Constant Bit Rate (CBR) (512 bytes/packet/sec) applications were used to generate network data traffic. Each CBR sends four packets every second. In total 10 source-destination pairs were spread randomly over the network. For each point in the figure, 100 simulation runs were carried out and the result was averaged.

3.2.3.2 Results and discussion

In simulation the cluster-based routing algorithm without caching method is called non-caching algorithm and our algorithm with caching method is called caching algorithm. We compare the caching algorithm with the non-caching algorithm to evaluate them.

The following metrics are used to evaluate the performance: (i) Control overhead: the total number of routing packets (or control packets) transmitted by a node, averaging over all the nodes; (ii) Packet delivery ratio: the ratio between the number of data packets received by the destinations and those originated by the sources; (iii) Average end-to-end packet delay: the time from when the source generates the data packet to when the destination receives it.

Figure 3.2 shows control overhead as a function of increasing network size. We can see that the caching algorithm outperforms the non-caching algorithm; it reduces about 5% to 10% of the control overhead compared with the non-caching algorithm. The improvement of our approach over the existing clustering approach is because of the cache scheme. Gateways on the backbone cache much of the routing information, they can reply some pre-cached routes when routes are required, therefore the control overhead caused by route discovery packets is reduced.

Figure 3.3 and Figure 3.4 show delivery ratio and delay as a function of increasing network size respectively. We can see that the caching algorithm and non-caching algorithm have similar performance in terms of delivery ratio and end-to-end delay. Thus, compared with the non-caching algorithm, the caching algorithm can achieve reduction in routing overhead without any degradation in other performance aspects. Therefore, on the while, we achieved low overhead without imposing delivery ratio and end-to-end delay.

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Figure 3.2: Control Overhead Vs Network Size

Figure 3.3: Delivery Ratio Vs Network Size

Figure 3.4: End-to-end Delay Vs Network Size

3.3 Adaptive Clustering for MANETs

This section presents an adaptive system used on clustering routing protocols for mobile ad hoc networks (MANETs). We propose an adaptive mechanism based on existing clustering routing algorithms. Instead of using fixed broadcast intervals for clustering maintenance we utilize the node mobility as feedback to tune the broadcasting interval of each node. Consequently, the hello messages broadcasted for clustering maintenance are generated more efficiently and the control overhead is reduced [YY04].

3.3.1 Adaptive System

Due to the unpredictable nature of ad hoc network system, a highly adaptive scheme is required to deal with the frequent topology change. Generally, adaptive systems are composed of two parts: feedback and update. In networking systems feedback is the input that the networking system obtains from the environment; which usually is some

specific characteristic of the network, such as queuing delay, node mobility, link duration and so on. Update on the other hand is the model used to adapt the networking system behavior depending on feedback. Adaptiveness can be achieved by letting nodes change system parameters or letting the system itself make the necessary changes. We propose an adaptive clustering routing protocol that utilizes the node mobility as feedback to tune the broadcasting interval of each node. As a result the broadcasted messages for clustering maintenance become adaptive to local mobility and are generated more efficiently.

3.3.2 An adaptive system for clustering routing

In traditional clustering algorithms [SS02], nodes broadcast packets at fixed frequencies in limited scope to help other nodes obtain knowledge of their neighborhood for maintaining intra-cluster [YF03] connections or for connections between clusters. But the fixed broadcasting frequency is not efficient for a network system with irregular dynamically changing topology. In certain occasions, e.g. when nodes belong to the same cluster do not move during a period of time, it is not necessary to broadcast packets for maintenance because there are no changes. However, if the mobility in the area is high, nodes need to broadcast messages more frequently to help neighboring nodes update the network topology. Therefore, the frequency of generation of hello messages for a node is related to the local mobility. In situations of high mobility, nodes need to broadcast messages more frequently to provide more accurate topology information; while in low mobility, infrequent broadcasting of messages is enough to maintain the clusters.

For the sake of simplicity, we present our algorithm in two dimensions, but it can be easily extended into three dimensions. Figure 3.5 illustrates our idea. Assume that there are two nodes: node A and node B , and node A moves with velocity V_1 and node B moves with velocity V_2 . The angle between V_1 and V_2 is α . The distance between node A and node B is d at time t .

Figure 3.5: Illustration of the adaptive algorithm

In Figure 3.5, we decompose V_2 into two directions; one direction along V_1 (X direction), the other direction vertical to V_1 (Y direction). Then, we calculate the distance between node A and node B after a time interval of Δt . At direction X , node B moves $(V_2 \Delta t \cos \alpha - \Delta t V_1)$ with respect to node A ; at direction Y , it moves $(V_2 \Delta t \sin \alpha)$ with respect to node A . After a time interval of Δt , the distance between two nodes is $d' = \sqrt{(\cos \alpha \Delta t V_2 - \Delta t V_1)^2 + (\sin \alpha \Delta t V_2)^2} + d = \sqrt{(\Delta t V_2)^2 + (\Delta t V_1)^2 - 2 \cos \alpha \Delta t^2 V_1 V_2} + d$. When α is equal to 180° , $d'_{max} = \sqrt{(\Delta t V_1 + \Delta t V_2)^2} + d = \Delta t (V_1 + V_2) + d$.

Assuming that the transmission range of a node is D and the maximum possible velocity of a node is V_{max} , the time that a node with a velocity V_o spends to move at a distance of D with respect to its neighbors is at least $\frac{D}{V_o + V_{max}}$. Therefore, if a node knows its velocity, it can estimate the lower bound t_l of the period it will take to move out of the transmission range of its neighbors.

The method to get the velocity of a node is aided by the development of GPS [CHH01]. Each node determines its own geographical location by the aid of GPS. A node can calculate its instantaneous velocity by calculating the position change over a small time

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

interval.

Considering the worst case, the broadcast interval of node x should be inversely proportional to $V_x + V_{max}$. In implementation, the broadcast interval of node x is set to $\frac{D}{V_x + V_{max}}$. This setting is conservative, since the broadcast interval is set to the minimum value for a node to move out of its neighbors' transmission range.

A node will piggyback its broadcasts interval in the broadcasted messages. When a node y receives a hello message from node x , it will update the timeout field for the entry of x depending on the broadcast interval included in the hello message from node x . The value of the timeout field is proportional to the corresponding node's broadcast interval. If the sender's mobility is high, it stays for a short time in the neighbor table; otherwise, it stays a long time in the neighbor table. This mechanism helps to make the neighboring information more accurate and fresh.

3.3.3 Simulation Results and Discussion

Table 3.2: Simulation Parameters

Simulator	NS-2
MAC	802.11DCF
Transmission Range	250m
Bandwidth	2Mb/s
Simulation area	500x500, 700x700, 870x870, 1000x1000
Number of nodes	50, 100, 150, 200
Movement model	Random Waypoint
Max Speed	20 m/sec
Pause Time	10 sec
Simulation Time	600 sec
Traffic	UDP/CBR

The simulations were performed using the ns-2 network simulator with wireless extensions. Table 3.2 shows the parameters of our simulation. The message exchange

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

uses a MAC layer that realizes the default characteristics of the distributed coordination function (DCF) of IEEE 802.11. The channel capacity and transmission range are 2Mbps/sec and $D = 250m$ respectively. The mobility model is random waypoint model and the maximum velocity of a node is 20m/s, thus $V_{max} = 20m/s$. In adaptive algorithm each node will use $\frac{D}{V_x + V_{max}}$ to calculate its own broadcast interval and piggyback this broadcast interval in hello messages to let its neighbors set the corresponding timeout field for itself. We varied the network size to be 50, 100, 150 and 200 in order to show that the adaptive clustering algorithm is still scalable when network size increases. We fixed the density of the network and keep the simulation area proportional to the number of nodes. The scenarios were randomly generated with input parameters such as maximum speed, pause times, number of nodes, area etc. The pause time was fixed to 10 seconds. Constant Bit Rate (CBR) (512 bytes/packet/sec) applications were used to generate network data traffic. Each CBR sends four packets every second. In total 10 source-destination pairs were spread randomly over the network. For each point in the figure, 100 simulation runs were carried out and the results are averaged. Since the confidence interval is very small in the simulation, we take it off from the figure for clarity.

Since $\frac{D}{V_x + V_{max}}$ falls into the range of $[\frac{D}{2V_{max}}, \frac{D}{V_{max}}]$. We compare our adaptive clustering algorithm with two original clustering algorithms with different broadcast intervals as $\frac{D}{2V_{max}}$ and $\frac{D}{V_{max}}$ respectively. Control overhead, packet delivery ratio and end-to-end packet delay are used to evaluate the performance.

Figure 3.6 shows control overhead as a function of increasing network size. If the broadcast interval of the original clustering algorithm is set to $\frac{D}{2V_{max}}$, the adaptive clustering algorithm has lower control overhead than the original clustering algorithm; it reduces approximately 20% of the control overhead. In this case, the adaptive clustering algorithm generates less control overhead since broadcast interval of each node in the adaptive clustering algorithm is always greater than $\frac{D}{2V_{max}}$. If the broadcast interval of

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Figure 3.6: Control overhead Vs. Network Size

Figure 3.7: Delivery ratio Vs. Network Size

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Figure 3.8: End-to-end delay Vs. Network Size

the original clustering algorithm is set to $\frac{D}{V_{max}}$, the adaptive clustering algorithm has higher control overhead than the original clustering algorithm since each node always broadcasts with a interval smaller than $\frac{D}{2V_{max}}$.

Figure 3.7 shows delivery ratio as a function of increasing network size. We can see that the adaptive clustering algorithm has higher delivery ratio than other two algorithms. Under conditions of low mobility, broadcasting hello messages in high frequency is not necessary and nodes in the adaptive clustering algorithm broadcast with lower frequency compared to nodes in the original clustering algorithm. As a result, control overhead is reduced. While, under conditions of high mobility, in order to distribute more accurate information to neighbors. Nodes broadcast packets more frequently to inform the change to their neighbors.

Figure 3.8 shows delay as a function of increasing network size. We can see that the adaptive clustering algorithm has similar end-to-end delay to the original clustering

algorithms. Therefore, overall the adaptive clustering algorithm achieve a high delivery ratio without causing significant increase in control overhead and end-to-end delay.

3.4 An Improved Cluster-based Routing Algorithm based on Combination of RCC & MOBIC & Caching & Adaptivity

Having discussed and proposed some improvement mechanisms, we discuss combinations of different clustering techniques and routing schemes to improve cluster-based routing performance.

Five design factors were proposed in section 3.1.6 to guide the design of the clustering techniques. Clustering techniques discussed above are classified according to these five factors.

- Number of clusters: HID;
- Stability: LCC, MOBIC;
- Convergence: RCC;
- Load distribution;
- Clustering overhead: Adaptive clustering.

For the first factor, in order to let nodes with high degrees(number of neighbors) be a clusterhead, cluster will be reformed often, depending on the degree of clusterheads. If the first factor is desired, it is difficult to achieve the other four factors. The fourth factor is also hard to be fulfilled with other factors simultaneously.

The second factor and the fourth factor can not be achieved simultaneously. On the basis of the second factor, nodes with specific characteristics(small identity number, low

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

mobility) are inclined to be clusterheads for a longer duration compared with common nodes. On the contrary, to achieve the fourth factor, each node is prevented from being clusterhead too long. Hence, if the fourth factor is desired, the second factor will not be fulfilled.

The third factor can be combined with other clustering techniques at the initiation of clustering. Convergence can be achieved by using random processing(RCC) at the beginning. Then, after the partition of network is completed, RCC is discarded and other clustering techniques are applied for different factors during the maintenance.

The fifth factor can also be achieved with other factors. Adaptive clustering can be used to reduce the clustering overhead during the maintenance of the clustering mechanisms.

Here we propose a hybrid way of combining multiple clustering techniques to improve clustering performance.

- At initiation, RCC is used to partition the network for good convergence. RCC is also used to reform clusters during maintenance, i.e. a node will use random clustering when it does not belong to any cluster.
- When two clusterheads compete each other, MOBIC is used to decide the competition. And as a result the node with low mobility will win. This scheme helps clustering algorithms obtain good stability by retaining the node with low mobility as clusterhead.
- During the cluster maintenance, adaptive clustering is applied to reduce the clustering control overhead.
- When clusters are organized, passive cached method is used based on the clusters to improve the routing performance.

3.4.1 Simulation and Comparison

In this section, we simulate the random processing and mobility metric based adaptive&cached cluster-based routing protocol(RMACP) described in section 3.4. The simulation runs on the Network Simulator ver.2 (NS2) and mobility extensions to NS from the Carnegie Mellon University.

We compare the performance of three clustering approaches(CBRP, RMACP, MOBIC) by increasing mobility. We vary the mobility to be 0, 2, 4, 6, 8, 10m/sec. Table 3.3 shows the environment of the simulation. The message exchange uses a MAC layer that realizes the default characteristics of the distributed coordination function(DCF) of IEEE 802.11. The channel capacity is 2 Mbits/sec and transmission range is 250 m. The network has 50 nodes randomly distributed in a 670x670 simulation area. The pause time is fixed to 10 second, while the mobility speed for each node varies between 0 to 10m/sec. Constant Bit Rate(CBR)(512 bytes/packet/sec) applications are used to generate network data traffic. Each CBR sends four packets every second. In total 10 source-destination pairs are set up randomly over the network. For each point in the figure, 100 simulation runs were carried out and the results were averaged. Since the confidence interval is very small in the simulation, we take it off from the figure for clarity.

RMACP is a combination clustering mechanism of RCC, MOBIC schemes and adaptive&caching techniques, while CBRP which uses LCC clustering scheme is purely ID based approach, MOBIC is purely mobility based clustering approach.

Figure 3.9 shows the number of control packets as a function of various mobility on the performance of CBRP, RMACP and MOBIC. In the figure, we can see that RMACP generates lower control overhead than the other protocols and MOBIC generates lower control overhead than CBRP. Since RMACP uses mobility metric and random processing to make clusters more stable and converge well, the control overhead lower

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Figure 3.9: Control Overhead vs. Mobility

Figure 3.10: Delay vs. Mobility

CHAPTER 3. IMPROVING CLUSTER-BASED ROUTING ON CACHING & ADAPTIVITY

Table 3.3: Simulation parameters

Simulator	NS-2
MAC	802.11DCF
Transmission Range	250m
Bandwidth	2Mb/s
Simulation area	670m x 670 m
Number of nodes	50
Movement model	Random Waypoint
Max Speed	2,4,6,8,10 m/sec
Pause Time	10s
Simulation Time	600s
Traffic	UDP/CBR
Broadcast Interval	2.0s

control overhead is generated than other protocols. Additionally, adaptive and caching techniques are applied to reduce control overhead. MOBIC generates less overhead than CBRP since it uses mobility metric to increase stability of clusters, but not as less as

Figure 3.11: Delivery ratio vs. Mobility

RMACP because adaptive and caching techniques are applied in RMACP.

Figure 3.10 shows the delay as a function of various mobility. RMACP has lower delay than CBRP and MOBIC because it converges faster by using random processing.

Figure 3.11 shows the delivery ratio as a function of various mobility on the performance of CBRP, RMACP and MOBIC. From the figure, we can see that the delivery ratio of three clustering mechanisms are very close to one another. Therefore, RMACP reduces control overhead without any degradation on the delivery performance.

Finally, with the above simulation results we conclude that RMACP outperforms existing clustering techniques due to the combination of advantages of different clustering techniques.

3.5 Summary

In this chapter, we first introduce the existing clustering techniques and give the design principles for clustering schemes. Based on the principles, we discussed the advantages&disadvantages of the existing clustering approaches. Second, we employ passive cached method and adaptive maintenance scheme to reduce the control overhead for routing and maintenance of the clustering approaches. Finally, we propose RMACP scheme to achieve the best performance by combining different clustering techniques such as RCC, LCC and MOBIC into an integrated scheme.

Chapter 4

A Random Clustering Based Hierarchical Routing Algorithm

In this Chapter, firstly we present a Random Clustering based Hierarchical Routing(RCHR) protocol. Then we describes hierarchy construction, route discovery and maintenance of the hierarchical structure in detail. Finally, our protocol is evaluated in simulations by comparing with other existing routing protocols. The comparison shows that our hierarchical approach is more scalable than the existing routing protocols(both flat routing protocols and hierarchical routing protocols).

4.1 Clustering technique for hierarchical strategy

Clustering techniques were discussed in detail in chapter 3. Hierarchical schemes use clustering algorithms at each level to organize nodes into clusters. The formation of hierarchical structure is consequently triggered from the lowest level to the highest level until all the levels are organized. The convergence time to complete the establishment of the hierarchical structure is the summation of clustering time of all levels. Therefore, in order to gain a short convergence time, we must complete the clustering of each level as soon as possible. As mentioned in Section 3.1.4 RCC is the most rapid clustering technique compared with other clustering mechanisms, therefore, in order to achieve short

convergence time for the hierarchical structure RCC is used as the clustering method in our hierarchical algorithm. The main idea of RCC is that any node, which does not belong to any cluster, declares itself as a clusterhead by broadcasting to its neighbors; when a node receives a broadcasted declaration from its neighbor, it becomes a member of the declarer.

4.1.1 Level status in hierarchical structure

After the formation of the hierarchical structure with RCC clustering technique, the network is organized into multiple levels. Assume that there are M levels and the bottom level is called level-0. The K th level is called level- K . The level status of a node is defined as the highest level that the node belongs to.

For example, the level status of nodes that are not clusterheads at any level is equal to 0. The level status of nodes that are only clusterheads at level-0 but are not clusterheads at any other level is equal to 1. In turn, the level status of nodes that are clusterheads at level- $(K - 1)$ but are not clusterhead at level- K is equal to K . The level status of nodes at the top level is equal to M . In total there are $M + 1$ levels, from level-0 to level- M . Usually a level- K clusterhead is elected into a level- $(K + 1)$ member when the level- $(K + 1)$ comes into being. A level- K node functions as clusterhead at K different levels(from level-0 to level- $(K - 1)$); while it is a normal node at level- K and also a member for $K + 1$ levels(from level-0 to level- K). Nodes with a status of 0 only belong to level-0, while nodes with a level status of $K(0 < K < M)$ belong to $K + 1$ levels(from level-0 to level- K). In the next section we will introduce how to build the cluster tables depending on the level status.

4.1.2 Cluster tables

A node maintains two tables for each level it belongs to: a neighbor table and a neighboring cluster table. To explain neighbor table and neighboring cluster table clearly, we

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

introduce a new term: *Jump*. *Jump* is similar to *Hop*, but there is a difference between them. One *Jump* means the hop distance between a level- K node and its neighboring level- K node at level- K . One *Jump* is equal to one *Hop* at level-0. At level- K ($K > 0$) one *Jump* is more than one hop. In the following discussion if a node is one *Jump* away to another node at the same level, they are neighbors to each other.

A node maintains a neighbor table and a neighboring cluster table for each level that it is a member of. Therefore, a level- K node maintains $2(K + 1)$ tables if the tables for each level are not empty.

A neighbor table records the neighboring nodes at the corresponding level. A neighbor table for level- K is called level- K neighbor table. A level- K neighbor table has the following fields:

- Node_Address
- Level_Status

The *Node_Address* field is the neighbor's address. The *Level_Status* field indicates level status of the neighbor.

A level- K neighboring cluster table contains the following fields:

- Neighbor_Cluster_Head
- Next_Jump_List

The *Neighbor_Cluster_Head* field records address of the level- K clusterhead which is two or three *Jumps* away at level- K . The *Next_Jump_List* indicates the list of the next *Jump* to the level- K clusterhead in the *Neighbor_Cluster_Head* field.

Figure 4.1 illustrates a simple hierarchical structure. In the figure, node 5 and node 7 are level-1 nodes; node 15 is a level-2 node and node 2 is a level-3 node; the others are level-0 nodes.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.1: Hierarchical structure from level-0 to level-3

Table 4.1: level-0 neighbor table for node 2

Node_Address	Level_Status
6	0
10	0
11	0

Table 4.2: level-0 neighboring cluster table for node 2

Neighbor_Cluster_Head	Next_Jump_List
5	10
7	6

Table 4.3: level-1 neighbor table for node 2

Node_Address	Level_Status
5	1
7	1

Table 4.4: level-1 neighboring cluster table for node 2

Neighbor_Cluster_Head	Next_Jump_List
15	5

Table 4.5: level-2 neighbor table for node 2

Node_Address	Level_Status
15	2

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Since node 2 belongs to four levels, it should contain 8 tables from level-0 to level-3 if each table is not empty. The level-0 tables of node 2 are illustrated in table 4.1 and table 4.2. The level-1 tables of node 2 is illustrated in table 4.3 and table 4.4. Since node 2 has only one level-2 neighbor (node 15, which is one *Jump* away), no level-2 neighboring clusterhead exists for node 2 at level-2. Therefore, the level-2 neighboring cluster table of node 2 is empty. Table 4.5 shows that the level-2 neighbor table of node 2. Level-3 tables of node 2 are also empty because there is no neighboring level-3 node near node 2 at level-3.

There are two kinds of member at each level: non-clusterhead and clusterhead. Because a level- $(K + 1)$ member comes from a level- K clusterhead and the entries(level- K clusterheads which are two or three *Jumps* away) in a level- K neighboring cluster table are actually level- $(K + 1)$ members. A level- $(K + 1)$ node obtains its level- $(K + 1)$ neighbor table by mapping entries in its level- K neighboring cluster table into its level- $(K + 1)$ neighbor table when it organizes its level- $(K + 1)$ neighbor table. In other words, for a level- $(K + 1)$ node, entries in its level- $(K + 1)$ neighbor table come from its level- K neighboring cluster table. Thus, a level- $(K + 1)$ neighbor table for a node corresponds to its level- K neighboring cluster table. Furthermore, level- $(K + 1)$ neighbor tables of neighboring node are exchanged through hello messages to form the level- $(K + 1)$ neighboring cluster tables. The maintenance of neighbor tables and neighboring cluster tables through *Hello* messages will be described in the following section.

4.1.3 One-level hierarchical clusters

In this section we will look at the formation of level-0 nodes into clusters. This is the same as building a hierarchical structure with only one level. We use level-0 *Hello* messages to organize the network into level-0 clusters. A level-0 *Hello* message contains the following fields.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

- *Level_Status*
- *IP_Addr*
- *Neighbor_List*
- *Neighboring_Cluster_List*

The *Level_Status* is the level status of the *Hello* message sender. The *IP_Addr* indicates the IP address of the sender. The *Neighbor_List* records all the entries in the level-0 neighbor table of the sender. The *Neighboring_Cluster_List* records node addresses of all the entries contained in the level-0 neighboring cluster table of the sender.

The *Neighboring_Cluster_List* field of level-0 *Hello* message is optimal depending on the sender's status at level-0. A node includes *Neighboring_Cluster_List* if it is not a clusterhead at level-0; otherwise, the field is empty. Since the *Neighboring_Cluster_List* is propagated by nodes between two neighboring clusterheads and is used to help a clusterhead obtain routes to neighboring clusterheads, only non-clusterhead nodes between neighboring clusterheads at level-0 need to contain *Neighboring_Cluster_List* in its level-0 *Hello* message.

4.1.3.1 Clusters formation

RCC is used to organize a network into clusters at level-0. At initiation the level status of each node is undefined (this status is different from 0). An undefined node will delay a random time before broadcasting its first level-0 *Hello* message. If a node receives a level-0 *Hello* message from another node with a level status greater than 0 within its random delay, it will change its level status from undefined to 0; otherwise, at the end of the random delay it will change itself from an undefined node to level-0 clusterhead (level_status=1) and announce its level status to its neighbors by broadcasting a one hop level-0 *Hello* message including level status equal to 1. Although random processing

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

is employed, it is still possible that two undefined nodes announce themselves as level-0 clusterheads to each other at the same time. Since only one clusterhead is allowed in one cluster, contention happens in this case. The addresses of these two nodes are compared and the node with the lower address number wins the contention. As a result, the node with the higher address number will change itself from an undefined node to a level-0 member($\text{level_status}=0$). The level-0 clustering will be completed after two cycles of broadcasting of level-0 *Hello* messages in neighborhood for each node.

4.1.3.2 Topology discovery and table update

After level-0 clusters are formed through the RCC clustering technique, level-0 *Hello* messages will continue to be broadcasted periodically to maintain the clusters. A node updates its level-0 neighbor table and level-0 neighboring cluster table when it receives level-0 *Hello* messages from another node.

When a node *A* receives a level-0 *Hello* message from its neighbor *B*, it will update its level-0 neighbor table and level-0 neighboring cluster table with the processes shown in the flowchart of figure 4.2. The flowchart illustrates operations of a node(say node *A*) on receiving level-0 *Hello* messages from another node(say node *B*), and also the operations of clusters formation with level-0 *Hello* messages.

Step 1: If node *A* is not undefined, go to step2; otherwise, it checks node *B*'s level status; if node *B* is a level-0 clusterhead, node *A* will change its level status from undefined to 0.

Step 2: Node *A* will check if node *B* is already included in its level-0 neighbor table; if yes, it will update the level status of node *B* and sets the new timeout for the entry of node *B*; if not, go to step 3

Step 3: Node *A* appends node *B* into its level-0 neighbor table; it will set the level status of node *B* and the timeout for the entry of node *B*. Each entry in the neighbor

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

table is associated with a timeout. An entry will be removed if a message from that entry's node is not received within the timeout interval.

After node A completes the updating of its level status and its level-0 neighbor table, it begins to update its level-0 neighboring cluster table. Node A tries to find level-0 clusterheads which are two or three *Jumps* away from the *Neighbor_List* and *Neighboring_Cluster_List* contained in the level-0 *Hello* messages from node B . Node A checks node B 's *Neighbor_List* contained in the *Hello* message to know whether there is a level-0 clusterhead in node B 's neighborhood. If it finds a level-0 clusterhead that is not itself, it adds the level-0 clusterhead in its level-0 neighboring cluster table and appends node B into the *Next_Jump_List* field of corresponding entry for the clusterhead; if node A is the only level-0 clusterhead in node B 's *Neighbor_List*, it continues to check the *Neighboring_Cluster_List* of the *Hello* message from node B ; if the *Neighboring_Cluster_List* is not empty, node A will add the level-0 clusterheads contained in the *Neighboring_Cluster_List* into its level-0 neighboring cluster table and appends node B into the *Next_Jump_List*.

4.1.4 Multi level hierarchical clusters

The clustering and topology discovery mechanism for level-0 can be extended to a multi-level hierarchy. The number of levels in a hierarchical system will vary based on the network size. For scalability, more hierarchy is needed for a network with a larger population.

Actually a hierarchical system is formed recursively. Clusterheads at a low level become members at the next higher level and the clustering algorithm used in the low level is reused to organize nodes at the higher level into clusters. A level- M system is established when the clustering algorithm is applied to each of the M levels in turn.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.2: Update procedure upon receiving a Hello message

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

In the following sections we present a multi-level hierarchical clustering protocol. Clustering and message exchanging techniques are extended to build a multi-level hierarchy.

Clusters at each level are created and maintained by exchanging *Hello* messages within one *Jump*. At level- K ($K > 0$), a level- K node exchanges *Hello* messages with its one *Jump* level- K neighbors. Since one *Jump* is larger than one hop at the level which is higher than level-0, a level- K ($K > 0$) node uses multiple unicast to propagate the *Hello* messages to its level- K ($K > 0$) neighbors.

4.1.4.1 Clusters formation at level- K ($0 < K < M$)

At each level, RCC is used to organize clusters. When a node becomes level- $(K - 1)$ clusterhead, it is placed into level- K and becomes level- K member. A level- K node will check its level- K neighbor table periodically, if its level- K neighbor table is empty or there is no node having level status which is greater than K in its level- K neighbor table, it announces itself as level- K clusterhead after a random delay if it does not receive any *Hello* message from level- K clusterhead during this random period. It then changes its level status from K to $K + 1$ and will include this information in its next round of level- K *Hello* message.

4.1.4.2 Topology discovery and table update at level- K ($0 < K \leq M$)

A node learns of its neighboring topology through *Hello* messages at each level. A level- K *Hello* message is generated and exchanged periodically within a node's one *Jump* scope. A node which is a member at multiple levels produces multiple *Hello* messages following different cycles at different levels. Commonly, a node sends *Hello* messages of low levels more frequently than that of high levels. In other words, the cycle of generating *Hello* messages at high levels is longer than that at low levels. Generation of *Hello* messages at

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

5 (sender ID)	1 (level status)
3 (neighbor ID)	0 (level status)
9 (neighbor ID)	0 (level status)
10 (neighbor ID)	0 (level status)
12 (neighbor ID)	0 (level status)
I am clusterhead in level-0, do not include neighboring cluster list	

Figure 4.3: Level-0 hello message of node 5

5 (sender ID)	1 (level status)
2 (neighbor ID)	2 (level status)
15 (neighbor ID)	3 (level status)
Level-1 neighboring cluster table is empty, hence this part is null.	

Figure 4.4: Level-1 hello message of node 5

different levels follows different cycles, while nodes at the same level use the same cycle to produce *Hello* messages.

Usually a level- K node generates different $K + 1$ *Hello* messages(level-0 *Hello* message, level-1 *Hello* message, ..., level- K *Hello* message) in different cycles at different levels. However, different *Hello* messages at different levels have the same data structure. Normally, a level- K *Hello* message is composed of four parts(*Level_Status*, *IP_Addr*, *Neighbor_List* and *Neighboring_Cluster_List*) as described in section 4.1.3. *Neighbor_List* and *Neighboring_Cluster_List* will contain the corresponding content of level- K tables(neighbor table and neighboring cluster table). *Level_Status* indicates the level status of the sender. *IP_Addr* is the address of the sender.

For instance, referring to the hierarchical structure of figure 4.1 the level-0 *Hello* message and the level-1 *Hello* message generated by node 5 are illustrated in Figure 4.3 and Figure 4.4 respectively. Node 5 is a clusterhead in level-0, hence it does not include neighboring cluster table in level-0 *Hello* message.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

At level- K upon receiving a level- K *Hello* message from its level- K neighbor B , node A updates its level- K neighbor table and level- K neighboring cluster table similarly as that updates of level-0 described in flowchart of figure 4.2 . Node A verifies that node B is contained in its level- K neighbor table and updates the entry for node B in level- K neighbor table; if verification fails, node A adds node B into its level- K neighbor table. If node A and node B are all clusterhead at level- K , contention happens between two nodes and the node with lower address number will win.

Each time a node updates its level- K neighbor table, it further ensures that in its level- K neighbor table there is at least one node with level status greater than K ; otherwise, it announces itself as level- K clusterhead (or level- $(K + 1)$ node) after a random delay. This process guarantees that each node belongs to at least one level- K cluster if the node is not at the top level. After it completes the update of its level status and level- K neighbor table, it begins to update its level- K neighboring cluster table. It tries to discover two or three *Jumps* away level- K clusterhead from neighbor list and neighboring cluster list contained in level- K *Hello* message from neighboring node B . Node A checks node B 's level- K neighbor list contained in *Hello* message to discover whether there is level- K clusterhead (or level- $(K + 1)$ node) in node B 's neighborhood. If it finds a level- K clusterhead and it ensures that the level- K clusterhead is not itself, it adds the level- K clusterhead into its level- K neighboring cluster table and appends node B into the *Next_Jump_List* field as the next *Jump* for the corresponding neighboring level- K clusterhead. If node A is a level- K clusterhead, it continues to add entries contained in the level- K neighboring cluster table of node B into its level- K neighboring cluster table and appends node B into the *Next_Jump_List* field.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.5: Hierarchical structure and hierarchical tree

4.2 Route Discovery

A level- K (K is not the number of top level) node chooses the node whose level status is greater than K in its level- K neighbor table as its parent node. Each node, which is not at the top level, has at least one parent node and is dominated by its parent node. A level-0 node is dominated by at least one node whose level status is greater than 0; a level- $(K - 1)$ node is dominated by at least one whose level status is greater than K . Actually each node (except level-0 node) dominates a tree as a root node, a level- K node dominates a tree with $(K + 1)$ levels (from level-0 to level- K). A node will send the view of its dominated tree to its parents. Thus, the node at the top level records a tree browsing from itself to the bottom (level-0) and records hierarchical addresses of all members of the tree. Hierarchical address of a node is the sequence of the addresses of the nodes on the path from the top level parent to itself. Figure 4.5 illustrates an example of hierarchical tree. The hierarchical address of node 13 from node 2 to itself is (2, 15, 15, 13). While, the hierarchical address of node 13 stored in node 15 is (15, 15, 13). The hierarchical address of node 15 is (2, 15, 15, 15). Node 15 dominates a node set composed of 3, 5, 9, 10, 12, 13, 14, 15 and it contains eight hierarchical addresses for all the nodes in the set. A level- K ($K > 0$) node piggybacks its identity address and hierarchical address of all members in its dominated tree when it sends *Hello* message to its parent at level- $(K + 1)$. When its parent receives this message, it appends itself into the hierarchical address of all its dominated members and keep sending the dominated members to its parent until the top level is reached. Upload of all member addresses is completed by unicasting from the bottom to the top once and the route length is equal to the number of the top level. Hence, it is a scalable way to help nodes at the top level obtain the topology information of all the downlink nodes. When the hierarchical system is established well, the routing process along the hierarchical structure is quite scalable and the robustness of routes is achieved.

4.2.1 Route Requests

A node disseminates a Route Request(RREQ) when it decides that it needs a route to a destination without any available routing information.

A RREQ message contains following fields:

- Sequence_Number
- Source_IP_Addr
- Destination_IP_Addr
- Next_Hop
- Forwarding_List

Each RREQ message is stamped by an unique sequence number. The *Sequence_Number* field in the RREQ message is the source node's own sequence number, which is incremented before inserting in a RREQ. Before broadcasting the RREQ, the sending node buffers the RREQ sequence number and the originator IP address (its own address) of the RREQ for the period of discovering path. In this way, when the node receives the packet again from its neighbors, it will not re-process and re-forward the packet. This mechanism avoids duplication in route discovery. The field of *Source_IP_Addr* and *Destination_IP_Addr* indicates the identities of the source and the destination. Since a RREQ message is forwarded from low levels to high levels until the destination is found or the top level is reached. For a node which is not at the top level, the *Next_Hop* field indicates the parent node of the current forwarding node; while for a node which is at the top level the *Next_Hop* field indicates the neighboring node at the top level. The *Forwarding_List* field includes the forwarding nodes ever involved in the *Next_Hop* field along the hierarchy from the source to the destination.

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

When a node requires a route to a destination, it obeys following rules to find a route by sending a route request.

- Step 1: When a node receives a route request, it checks whether it has received a RREQ with the same originator IP address and RREQ sequence number within at least the last period of path discovering; if such a RREQ has been received, it discards the RREQ; otherwise, go to step 2.
- Step 2: On receiving a route request, a node performs different operations depending on its status. If it is a level-0 node, go to step 4; if it is not a level-0 node, it discovers the destination by checking its dominated tree. If it finds the destination in its dominated tree, go to step 7; otherwise, go to step 3.
- Step 3: If the node's address is equal to the value indicated in the *Next_Hop* field of RREQ, it appends itself into the *Forwarding_List* and sets the *Next_Hop* field of the RREQ to one of its parent nodes, forwarding the RREQ to the node indicated in the *Next_Hop* field; otherwise, go to step 4.
- Step 4: Continue to forward the RREQ to the node indicated in the *Next_Hop* field.
- Step 5: Repeat from step 2 to step 4 until the top level is reached or the destination is found in the dominated tree. If the destination is found in the dominated tree, go to step 7; otherwise, go to step 6.
- Step 6: When a node at the top level receives a route request, it discovers the destination by checking its dominated tree; if it finds the destination in its dominated tree, go to step 7; if it can not find the destination, it must append its address in the forwarding list of the RREQ message and includes each of its neighbors at the

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

top level in the *Next_Hop* field, forwarding the RREQ to the node indicated in the *Next_Hop*.

- Step 7: If a node finds the destination in its dominated tree, it initiates a route reply and forwards it back to inform the source node.

We give an example to illustrate the forwarding of route requests. In figure 4.5 when node 14 requires a route to node 6, a RREQ is initiated by node 14. Since node 14 is a level-0 node, it sets the *Next_Hop* field to its parent(node 15) and forwards the RREQ message to the node 15. When node 15 receives the RREQ message, it checks whether it contains node 6 in its dominated tree; when node 15 can not find node 6 in its dominated tree, it appends itself into the forwarding list and fills the *Next_Hop* field with its parent node 2, forwarding the RREQ message to node 2. However, node 2 is more than one hop away from node 15. Node 2 is two *Jumps* away from node 15 at level-1. Hence, node 15 checks its level-1 neighbor clustering table and finds that the next *Jump* to node 2 is node 5 at level-1. It then tries to forward the RREQ message to node 2 relaying by node 5 at level-1. By checking level-0 neighbor clustering table, node 15 knows the next hop to node 5 is node 12. In order to store the relaying nodes to parent node 2, node 15 initiates a temporal triple array to store the relaying nodes to node 2, setting (*Level - 0_relaying_node*, *Level - 1_relaying_node*, *Level - 2_relaying_node*) to (12, 5, 2). Node 15 piggybacks the temporal triple array in the RREQ message and sends the RREQ message to node 12. When node 12(a level-0 node) receives RREQ message from node 15, it tries to send the RREQ message to the level-1 relaying node in the temporal triple array. It checks its level-0 tables and determines that node 5 is its neighbor; it changes the temporal triple array to (5, 5, 2) and forwards the RREQ message to node 5. When node 5(a level-1 node) receives the RREQ message, it tries to forward the RREQ message to the level-2 node in the temporal triple array. It checks level-1 tables and level-0 tables and changes the temporal triple array to (10, 2, 2), sending the RREQ message

to node 10. Finally, node 10 relays the RREQ message to node 2 and the temporal triple array used for forwarding is discarded at node 2. In this way, the RREQ message are transmitted from node 15 to node 2. When node 2 receives the RREQ message, it checks and finds that node 6 belongs to its dominated tree. Then, node 2 initiates a route reply message and informs the source node. In this example the *Forwarding_List* field of the RREQ message is (15,2), which is quite shorter than that list of DSR(15, 12, 5, 10, 2). Hence, forwarding along the hierarchy compresses the forwarding list carried in the RREQ message and makes the packet header more scalable.

To reduce congestion in a network, repeated attempts of a RREQ by a source node at route discovery for a single destination must utilize a binary exponential backoff [PBRD03].

4.2.2 Route Replies

When a node, which contains the destination in its dominated tree, generates a RREP message, it appends the hierarchical address of the destination into the *Forwarding_List* of the RREQ message and forwards the expanded list to the source.

A node generates a Route Reply(RREP) in two cases:

- (i) it contains both the source and the destination in its dominated tree;
- (ii) it contains only the destination in its dominated tree.

According to above two cases, a node forwards the RREP message in two manners:

- If it contains the source in its dominated tree, it has the hierarchical address of the source. Therefore, it can use the hierarchical address to return the RREP message to the source directly;
- If it does not contain the source in its dominated tree, it uses the reverse path of the *Forwarding_List* in the RREQ message to forward the RREP to the source.

For example in figure 4.5, when node 14 requires a route to node 6, it sends the RREQ message to node 2 which stores the information of the destination. Afterward the *Forwarding_List* in the RREQ message is changed to (15,2) by node 2. The hierarchical address of the destination is provided by node 2 as (2, 2, 2, 6), it can be shorten to (2,6) and appended into the forwarding list of RREQ. Thus, the forwarding list(14, 15, 2, 6) from the source to the destination is included in the RREP message and sends back to the source along the reverse route.

4.3 Data Delivery

The forwarding list of nodes along the hierarchy replied by RREP is sufficient to deliver a packet to its destination from anywhere in the network. Referring to figure 4.5, considering the delivery of a packet from node 14 to node 6, node 14 can find the forwarding list from itself to node 6 through emitting RREQ messages and receiving RREP messages. The list of nodes along the hierarchy from node 14 to node 6 is 14, 15, 2, 6. After a route is discovered, data packets are transmitted from the source to the destination guided by the discovered forwarding list.

Firstly node 14 tries to send data packets to node 15. When node 14 detects node 15 within its neighborhood, it sends data packets to node 15 directly. After node 15 receives data packets from node 14, it checks the forwarding list in data packets and intends to send data packets to node 2, which is the next node in the forwarding list. As discussed in section 4.2.1, a temporal triple array is initiated to help to relay data packets from node 15 to node 2. Finally, when node 2 receives the data packets, it deliveries the data packets to node 6 along the downwards hierarchical path.

The main idea of data delivery is that forwarding data packets upwards along the hierarchy until the root of the tree which contains the destination node, then forwarding packets downwards using the hierarchical address from the root to the destination node.

4.4 Maintenance

4.4.1 Maintaining hierarchical structure with forwarding pointers

In this section, we apply forwarding pointer to avoid the bottleneck effect for the nodes at high levels. As a result, the communication overhead caused by node movement in hierarchy maintenance is reduced.

4.4.1.1 Forwarding Pointers

In hierarchical system a node at the top level stores information of all members in its dominated tree. In mobile environment, a node is likely to move from a zone to another zone. In traditional maintenance of hierarchical system, when a node drifts from a zone to another zone, it registers with the dominator of the new zone. And the dominator of the old zone removes the information of the drifting node. This mechanism helps each dominator at high levels have fresh information of its dominated nodes. However, this mechanism introduces much communication overhead to nodes at high levels. For instance in the Figure 4.5, when node 13 moves out of transmission range of node 15 and moves into transmission range of node 5, its hierarchical address changes from (13, 15, 15, 2) to (13, 5, 15, 2). This changed hierarchical address should be propagated along node 15 and node 2 until all the parent nodes of node 5 receives the update. As a result extra communication overhead is generated to nodes (node 15 and node 2) at high levels. If a node at high levels dominates a large number of nodes in its dominated zone, updating with its dominated members will eat up its resource when a lot of members move around. Moreover, nodes at the high level must deal with lots of other tasks, including route request, route reply, clusters formation and so on. Hence, they should not be troubled by frequent updates from its dominated members at low levels. It may be too expensive for nodes at high levels to update their dominated members each time

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

a member moves. Tasks caused by update of nodes at low levels are intended to be distributed into nodes other than those at high levels. If forwarding pointer is employed in the above scenario, nodes at high levels will not be troubled by the updating with nodes at low levels.

Thus, when a node(X) drifts from one zone to another zone, in order to reduce the communication overhead as well as the updating load at the previous dominator and the current dominator, the entry in X 's previous dominator is not updated, each time the mobile node X moves to a new location [JL95]. Instead, at X 's previous dominator, a forwarding pointer is set up to point to the current dominator in the new location. Now, routes required to a given node will first query the node's previous dominator to determine the first location at which the node was registered and then follow a chain of forwarding pointers to the user's current dominator. To reduce the time taken by the lookup procedure, the length of the chain of forwarding pointers is limited to grow up to a maximum value of M .

4.4.1.2 Maintenance with Forwarding Pointer

There are two steps to set up forwarding pointers:

- a) Request a forwarding pointer: When a level- K node finds that it changed level- K clusterhead, it appends the address of previous level- K clusterhead in a list of lost clusterheads and piggyback the list in level- K *Hello* message. By broadcasting this messages to its current level- K clusterhead, it requests its current level- K clusterhead to inform its previous level- K clusterhead to set up a forwarding pointer to current level- K clusterhead for itself.
- b) Reply the requested pointer: When a level- K clusterhead receives a level- K *Hello* message which contains a list of lost level- K clusterheads, it tries to inform the lost level- K clusterheads included in the list to set up forwarding pointers to

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

itself. It verifies that the lost level- K clusterhead is in its level- K tables (neighbor table and neighboring cluster table); and then sends messages to notify the lost clusterhead; if the verification fails, it updates hierarchical address of the drifting node with nodes at high levels. As a result the requested pointer is replied.

The list of lost level- K clusterhead of a node is reset to empty once it is sent out with level- K *Hello* messages. In implementation, K is usually not greater than 2 in order to limit the length of the chain of forwarding pointers.

One example is given to illustrate the above idea. For instance, when node 13 drifts from the cluster of node 15 to the cluster of node 5, it carries the information of the old cluster it belongs. When node 13 finds that it leaves the cluster of node 15 and joins the cluster of node 5, it piggybacks a request in level-0 *Hello* message and asks its new clusterhead (node 5) to notify its old clusterhead (node 15). After node 5 receives a request from node 13, it tries to reply this request to the lost clusterhead (node 15). When node 15 receives the request, it will pointer node 13's location to clusterhead 5. The next time the data sent to node 13 is still forwarded to node 15, which is the original cluster of node 13. And node 15 will use forwarding pointer to send data packets for node 13 to its new location. By setting up forwarding pointers at low levels, communication overhead generated by update hierarchical address with nodes at the high level is reduced. On the other hand, establishing forwarding pointers at low levels travels smaller hop distance.

4.4.2 Route Error Messages

The route error (RERR) message is sent whenever a link break causes one or more destinations to become unreachable from some of the node's neighbors. RERR messages are used to notify other nodes that the loss of that link has occurred. The RERR message indicates those destinations (possibly subnets) which are no longer reachable by way of the broken link. Route Error (RERR) messages may be either broadcast (if there are

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

many precursors), unicast (if there is only 1 precursor), or iteratively unicast to all precursors (if broadcast is inappropriate). A RERR message is sent by the reverse way to inform the source about the broken route, which may result a new route discovery at the source.

The RERR message should contains following fields:

- `Unreachable_Destination_Address`: The IP address of the destination that has become unreachable due to a link break.
- `Unreachable_Destination_Sequence_Number`: The sequence number for the destination listed in the previous Unreachable Destination IP Address field.

4.5 Performance Evaluations

To evaluate the performance of our hierarchical protocol, three sets of simulations have been performed.

The simulations were performed using the ns-2 network simulator with wireless extensions. The message exchange uses a MAC layer that realizes the default characteristics of the distributed coordination function (DCF) of IEEE 802.11. The channel capacity and transmission range are 2 Mbits/sec and 250m respectively. We varied the network size to be 50, 100, 200 and 300 in order to evaluate the scalability of the hierarchical protocol. We fixed the density of the network and keep the simulation area proportional to the number of nodes. The scenarios were randomly generated with input parameters such as maximum speed, pause time, number of nodes, area etc. The mobility model used was the Random Waypoint Model with pause time fixed to 10 seconds, while the max speed of mobility is 4m/sec. Constant Bit Rate (CBR) (512 bytes/packet/sec) applications were used to generate network data traffic. Each CBR sends four packets every second. Totally 10 source-destination pairs were spread randomly over the network. For each

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.6: Convergence time taken to establish a hierarchical system

point in the figure 100 simulation runs were carried out and the results were averaged. Since the confidence interval is very small in the simulation, we take it off from the figure for clarity.

We implement our hierarchical routing algorithm into three levels. At initiation, level-0 is partitioned by clustering and clusterhead at level-0 becomes level-1 nodes. In succession, nodes at level-1 is partitioned and level-1 clusterhead becomes level-2 nodes. Hence, a network system with three levels is established.

In the first set of simulations, the general characteristics of the hierarchical algorithm is evaluated.

The figure 4.7 shows the number of nodes at each level for different network size ranging from 50 to 300. The figure indicates that majority of nodes are level-0 nodes and the number of nodes at the higher level decrease sharply. This characteristic helps network with a large population to have limited nodes at the highest levels through

Figure 4.7: Number of nodes at each level

building hierarchical system. Moreover, large network requires greater number of levels in order to achieve small number of clusters at the high levels.

As mentioned in Section 2.6, typically three metrics are used to evaluate the performance of hierarchical routing protocols: convergence, stability and distribution. Figure 4.6 shows the convergence time to networks with different size. We can see that the convergence time is almost constant when network size increases. Since cluster is organized locally and no global information is required, convergence time of the network system is independent of network size. It is decided by the number of levels about to be built for the hierarchical system. More levels need to be built; more convergence time is required to establish the hierarchical system.

Assume M levels are set up in a hierarchical system and the time used to complete the establishment of the hierarchical system is equal to CT_M . Since the K th level begins to be formed when the formation of the $(K - 1)$ th level is completed, the time used to

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

establish the entire hierarchical system is equal to total of convergence time of each level. Thus, CT_M can be calculated as equation Eq. 4.1. In the equation C_i is the convergence time for the i th level.

$$CT_M = C_0 + C_1 + C_2 + C_3 + \cdots + C_{M-1} \quad (\text{Eq. 4.1})$$

We have discussed in section 4.1.4.2 that the clustering process is completed by exchanging *Hello* messages with neighborhood at each level. Therefore, C_i is decided by the cycle of broadcasting *Hello* messages at the i th level. In clustering algorithms *Hello* messages are generated periodically at each level. Because RCC is used to organize clusters at each level, two cycles of *Hello* messages are required to finish the clustering. At initiation, in order to partition the network into clusters and build the hierarchy rapidly *Hello* messages are produced with a short cycle in the first two round. After two round of broadcasting with short cycle, each node is included in at least one cluster and the formation of the hierarchy is converged. Then *Hello* messages at each level will be generated in longer cycles to avoid producing too much control overhead.

Assume that *Hello* messages are broadcasted with two short cycle of T_i at the i th level before convergence. Thus, CT_M can be calculated as: $CT_M = 2 * T_0 + 2 * T_1 + 2 * T_2 + 2 * T_3 + \cdots + 2 * T_{M-1}$.

In the simulation, nodes generate level-0 and level-1 *Hello* messages each 0.2 second and 0.4 second respectively in the first two round. By calculation, the convergence time for the hierarchical system is $CT_2 = 2 * T_0 + 2 * T_1 = 0.2 * 2 + 0.4 * 2 = 1.2$. We perform simulation testify this convergence time. We can see that in figure 4.6 simulated convergence time of different networks with different population are all close to calculated convergence time of 1.2. They are greater than 1.2 by reason of retransmission of *Hello* messages due to collision in wireless medium.

Our hierarchical routing protocol has a good convergence by using RCC as clustering algorithm. In section 3.1.6, we have explained that RCC converges faster than other

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.8: Convergence time of three clustering algorithms

clustering algorithms. Since RCC works with random process, each node need not know any information of neighboring nodes when it begins clustering. While in LCC and MOBIC, a node must obtain ID number and mobility of neighbors before it begins clustering. Two cycles are enough for RCC to complete the partition of one level, while more than two cycles are needed for LCC and MOBIC. Figure 4.8 shows the comparison of convergence time between RCC and other clustering algorithms(LCC, MOBIC), it outperforms others obviously.

Since 1.2 second is very short, all routing tasks can be kick off after nearly 1.2 second. We simulate the system more than 100 seconds and it can work well and stably. Hence, time spent to initiate the hierarchical structure will not occupy more than one percent of the whole routing process.

In this section, finally we evaluate the distribution of the hierarchical algorithm. As described in section 4.4.2, forwarding pointer is employed to distribute communication load of nodes at high levels and reduce communication overhead. In the simulation forwarding pointer is only applied in level-0. Figure 4.9 shows the number of the requested

Figure 4.9: Number of Employed Forwarding Pointer

pointer and number of replied pointer used to find the lost clusterheads. The figure indicates that 75% to 95% of lost clusterheads are found by forwarding pointers. This hit ratio of the lost clusterheads is quite high. Thus, for most of the node drifting, the forwarding pointer can be set up successfully. As a result, the communication overhead as well as the updating load at nodes at the high levels is reduced. Moreover, instead of updating hierarchical addresses(over 75% update of hierarchical addresses) from the bottom level to the top level and troubling a lot of clusterheads in a chain, communication overhead is reduced by communicating only with neighboring clusters and sending forwarding pointers.

4.6 Comparison with other routing protocols

4.6.1 Comparison with Flat Routing Protocols

In this section, we take insight into the scalability of RCHR with other classes protocols by increasing network sizes. We vary the network size to be 50, 100, 200, 300. We fix the density of the network and keep the simulation area proportional to the network

Figure 4.10: Delivery Ratio vs. Network Size (Low Mobility=2m/s)

size. We compare RCHR(our approach) with proactive protocol(DSDV) and reactive protocol(AODV) to evaluate the capacity of these protocols. Since AODV outperforms DSR, we compare RCHR with AODV instead of DSR as the representative of reactive protocols. Table 4.6 shows the parameters of our simulation.

Table 4.6: Simulation parameters

Simulator	NS-2
MAC	802.11DCF
Transmission Range	250m
Bandwidth	2Mb/s
Simulation area	500x500, 700x700, 1000x1000, 1200x1200
Number of nodes	50,100,200,300
Movement model	Random Waypoint
Max Speed	2 m/sec, 10m/sec
Pause Time	10s
Simulation Time	600s
Traffic	UDP/CBR

Comparison in Low Mobility

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.11: Control Packets vs. Network Size (Low Mobility=2m/s)

Figure 4.12: Delay vs. Network Size (Low Mobility=2m/s)

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.10 shows the delivery ratio as a function of the increasing network size in low mobility with the maximize speed as 2m/sec for three protocols. We can see that in low mobility RCHR has highest delivery ratio. And delivery ratio of DSDV decreases very fast when the network size increases. RCHR has slightly higher delivery ratio than AODV because RCHR can provide multiple routes with the hierarchical structure while AODV can only provide single route. Then, RCHR is more robust than AODV in mobile environment. DSDV has the worst performance because of the huge control overhead caused by exchanging routing table globally. When the number of nodes increases, the network will be congested by the huge control overhead.

Figure 4.11 shows the control packets as a function of the increasing network size in low mobility with the maximize speed as 2 m/sec for three protocols. The figure shows that when network size is small, AODV has less control overhead than other protocols because no overhead is generated by maintenance with AODV. When the network size is greater than 200. RCHR has less control overhead than other protocols. When network size increases, RCHR increases control overhead far less than AODV and DSDV. Therefore, the performance of RCHR will not degrade much when the network size increases.

Figure 4.12 shows the delay as a function of the increasing network size in low mobility with the maximum speed as 2 m/sec. The figure shows that DSDV has less delay than RCHR to different network size because DSDV always choose the shortest route and it does not has the route discovery process. RCHR has less delay than AODV.

Comparison in High Mobility

Figure 4.13 shows the delivery ratio as a function of the increasing network size in high mobility with the maximize speed as 10m/sec for three protocols. We can see that RCHR has slightly higher delivery ratio than AODV. Performance of RCHR and AODV

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.13: Delivery Ratio vs. Network Size (High Mobility=10m/s)

Figure 4.14: Control Packets vs. Network Size (High Mobility=10m/s)

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

does not drop much when the network size increases. In contrast, delivery ratio of DSDV decreases extremely fast. This performance degradation is caused by the congestion of the control overhead when the network size increases. Therefore, DSDV is not scalable for large network when network mobility is high.

Figure 4.14 shows the control packets as a function of the increasing network size in high mobility with the maximize speed as 10 m/sec for three protocols. Due to the frequent broadcast of distance vectors in high mobility, the congestion and collision of DSDV is very serious, resulting in tremendous loss of delivering data packets. Therefore, the control overhead of DSDV does not increase (actually reach its upper bound) as the network size increases due to the serious collision caused by periodic broadcasting, in this case its delivery ratio is very low and we can conclude DSDV is only applicable to small network. The figure shows that when network size is small AODV outperforms other protocols and has less control overhead. RCHR has higher control overhead than AODV but less than DSDV. When network size increases, RCHR increases control overhead far less than AODV.

Figure 4.15 shows the delay as a function of the increasing network size in high mobility with the maximum speed as 10 m/sec. The figure shows that DSDV has less delay than RCHR and RCHR has less delay than AODV.

Comparison with hybrid protocols

Although routing zones are utilized in hybrid routing protocols, hybrid routing protocols are not hierarchical schemes due to the same functionality for each node in the networking system. Since hybrid protocols are the combined schemes of proactive protocols and reactive protocols, they are of scalable problems of both proactive protocols and reactive protocols. For instance, ZRP uses on-demand type to route between zones; the number of zones increases when the network size increases, which results in same scalable problems of reactive protocols:

Figure 4.15: Delay vs. Network Size (High Mobility=10m/s)

- Flooding may incur exhaustion of network resources when the increase in the offered load aggravates and the network size expands.
- The maintenance of discovered routes also become hard to be managed when hop counts is large due to the long distance of routes.

4.6.2 Comparison with Hierarchical Routing Protocols

After compare RCHR with proactive protocols, reactive protocols and hybrid protocols, we preform simulation to compare RCHR with existing hierarchical routing schemes. The same simulation environment showed in table 4.6 is used. Here, RCHR is compared with HSR and CGSR introduced in section 2.5.

Firstly communication overhead of theses three hierarchical routing protocols are compared. Figure 4.16 shows the simulation results of three hierarchical routing algorithms. CGSR generates quite a lot of more overhead than HSR and RCHR. RCHR

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.16: Control Overhead vs. Network Size (Mobility=10m/s)

Figure 4.17: Delivery Ratio vs. Network Size (Mobility=10m/s)

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

Figure 4.18: Delay vs. Network Size (Mobility=10m/s)

generates less communication overhead than HSR. Since CGSR modifies the DSDV algorithm by using a cluster-head-to-gateway routing approach. Each node must keep a cluster member table where it stores the destination cluster head for each mobile node in the network. Cluster member tables are broadcasted by each node periodically using the DSDV algorithm (CGSR uses DSDV as the underlying routing scheme). The major advantage of CGSR is that it greatly reduces the routing table size comparing to traditional DV protocols(i.e. DSDV) due to only one entry for all nodes in the same cluster and it increases the robustness of routes by routing alternatively along clusterheads and gateways. However, it generates heavy communication overhead for broadcasting cluster member table in the type of DSDV. HSR is based on link state routing. At the first level of clustering, each node gathers the information of neighborhood and broadcasts it within the cluster. The clusterhead summarizes link state information within its cluster and broadcasts it to the neighboring clusterheads (via gateways connecting two neigh-

boring clusters). The knowledge of connectivity between neighboring clusterheads leads to the formation of level 2 clusters and in turn information of larger scope is broadcast in larger limited scope to form clusters at high levels. This broadcasting within limited scope reduce the communication overhead and avoid the global propagate of communication overhead. Different from HSR, RCHR uses multiple unicasts to establish clusters at high levels. Instead of broadcasting link state information to all gateways to communicate with neighboring clusters in HSR, only one gateway between two neighboring clusters is chosen to propagate link state information in RCHR. Hence, RCHR generates less communication overhead than HSR. On the other hand, since forwarding pointer is applied to reduce communication overhead caused by updating, RCHR outperforms HSR.

Secondly, we perform simulation to compare the delivery ratio of three hierarchical routing schemes. Figure 4.17 shows that RCHR has slightly higher delivery ratio than HSR, but much higher delivery ratio than CGSR. RCHR and HSR have more scalable than CGSR because they build more levels.

Finally, we perform simulation to compare the delay of three hierarchical routing schemes. Figure 4.18 that CGSR has smaller delay than the other two protocols since it uses DV scheme to update routing tables. RCHR and HSR have similar delay.

4.6.3 Comparison with Location Based Routing Protocols

Research [LJC⁺00] [BCSW98] [MWH01] has shown that location information can improve routing performance in ad hoc networks. However, considering that each node must refresh its location information(i.e. GLS [LJC⁺00], DLM [XLN01]) by communicating with its location service, when network size increases communication overhead used to update location information at location service will eat up the limited bandwidth. Hence, global updating is not a scalable way to gather routing information in

CHAPTER 4. A RANDOM CLUSTERING BASED HIERARCHICAL ROUTING ALGORITHM

location-based routing algorithm; while, in hierarchical routing scheme information updating only affects local area with limited scope. On the other hand, under mobile condition, in location-based algorithms if location information is inaccurate by the time the information is used(i.e. LAR [KV98], DREAM [BCSW98]), the routing scheme will not perform well; while, in hierarchical algorithms network is managed in local area; this characteristic makes the maintenance more easy and fast, as a result it is scalable for the large network. Comparing the hierarchical routing algorithm with location-based routing algorithm according to above discussion, the previous is more scalable for MANETs with large size.

4.7 Summary

In this chapter, we present a random clustering based hierarchical scheme(RCHR). We describe the formation and the maintenance of the hierarchical structure in detail. We apply forwarding pointers in the maintenance of the proposed hierarchical system in order to distributed communication task and speed up the convergence of the hierarchical structure. The protocol is evaluated via NS2 and compared with other existing routing protocols. Comparison between RCHR and other clustering routing algorithms shows that RCHR outperforms other clustering algorithms in scalability. Comparison between RCHR and proactive&reactive protocols shows that clustering algorithm RCHR is more scalable than flat routing protocols. Comparison between RCHR and CGSR&HSR shows that it outperforms CGSR&HSR with higher delivery ratio and lower control overhead. In conclusion, RCHR is a scalable and efficient hierarchical algorithm.

Chapter 5

Conclusion and Future Work

5.1 Conclusion&Contribution

In this thesis we address the scalability for MANET routing. We focus on the clustering and hierarchial routing approaches due to their good scalability. We envision the tradeoff between the delivery performance and the control overhead for clustering schemes. We propose two clustering approaches which enhance the delivery ratio without imposing higher control overhead and end-to-end delay. On the other hand, we provide insight of the design about the hierarchical routing scheme. We propose a hierarchial routing approach with complete hierarchical maintenance. In order to achieve better stability and lower control overhead, we employ random clustering and the forwarding pointer in hierarchical approach. Simulation shows that our hierarchical routing scheme has better stability and lower overhead than the traditional routing schemes for MANETs.

Our contributions are listed as follows:

- In Chapter 2, we provide the literature review of existing routing approaches and the scalable problem associate with these routing protocols. Based on the analysis of different categories of routing protocols, we find that the hierarchical routing approach is a promising way to achieve scalability.

CHAPTER 5. CONCLUSION AND FUTURE WORK

- In Chapter 3, we review the clustering techniques, which is the basic technique for hierarchical routing. By analyzing existing clustering schemes we give the various strength of each scheme. Moreover, we discuss the design principles for clustering techniques. Depending on the design principles, we propose cached cluster-based routing protocols and the random processing and mobility metric based adaptive routing protocols. These two approaches improve the clustering techniques by achieving lower control overhead and higher delivery ratio.
- Hierarchical structure is a good solution to provide scalability in large networking systems. In chapter 4 we present a random clustering based hierarchical routing algorithm(RCHR) and describe clustering and maintenance of the hierarchical algorithm in detail. We propose RCHR in order to overcome the problems with traditional hierarchical approaches. For good convergence of clustering, we make use of random processing to partition the network into clusters. Instead of broadcasting in multiple hops in traditional hierarchical algorithms, multiple hierarchies are set up through broadcasting in one hop and unicasting in multiple hops. Forwarding pointer is used in maintenance in order to distribute communication tasks and reduce communication overhead. Employment of forwarding pointer stabilizes the hierarchical structure and reduces the high cost consumed to update hierarchical addresses from low levels to high levels. Furthermore, our proposed hierarchical scheme can increase hierarchies along with the expanding of network size adaptively.

5.2 Future Work

Our future works consist of five aspects:

- Mobility Management and Clustering
- Position-based Forwarding Pointer

- Load Balance of Hierarchical Protocols
- Using Different Power Level at Different Levels to Achieve Power Efficiency and Scalability
- Simulation with Different Network Size and Different Mobility Models

5.2.1 Mobility Management and Clustering

MANETs are not inherently scalable for number of nodes, control overhead, degree of mobility, network density and traffic load. In fact, no flat networks are scalable, include MANETs. Furthermore mobility in MANETs exacerbates this problem due to ephemeral associations between nodes. So mobility management is also a scalable issue in MANETs.

The mobility management [PGHC99] [HGPC99] [ZD97] is that a set of mechanisms provide a time-varying mapping between the identifier of a mobile node and its location to the network structure. Different mobility patterns affect the various protocols in different ways. In particular, the performance of routing algorithms is influenced by the choice of mobility pattern.

In MANETs if we can predict the mobility pattern of nodes, the moving direction and the drifting zone of a mobile node can be predicted. Hence, when obtaining approximate information of a node, depending on its mobility pattern, we can restrict the searching area, i.e. searching directionally or searching in a determined area. This scheme reduces the broadcast overhead caused by discovery process. On the other hand if we can predict the mobility pattern, we can update and organize the network more adaptively and purposively according to the mobility of a node. The information of the nodes with high mobility will be updated with high frequency, while, the information of the nodes with low mobility will be updated with low frequency. Thus, excessive communication overhead can be avoided and it makes the network more scalable.

Mobility management in ad hoc wireless networks faces many challenges [HKG⁺01]. The main problem is the prediction of the mobility pattern of the network topology. In order to keep accurate routes, the routing protocols must dynamically readjust to such changes. Different mobility patterns have in general different impact on a specific network protocol or application. The mechanism to adapt network behaviors referring to the mobility pattern is also important. These problems will be focused in future work.

5.2.2 Combination of Position-based Schemes with Clustering Techniques

Position based routing strategies, since they do not require the exchange of routing tables, are especially attractive in highly mobile environments where topological changes are frequent and routing tables become obsolete very quickly [HL85].

In general using geographic position as a means of routing/mobility management, has the following advantages [XPA00]:

- (i) It makes feasible the realization of a flat routing algorithm.
- (ii) It minimize the amount of initial manual configuration required to set up the network, thus making the network completely self-organizing.
- (iii) It makes feasible the process of routing packets with geographic destination.
- (iv) It reduces for reaching routing overhead propagation by making the overhead propagation distance directly proportional to the amount of change in a node's position.
- (v) It makes feasible the use of motion prediction techniques, which can be used to provide additional reduction of position update overhead.

Thus, we intend to integrate geographic position based mechanism into our clustering technique and make use of the approximate location of nodes in the network

CHAPTER 5. CONCLUSION AND FUTURE WORK

for geographic routing. We plan to combine the geographic techniques and clustering mechanisms together. Firstly the whole network is divided into clusters using clustering algorithm, and in every cluster a leader is chosen which maintains the cluster and acts as the location server to distribute location to cluster members. And all the leaders of clusters compose a dominating set to connect all the clusters in the network. Within a cluster a set of nodes rely on the position information gathered by a leader thus reducing the overall cost of flooding the location information of all the nodes into the network. When a route is required, the source node sends a request packet to its leader, leader queries the destination's location through the dominating set of leaders by sending the request packet to neighboring clusters. The request carries the position of the source's leader. When the destination's leader receives the request, it will send the reply packet carrying its location to the source's leader by geographic forwarding. When the source's leader receives the reply packet, it relays it to the source. Therefore, the source node uses the geographic forwarding to transmit data packets. The procedure is illustrated in Figure 5.1. This approach relieves the overhead of cluster leaders. The shortest path is also achieved through geographic forwarding.

Due to the provision of the location of any node specified by the group location server, the nodes will know the location of the destination, and can reduce routing overhead and delay in finding the route since directional forwarding takes the place of flooding to the entire network. On the other hand with the location of a mobile node, it is easy to restrict the searching area when it moves away.

5.2.3 Load Balance of Hierarchical Protocols

For hierarchical routing protocols while the hierarchical structure is set up, particular leader is chosen within each cluster. Since the leader carries the maintenance within cluster and manages the communication with other clusters, it keeps on working. Thus,

Figure 5.1: Location based clustering approach

leader nodes consume power faster than other nodes. However, in real life application all nodes have uniform resource, if some nodes persist as leaders for a long time, they will run out of power because of frequent communication within and out of clusters. In order to prevent a node from exhaustion of power, we intend to balance the resource consumption evenly in the entire network, which also indicates that each node has the same probability to be a leader in hierarchical routing schemes. In Section ??, we propose an algorithm to maintain a load balanced system by choosing the node which ever acts less time for a clusterhead to be the leader. Next step we plan to choose the cluster member to take the place of leader in the same cluster when a leader persist too long within a cluster. After the replacement, two properties are desired:

- the new leader almost has the same covering area with the old leader
- the replacement will cause as few as cluster members to change status

5.2.4 Using Different Power Level at Different Levels

In clustering routing algorithms, we augment the connectivity between the nodes of the ad hoc network using higher power links to interconnect cluster heads. For the purposes of routing, the links between cluster heads are considered identical to the links between the regular nodes. The route discovery procedure used is identical to flat routing. Route request messages are flooded along the links between regular nodes as well as the links between cluster heads.

We assume that regular nodes are allowed to connect to their cluster heads only. All route discovery processes take place through the cluster heads. The procedure for discovering the route between a pair of nodes (source and destination) is as follows. The Source sends a route request message to its cluster head. If the destination is not within the same cluster, the clusterhead of the source floods this route discovery message to all cluster heads that it has connectivity to. The clusterhead connected to the destination sends a route reply message along the route of the first route request message that it receives.

Similarly, we use higher power links at high levels, and the nodes at the low levels use lower power.

5.2.5 Simulation with Different Network Size and Different Mobility Models

Because NS2 is not a scalable simulator, it is hard to simulate a network containing more than 500 hundred nodes with NS2. By now many papers have investigated the performance of routing protocols based on the 2Mbps IEEE 802.11 WLAN protocols. In their simulation, the number of nodes in the network usually ranges from 100 to 400 and the traffic generation per node ranges from 2 to 4 packets per seconds. With the limitation of the NS2 simulator, in the simulations of the thesis the number of nodes

CHAPTER 5. CONCLUSION AND FUTURE WORK

only ranges from 100 to 400 nodes. In the future, we expect to extend the simulation for network with larger size on other scalable simulators. For all simulations in this thesis, only Random Way-point Mobility Model is used because it is a general model. In simulation it is likely to find out some unexpected scenarios that are explore in analysis with general models[FP01]. However, it is not realistic to use general models to dig out specific problems for different applications of MANETs. For example, group mobility model is preferred for military scenarios; static model should be used for sensor scenarios. We will study different mobility model for simulation in the future work.

References

- [BC98] G. Bianchi and A. T. Campbell. A programmable mac. In *Proceedings the 98th Conference on Universal Personal Communication*, 1998.
- [BC04] J. Boleng and T. Camp. Adaptive location aided mobile ad hoc network routing. In *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference (IPCCC '04)*, Phoenix, Arizona, April 15-17 2004.
- [BCS99] S. Basagni, I. Chlamtac, and V. Syrotiuk. Geographic messaging in wireless ad hoc networks. In *Proceedings of the 49th IEEE International Vehicle Technology Conference*, Houston, TX, USA, 1999.
- [BCSW98] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom'98*, Dallas, TX, Oct. 1998.
- [BE81a] D. J. Baker and A. Ephremides. A distributed algorithm for organizing mobile radio telecommunication networks. In *Second International Conference on Distributed Computer Systems*, pages 476–483, Apr. 1981.
- [BE81b] D. J. Baker and A. Ephremides. Multicluser, mobile, multimedia radio network. *IEEE Transactions on Communications*, 11(29):1694–1701, November 1981.
- [Bia00] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18:535–547, 2000.

REFERENCES

- [BJKM04] Nah-Oak Song Byung-Jae Kwak and Leonard E. Miller. On the scalability of ad hoc networks: a traffic analysis at the center of a network. In *2005 IEEE Wireless Communications and Networking Conference*, 2004.
- [BJM03] J. Broch, D.B. Johnson, and D.A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. In *draft-ietf-manet-dsr-09.txt*, Apr. 2003.
- [BKL01] P. Basu, N. Khan, and T. Little. A mobility based metric for clustering in mobile ad hoc networks. In *Proceedings of International Workshop on Wireless Networks and Mobile Computing(WNMC2001)*, Scottsdale, Arizona, Apr. 2001.
- [BOT01] B. Bellur, R. G. Ogier, and F. L. Templin. Topology broadcast based on reverse-path forwarding (tbrpf) internet draft. *draft-ietf-manet-tbrpf-01.txt*, June 2001.
- [CG97] C. C. Chiang and M. Gerla. Routing and multicast in multihop, mobile wireless networks. In *Proceedings of IEEE ICUPC'97*, San Diego, CA, Oct. 1997.
- [CHH01] S. Capkun, M. Hamdi, and J. Hubaux. Gps-free positioning in mobile ad-hoc networks. In *Hawaii International Conference on System Sciences*, 2001.
- [CHWM97] C.C.Chiang, H.Wu, W.Liu, and M.Gerla. Routing in clustered multihop mobile wireless networks with fading channel. In *Proceedings of SICON97*, Singapore, April 1997.
- [CML99] D. H. Cansever, A. M. Michelson, and A. H. Levesque. Quality of service support in mobile ad-hoc ip networks. In *Proceedings of MILCOM*, 1999.
- [EBS03] E.Shih, V. Bahl, and M. Sinclair. Reducing energy consumption of wireless, mobile devices using a secondary low-power channel. In *MIT Laboratory for Computer Science*, 2003.
- [FP01] Sally Floyd and Vern Paxson. Difficulties in simulating the internet. *IEEE/ACM Trans. Netw.*, 9(4):392–403, 2001.

REFERENCES

- [GLA93] J. J. Garcia-Lunes-Aceves. Loop-free routing using diffusing computations. *IEEE/ACM Transactions on Networking*, 1:130 – 141, 1993.
- [GT95] M. Gerla and J. T. C. Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 3(1):255–265, 1995.
- [GWFM02] M. Gerharz, C. Waal, M. Frank, and P. Martini. Link stability in mobile wireless ad hoc networks. In *Proceedings of the 27th IEEE Conference on Local Computer Networks*, Tampa, Florida, 2002.
- [Haa97] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of ICUPC'97*, San Diego, CA, Oct.12 1997.
- [HGPC99] X. Hong, M. Gerla, G. Pei, and C. C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, 1999.
- [HKG⁺01] X. Y. Hong, T. J. Kwon, M. Gerla, D. L. Gu, and G. Y. Pei. A mobility framework for ad hoc wireless networks. *Lecture Notes in Computer Science*, 1987:185–196, 2001.
- [HL85] T-C Hou and O. K. Li. Position updates and sensitivity analysis for routing protocols in multihop mobile packet radio networks. In *Proceedings IEEE Globecom*, 1985.
- [HXG02] X. Y. Hong, K. X. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network Magazine*, 16:11–21, 2002.
- [ICP⁺99] A. Iwata, C-C Chiang, G. Y. Pei, M. Gerla, and T-W Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, Aug. 1999.
- [JL95] R. Jain and Y-B. Lin. An auxiliary user location strategy employing forwarding pointers to reduce network impacts of pcs. *Wireless Networks*, 1:197–210, 1995.

REFERENCES

- [JSAC01] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7, 2001.
- [Kes97] S. Keshav. *An Engineering Approach to Computer Networking: ATM networks, the Internet, and the telephone network*. Chapter 11, Addison Wesley Inc, 1997.
- [KF97] L. Kleinrock and K. Faroukh. Hierarchical routing for large networks. 1997.
- [KK00] B. Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 243–254, Boston, MA, August. 2000.
- [KK03] V. Kawadia and P.R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of Infocom 2003*, 2003.
- [KKRM01] U. Kozat, G. Kondylis, B. Ryu, and K. Marina. Virtual dynamic backbone for mobile ad hoc networks. In *IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001.
- [KSU99] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computer Geometry*, Vancouver, Canada, Aug. 1999.
- [KV98] Y. B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, Oct. 1998.
- [KVCP97] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, 27:49 – 64, 1997.
- [LH00] B. Liang and Z. J. Haas. Virtual backbone generation and maintenance in ad hoc network mobility management. In *proceedings of InfoCom 2000*, 2000.

REFERENCES

- [LJC⁺00] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, Massachusetts, 2000.
- [LPJW01] X.-Y. Li and Y. Wang P.-J. Wan. Power efficient and sparse spanner for wireless ad hoc networks. In *IEEE International Conference on Computer Communications and Networks*, 2001.
- [LTG98] S. J. Lee, C. K. Toh, and M. Gerla. Throughput evaluation of the iee 802.11 distributed coordination function. In *Proceedings of 5th International Workshop on Mobile Multimedia Communications*, 1998.
- [LTG99] S. J. Lee, C. K. Toh, and M. Gerla. Performance evaluation of table-driven and on-demand ad hoc routing protocols. In *Proceedings of IEEE PIMRC99*, Osaka, Japan, 1999.
- [MCE98] A. Michail, W. Chen, and A. Ephremides. Distributed routing and resource allocation for connection-oriented traffic in ad-hoc wireless networks. In *Proceedings of the Conference on Information Sciences and Systems*, 1998.
- [MWH01] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad-hoc networks. *IEEE Network*, 15:30–39, 2001.
- [Par94] A. K. Parekh. Selecting routers in ad hoc wireless networks. In *Proceedings of IEEE International Telecommunications Symposium*, Boston, MA, Aug. 1994.
- [PB94] C. E. Perkins and P. Bhagwar. Highly dynamic destination-sequenced distance-vector. In *Proceedings of Sigcom'94, ACM*, Aug. 1994.
- [PBRD03] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc on-demand distance vector (aodv) routing draft-ietf-manet-aodv-13.txt. *retrieval from <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt> on June, 2004*, 2003.

REFERENCES

- [PGHC99] G. Pei, M. Gerla, X. Hong, and C. C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proceedings of IEEE WCNC99*, New Orleans, LA, 1999.
- [PH02] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Antonio, TX, 2002.
- [PH03] A. Perrig and Y.-C. Hu. Secure routing in ad hoc networks. In *the Ninth Annual International Conference on Mobile Computing and Networking*, San Diego, California, 2003.
- [PR99] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, February 1999.
- [Pro] CMU Monarch Project. Wireless and mobility extensions to the ns-2 network simulator. retrieval from <http://monarch.cs.cmu.edu/cmu-ns.html> on November, 2002.
- [Rpp96] T. Rppaport. Wireless communications. *Prentice Hall*, 1996.
- [RT99] E. Royer and C.-K. Toh. A review of current routing protocols for ad hoc wireless networks. *IEEE Personal Communications*, pages 46–55, 1999.
- [SS02] R. K. Ghosh S. Srivastava. Cluster based routing using a k-tree core backbone for mobile ad hoc networks. In *Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, 2002.
- [TCG97] J. Tsai, T. Chen, and M. Gerla. Qos routing performance in multihop, multimedia, wireless networks. In *Proceedings of IEEE ICUPC*, 1997.
- [TCN02] B. Williams L. Wilcox T. Camp, J. Boleng and W. Navidi. Performance evaluation of two location based routing protocols. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies(INFOCOM)*, 2002.

REFERENCES

- [TDLR02] D. Towsley, C. Diot, B. N. Levine, and L. Rizzo. Network support for multicast communication. *IEEE Journal on Selected Areas in Communications*, 20, 2002.
- [XG02] K. X. Xu and M. Gerla. A heterogeneous routing protocol based on a new stable clustering scheme. In *Proceedings of IEEE Milcom2002*, pages 838–843, Boston, MA, Oct. 2002.
- [XLN01] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, Tampa, Florida, November 2001.
- [XPA00] S. Xu, S. Papavassiliou, and K. Amouris. On the performance of a scalable-tier position based routing protocol for mobile ad hoc wireless networks. In *Proceedings of Fifth IEEE Symposium on Computers and Communications*, July 2000.
- [YF03] Dahlia Malkhi Y. Fernandess. Secure routing in ad hoc networks. In *the Ninth Annual International Conference on Mobile Computing and Networking*, San Diego, California, 2003.
- [Yu00] J. Yu. Scalable routing design principles. In *Internet RFC/STD/FYI/BCP Archives RFC2791* (retrieval from <http://www.faqs.org/rfcs/rfc2791.html> on June, 2004), July 2000.
- [YY04] P. Yang and K. C. Yow. Passive cached clustering routing for manets. In *IEEE international conference on information and communication technologies:from theory to applications*, Damascus, Syria, 2004.
- [ZD97] M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility pattern. *IEEE Journal on Selected Areas in Communications*, 15:1239C1252, 1997.