

ABSTRACT

HU, SHUANG. Multicast Routing Protocols in Mobile Ad Hoc Networks. (Under the direction of Professor Arne A. Nilsson.)

The demand to exchange digital information using wireless communication system is growing fast nowadays. When there are independent mobiles users participating in sharing resources or interacting with each other, like wireless video sharing and interactive games, and no centralized connectivity is used, the application can be treated as the application of Mobile Ad Hoc Network. Two multicast routing protocols for Mobile Ad Hoc Networks (MANET) were proposed.

The multicast routing protocol without Quality-of-Service (QoS) support aimed at improving packet delivery ratio. Data delivery tree was built on top of the mesh structure. The mesh structure was updated when there was no data delivery tree to cover all receivers. The branch of the tree was selected based on the link stability prediction. To do so, Packet Delivery Prediction (PDP) was introduced. Models like battery model and route selection model were used at every node in the system to predict the stability of the node/link and make data delivery decision according to the scheme defined.

The multicast routing protocol with QoS support aimed at providing Quality of Service to real-time applications. Bandwidth and delay were the two major considerations for the proposed QoS-enabled protocol. The nodes in the system monitored the bandwidth consumption of itself as well as the neighboring nodes. Real-time traffic flow was admitted or rejected at the source based on the bandwidth availability feedback of the intermediate nodes. For admitted real-time traffic flow, the resource was set aside for a specific period of time, which was designed to take into account the mobility and packet inter-arrival delay requirement.

Simulation showed the QoS-disabled as well as QoS-enabled multicast routing protocol can achieve their major design goal. The QoS-enabled routing protocol can admit or reject real-time traffic flow according to available bandwidth at the nodes. Real-time traffic

was better served using QoS-enabled multicast routing protocol than using QoS-disabled counterpart.

Multicast Routing Protocols in Mobile Ad Hoc Networks

by
Shuang Hu

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh, North Carolina
2008

APPROVED BY:

Dr. Arne A. Nilsson
Chair of Advisory Committee

Dr. Michael Devetsikiotis

Dr. Mo-Yuen Chow

Dr. George N. Rouskas

DEDICATION

This dissertation is dedicated to my parents, Wuming Hu and Caixian Zuo; my sister,
Hong Hu.

BIOGRAPHY

Shuang Hu was born in Wuhan, China. She received the B.S degree majoring in Electrical Engineering and minoring in Computer Science and Engineering in 2000, and Master degree in Electrical Engineering in 2003, all from Huazhong University of Science and Technology, Wuhan, China. She is currently working towards the Ph. D. degree in Computer Engineering in Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, North Carolina.

Her research interests include multicast routing protocols in Mobile Ad Hoc Network (MANET), Quality-of-Service (QoS) support for real-time traffic using multicast routing protocol, and performance evaluation and analysis of wireless networks.

ACKNOWLEDGMENTS

This dissertation could not have been written without Dr. Arne A. Nilsson, who serves as my advisor, and guides, encourages and supports me throughout the course of the work. I would also like to thank Professor Michael Devetsikiotis, Professor Mo-Yuen Chow and Professor George N. Rouskas for offering their time, help and advice for my work.

I also appreciate each and every one of my friends who share their time and experience with me.

Finally, I would like to express my deepest appreciation to my parents, Wuming Hu and Caixian Zuo, who have been providing encouragement, suggestion and support as always. Special thanks should be given to my sister, Hong Hu, who is especially insightful and supporting. My gratitude is also extended to all my family members for their encouragement and love.

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
1. Introduction.....	1
1.1 Features of MANET	2
1.2 Multicasting in MANET	4
1.3 Multicast Routing Protocol Classification in MANET.....	5
1.4 Issues in Multicast in a Mobile Ad Hoc Network.....	7
1.5 Common Flaws in Performance Study in MANET	12
1.6 Contributions.....	12
2. Basic Multicast Routing Protocols	15
2.1 Mobility Models.....	15
2.1.1 Entity Models.....	16
2.1.1.1 Random Walk Model.....	16
2.1.1.2 Random Waypoint (RWP) Model.....	17
2.1.1.3 Random Direction (RD) Model.....	18
2.1.2 Group Model.....	19
2.1.2.1 Reference Point Group (RPGM) Model.....	19
2.1.3 Mobility Model Summary.....	21
2.2 Basic Routing Protocols.....	21
2.2.1 Flooding	22
2.2.2 Tree-Based Protocol: Multicast Ad-hoc On-Demand Distance Vector (MAODV).....	22

2.2.3 Mesh-Based Protocol: On-Demand Multicast Routing Protocol (ODMRP)....	27
2.2.4 Hybrid Protocol: Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR).....	31
2.3 Conclusion	32
3. Evaluation Metrics and Related Protocols	33
3.1 Data Overhead	33
3.1.1 Data Overhead Reduction for Tree-Based Protocol	33
3.1.2 Data Overhead Reduction for Mesh-Based Protocol.....	35
3.1.3 Data Overhead Reduction for Hybrid Protocol	37
3.2 Control Overhead.....	39
3.2.1 Preemptive Multicast Routing (PMR)	39
3.2.2 Neighbor Aware Multicast Routing Protocol (NAMP).....	41
3.3 Packet Delivery Ratio	44
3.3.1 Tree-based Protocol: Overlay Multicast based on Heterogeneous Forwarding (OMHF).....	45
3.3.2 Robust Multicasting in Ad Hoc Networks Using Trees (ROMANT)	49
3.3.3 Mesh-based Protocol: Reliable On Demand Multicast Routing Protocol (R- ODMRP).....	50
3.4 End-to-End Delay	52
3.4.1 Rate-adaptive Multicast (RAM)	52
3.4.2 A Priority Scheduler Using Fuzzy Logic.....	54
3.5 Conclusion	55
4. On-Demand Hybrid Multicast Routing Protocol.....	58
4.1 Overview.....	59
4.2 Routing Protocol Operation	60
4.2.1 Mesh Construction.....	60

4.2.2 Data Forwarding	62
4.2.3 Mobility Prediction	64
4.2.4 Battery Model	65
4.2.5 Route Selection Model (Packet Delivery Prediction).....	66
4.2.5.1 Fixed Prediction Threshold Based Data Forwarding Route Selection.....	67
4.2.5.2 Adaptive Prediction Threshold V.S. Fixed Prediction Threshold	68
4.2.6 Data Retransmission (optional).....	70
4.2.7 Mesh Maintenance	73
4.3 Message Format	74
4.3.1 Mesh Construction Message Format	74
4.3.2 Backup Node Selection and Data Retransmission (optional)	77
4.4 Data Node Structures and Routing Table	82
4.5 Summary	83
5. QoS-disabled Protocol Evaluation.....	84
5.1 Simulation Tools and Scenario Setup.....	84
5.1.1 Simulating Mobile Ad-hoc Networks.....	84
5.1.2 Scenario Setup	86
5.2 Comparison Evaluation with MAODV and ODMRP	89
5.2.1 Packet Delivery Ratio	89
5.2.1.1 Mobility Speed.....	89
5.2.1.2 Number of Group Members.....	91
5.2.2 Data Overhead	92
5.2.2.1 Mobility Speed.....	92
5.2.2.2 Number of Group Members.....	93
5.2.3 Control Overhead.....	93
5.2.3.1 Mobility Speed.....	93
5.2.3.2 Number of Group Members.....	95

5.2.4 Average End-to-End Delay	96
5.2.4.1 Mobility Speed.....	96
5.2.4.2 Number of Group Members.....	97
5.3 Summary	97
6. Quality of Service (QoS) Multicast Routing in MANET.....	99
6.1 Overview.....	100
6.1.1 QoS Support in MANETs.....	100
6.1.2 QoS-enabled Multicasting in MANET	102
6.1.3 Two Potential Bandwidth-violation Problems.....	103
6.2 MAC Layer Modification	105
6.3 QoS Operation Modification to On-Demand Hybrid Multicast Routing Protocol. .	
.....	106
6.3.1 Neighborhood Maintenance.....	106
6.3.2 Qos Mesh Construction Considering Bandwidth Violation Problems	106
6.3.3 Traffic/Data Forwarding	110
6.3.4 Mesh Maintenance	112
6.4 Bandwidth Control.....	112
6.4.1 Bandwidth Violation Avoidance.....	112
6.4.2 Node's Bandwidth Availability Estimation	113
6.4.3 Flow's Bandwidth Consumption Estimation	116
6.4.4 Traffic Admission/ Bandwidth Reservation	118
6.4.5 Bandwidth Releasing/ Route Break Detection	118
6.5 Message Modification.....	119
6.5.1 Mesh Construction Message Format	119
6.6 Data Node Structures and Routing Table	124
6.7 Summary	126

7. Quality-of-Service Multicast Routing Performance Simulation and Evaluation. . .	127
.....	
7.1 Scenario Setup	127
7.2 Performance Evaluation.....	130
7.2.1 Performance Metrics	130
7.2.2 Performance Analysis	130
7.3 Simulation Comparison of QoS-enabled and QoS-disabled Multicast Routing Protocols	133
7.3.1 Packet Delivery Ratio	134
7.3.1.1 Mobility Speed.....	134
7.3.1.2 Number of Group Members.....	136
7.3.2 Average End-to-End Delay.....	137
7.3.2.1 Mobility Speed.....	137
7.3.2.2 Number of Group Members.....	138
7.3.3 Control Overhead.....	140
7.3.3.1 Mobility Speed.....	140
7.3.3.2 Number of Group Members.....	141
7.4 Summary	141
8. Conclusions and Future Works	143
8.1 Conclusions.....	143
8.2 Future Works	144
BIBLIOGRAPHY.....	146

LIST OF TABLES

Table 1.1 Selective Qualitative Comparison of Wired and Wireless Multicast	11
Table 6.1 Join-Query-Probe message format.....	120
Table 6.2 Join-Reply message format.....	121
Table 6.3 Mesh_Refresh_Request message format	123
Table 7.1 Simulation Parameters	129

LIST OF FIGURES

Figure 1.1 Multi-hop Path in MANET	3
Figure 1.2 Example of peer-to-peer communication in MANETs	9
Figure 1.3 Example of remote-to-remote communication in MANETs.....	10
Figure 1.4 Example of mixed communication in MANET	10
Figure 2.1 Traveling pattern of a mobile node using the Random Waypoint Mobility Model	17
Figure 2.2 Movements of three mobile nodes using RPGM	20
Figure 2.3 Tree Creation in MAODV.....	27
Figure 2.4 Mesh Formation in ODMRP	29
Figure 2.5 Reply Forwarding.....	30
Figure 2.6 Meshing Configuration.....	31
Figure 3.1 A logical tree in the routing mesh	40
Figure 3.2 Formation of Secondary Forwarder List (SFL).....	43
Figure 3.3 Link Failure and Link Recovery in NAMP.....	44
Figure 3.4 Path setup via modified expanding ring search.....	47
Figure 4.1 Procedure to select Backup Node for the data retransmission (optional)	63
Figure 4.2 Data Retransmission Procedure (optional).....	72
Figure 5.1 Packet delivery ratio as a function of mobility speed.....	89
Figure 5.2 Packet delivery ratio as a function of number of group members.....	91
Figure 5.3 Data overhead as a function of mobility speed	92
Figure 5.4 Data overhead as a function of number of group members.....	93
Figure 5.5 Control overhead as a function of mobility speed.....	94
Figure 5.6 Control overhead as a function of number of group members	95

Figure 5.7 Average end-to-end delay as a function of mobility speed	96
Figure 5.8 Average end-to-end delay as a function of number of group members	97
Figure 6.1 Nodes in FQMM	101
Figure 6.2 An example of HRP problem	104
Figure 6.3 An example of HMRP problem.....	105
Figure 6.4 Mesh construction: request phase and reply phase.....	107
Figure 6.5 Traffic mark phase.....	110
Figure 6.6 New RT traffic flow enters the network during traffic forwarding phase.....	111
Figure 6.7 Traffic load of Node I.....	114
Figure 6.8 Traffic aggregation of new real-time flow	117
Figure 7.1 Traffic Admission Ratio versus Mobility Speed.....	131
Figure 7.2 Average End-to-End Delay versus Mobility Speed	132
Figure 7.3 Packet Delivery Ratio versus Mobility Speed.....	133
Figure 7.4 Packet Delivery Ratio versus Mobility Speed.....	135
Figure 7.5 Packet Delivery Ratio versus Number of Group Members.....	136
Figure 7.6 Average End-to-End Delay versus Mobility Speed	137
Figure 7.7 Average End-to-End Delay versus Number of Group Members	139
Figure 7.8 Control Overhead versus Mobility Speed	140
Figure 7.9 Control Overhead versus Number of Group Members	141

Chapter 1

Introduction

In wireless communication systems, there's a need for the rapid deployment of independent mobile users. Significant examples include establishing survivable sufficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks. The demand to exchange digital information outside the typical wired office environment is also growing. For example, a class of students may need to interact during a lecture; or business associates serendipitously meeting in an airport may wish to share files. Such network scenarios cannot rely on centralized and organized connectivity, and can be conceived as applications of Mobile Ad Hoc Networks. Each of the devices used by these information producers and consumers can be considered a node in an ad-hoc network.

Mobile Ad Hoc Networks (MANETs) are specific network configurations that appear in the context of ubiquitous computing and proliferation of portable computing devices. A mobile ad-hoc network (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links [19]. The topology of mobile ad-hoc networks is arbitrary [17]. In MANET, the routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.

Mobile ad-hoc networks (MANETs) require no fixed infrastructure or central administration. Mobile nodes in an ad hoc network work not only as hosts but also as routers, and communicate with each other via packet radios. Many applications like audio- and video streaming or whiteboard require the ability to transmit data to a group of receivers. As stated above, applications of MANET can be performed by personal area networking such as cellular phone and laptop, by emergency operations such as disaster relief, by civilian

environments such as meeting rooms and sports stadiums, and by military environments such as soldiers, tanks, and planes. This thesis focuses on providing efficient multicast communications on such networks.

1.1 Features of MANET

A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communication devices), which are simply referred to as “nodes”, and are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices, and there may be multiple hosts per router.

Typically, MANET has the following features [14] [18]:

- 1) Autonomous terminal. In MANET, each mobile terminal is an autonomous node, which may function as both a host and a router. In other words, besides the basic processing ability as a host, the mobile nodes can also perform switching functions as a router. So usually endpoints and switches are indistinguishable in MANET.
- 2) Distributed operation. Since there is no background network for the central control of the network operations, the control and management of the network is distributed among the nodes. The nodes involved in a MANET should collaborate amongst themselves and each node acts as a relay as needed, to implement functions like security and routing.
- 3) Multi-hop routing. Basic types of ad hoc routing algorithms can be single-hop and multi-hop, based on different link layer attributes and routing protocols. Single-hop MANET is simpler than multi-hop in terms of structure and implementation, with the cost of lesser functionality and applicability. In single-hop routing, nodes can communicate directly with each other when they are within transmission range of each other; while in multi-hop routing, ad hoc networks must support communication between nodes that are only indirectly connected by a series of wireless hops through other nodes. When delivering data packets from a source to its destination out of the direct wireless transmission range, the packets

should be forwarded via one or more intermediate nodes. In Figure 1.1, Node A communicates with node C over a multi-hop path, where they must enlist the aid of node B to relay packets between them in order to communicate. The large circles denote each node's transmission range. In this case, A's circle does not cover C.

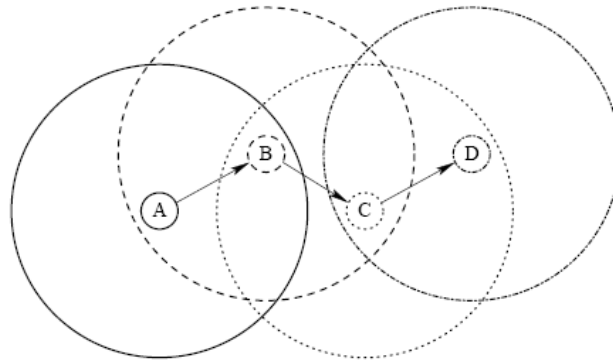


Figure 1.1 Multi-hop Path in MANET

4) Dynamic network topology [15]. Since the nodes are mobile, the network topology may change rapidly and unpredictably, and the connectivity among the terminals may vary with time. MANET should adapt to the traffic and propagation conditions as well as the mobility patterns of the mobile network nodes. The mobile nodes in the network dynamically establish routing among themselves as they move around, forming their own network on the fly. Moreover, a user in the MANET may not only operate within the ad hoc network, but may require access to a public fixed network (e.g. Internet).

5) Fluctuating link capacity [16]. Wireless links will continue to have significantly lower capacity than their wired counterparts. In addition, after accounting for the effects of multiple access, fading, noise, and interference conditions, etc, the realized throughput of wireless communications is often much less than a radio's maximum transmission rate. The nature of high bit-error rates of wireless connection might be more profound in a MANET. One end-to-end path can be shared by several sessions. In some scenarios, the path between any pair of users can traverse multiple wireless links and the links themselves can be heterogeneous.

6) Light-weight terminals. In most cases, the MANET nodes are mobile devices with less CPU processing capability, small memory size, and low power storage. Such devices need optimized algorithms and mechanisms that implement the computing and communicating functions.

In addition, some envisioned networks like mobile military networks or highway network may be relatively large, consisting of tens or hundreds of nodes per routing area. The need for scalability is not unique to MANETS. However, in light of the preceding characteristics, the mechanisms required to achieve scalability.

These features create a set of underlying assumptions and performance concerns for protocol design which extend beyond those guiding the design of routing within the higher-speed, semi-static topology of the fixed Internet.

1.2 Multicasting in MANET

In recently years, many research results have been proposed for group communication or multicast communication in MANET. Multicasting is the transmission of datagrams to a group of hosts identified by a single destination address and hence is intended for group-oriented computing [26]. In MANET, multicast can efficiently support a variety of applications that are characterized by close collaborative efforts. A multicast packet is typically delivered to all members of its destination group with the same reliability as regular unicast packets. Think about a scenario where a user is walking with a handheld device or waiting for a flight in an airport terminal. He/she does not know about his/her neighbor, and switches on the handheld device and tries to scan the network to detect if someone would be interested in playing games or start a similar application of interest. This kind of “community-centric” application draws a lot attention in the data communication world. This is a typical ad hoc network application, wherein users are mobile and a community of interest is formed on demand using portable devices, and multicast communication can fit in. Multicast can reduce the communication costs, the link bandwidth consumption, sender and

router processing and delivery delay. In addition, it can provide a simple and robust communication mechanism when the receiver's individual addresses are unknown or changeable. Multicast routing protocols for ad hoc networks have been proposed [20] [22] [23] [24] [25] in order to save the network bandwidth and node resource because they are the protocols for powerful communication used in multi-hop applications, and are more efficient than the approach of sending the same information from the source to each of the receivers individually. Most papers about Mobile Ad Hoc Networks (Multicast) are concentrated on proposing a new Multicast Routing Protocol, like multicast routing algorithm with swarm intelligence [27], or trying to optimize existing protocol by introducing strategies, like using support group for route reconstruction which supports local reconfiguration [29], deducing data-overhead of mesh-based ad hoc multicast routing protocols by Steiner tree meshes [2]. Many other papers contributed to the power/energy awareness of ad hoc networks. Other issues like security/security routing are also presented in some papers.

The presence of wireless communication and mobility makes an ad hoc network unlike a traditional wired network and requires that the routing protocols used in an ad hoc network be based on new and different principles. Routing protocols for traditional wired networks are designed to support a tremendous numbers of nodes, but they assume that the relative position of the nodes will generally remain unchanged [33]. In a mobile ad hoc network, however, there may be fewer nodes among which to route, and the network topology changes can be drastic and frequent as the individual mobile nodes move.

1.3 Multicast Routing Protocol Classification in MANET

This thesis will concentrate on multicast routing protocols operating in the network layer. Characteristics of MANETs are frequent changes of topology due to hosts' mobility, limit battery, low bandwidth and unreliable communication; and they impose special constraint when designing routing protocol for MANET.

To design an efficient and reliable routing strategy for the networks, many routing protocols have been designed for this particular network types. There are several criteria to classify these.

One criterion has to do with maintaining routing states and classifies routing mechanisms into three types: proactive routing protocols, reactive routing protocols and hybrid protocols.

Proactive routing protocols [30] attempt to keep the freshest route information from the whole network. In order to maintain fresh route information, these protocols set several tables to store these messages and periodically update throughout the entire network. Periodic protocols can be designed to adjust their periodic interval to try to match the rate of change in the network [11], but this approach will suffer from the overhead associated with the tuning mechanism and the lag between a change in conditions and the selection of a new periodic interval. In the worst case, which consists of bursts of topology change followed by stable periods, adapting the periodic interval could result in the protocol using a long interval during the burst periods and a short interval in the stable periods. This worst case may be fairly common, for example, as when a group of people enter a room for a meeting, are seated for the course of the meeting, and then stand up to leave at the end.

A different approach from proactive routing is the reactive routing protocols, or on-demand protocols [30]. Besides local links, these protocols initiate a flooding route discovery when requiring sending data to a specific destination and do not maintain the route information periodically [20][33]. They usually have two mechanisms, route discovery and route maintenance, to create and maintain a route efficiently to prevent highly overloading the whole network. Unlike proactive routing protocols, these protocols can save the resource (e.g. node's battery and network bandwidth) but not always transmit data immediately. On-demand protocols are based on the premise that if a problem or inconsistent state can be detected before it causes permanent harm; all work to correct a problem or maintain consistent state can be delayed until it is known to be needed.

The last is hybrid routing which incorporates merits of proactive and reactive routing. These protocols are designed to increase scalability by allowing nodes with close proximity to work together. They can be formed by some particular backbone to reduce the route discovery overheads and also by a single point failure. Hybrid routing protocols can exhibit a better performance than proactive and reactive schemes can. However, the memory requirement is greater and the path to destination may be suboptimal.

The second criterion classifies protocols according to the global data structure used to forward multicast packets. Existing protocols are tree-based, mesh-based or hybrid [30]. The protocols classified using this criterion will be discussed in Chapter 2.

1.4 Issues in Multicast in a Mobile Ad Hoc Network

The basic routing problem is to find an ordered series of intermediate nodes that can transport a packet across a network from its source to its destination by forwarding the packet along this series of intermediate nodes. In traditional hop-by-hop solutions to the routing table: for each known destination, the routing table lists the next node to which a packet for that destination should be sent.

The routing table at each node can be thought of as a view into part of a distributed data structure that, when taken together, describes the topology of the network. The goal of the routing protocol is to ensure that the overall data structure contains a consistent and correct view of the actual network topology. If the routing tables at some nodes were to become inconsistent, packets may loop in the network. If the routing tables were to contain incorrect information, packets may be dropped. The problem of maintaining a consistent and correct view becomes harder as there is an increase in the number of nodes whose information must be consistent, and as the rate of change in the actual topology increases.

For ad hoc applications, the majority of them are in areas where rapid deployment and dynamic reconfiguration are necessary while a wireline network is not available. These include military battlefields, emergency search and rescue sites, classrooms, and conventions where participants share information dynamically using their mobile devices. These applications lend themselves well to multicast operation. In addition, within a wireless medium, it is even more crucial to reduce transmission overhead and power consumption, since nodes of a MANET rely on batteries [34]; routing protocol must limit the amount of control information passed between nodes. Multicast can improve the efficiency of the wireless links, when sending multiple copies of messages, by exploiting the inherent broadcast property of the wireless medium when multiple mobile nodes are located within the transmission range of a node.

The challenge in creating a routing protocol for ad hoc networks is to design a single protocol that can adapt to the wide variety of conditions that can be present in any ad hoc network over time. For example, the bandwidth available between two nodes in the network may vary from more than 10 Mbps to 10 Kbps or less. The rate selection procedure can use the channel quality estimate to select an appropriate rate [10] [35]. The highest speeds are achieved when using high-speed network interfaces with little interference, and the extremely low speeds may arise when using low-speed network interfaces or when there is significant interference from outside sources or other nodes' transmitters. Similar to the potential variability in bandwidth, nodes in an ad hoc network may alternate between periods during which they are stationary with respect to each other and periods during which they change topology rapidly. Conditions across a single network may also vary, so while some nodes are slow moving, others change location rapidly.

The routing protocol must perform efficiently in environments in which nodes are stationary and bandwidth is not a limiting factor. Yet, the same protocol must still function efficiently when the bandwidth available between nodes is low and the level of mobility and topology change is high. Because it is often impossible to know a priori what environment the protocol

will find itself in, and because the environment can change unpredictably, the routing protocol must be able to adapt automatically.

Most routing protocols include at least some periodic behaviors [20], meaning that there are protocol operations that are performed regularly at some interval regardless of outside events. These periodic behaviors typically limit the ability of the protocols to adapt to changing environments. If the periodic interval is set too short, the protocol will be inefficient as it performs its activities more often than required to react to changes in the network topology. If the periodic interval is set too long, the protocol will not react sufficiently quickly to changes in the network topology, and packets will be lost [4].

Besides the issues for any ad hoc routing protocol listed above, wireless mobile multicasting faces several other challenges. Multicast group members can move, thus preclude the use of a fixed multicast topology. Transient loops may form during reconfiguration of distributed structure (e.g., tree) as a result of the mobility [36]. Therefore, the reconfiguration scheme should be kept simple to maintain the channel overhead low. As stated, dynamic group membership and constant update of delivery path due to node movement impose challenges for multicasting in Mobile Ad Hoc Network.

Also, the traffic types in ad hoc networks are quite different from those in an infrastructure wireless network [31], including:

1) Peer-to-peer traffic. Communication between two nodes is within one hop. Network traffic is usually consistent. Example of this kind of scenario is two-sided conferencing application.

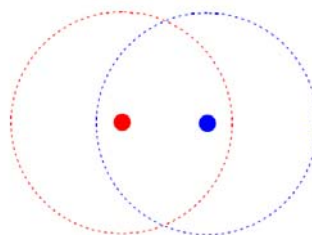


Figure 1.2 Example of peer-to-peer communication in MANETs

2) Remote-to-remote traffic. Communication between two nodes is beyond a single hop but maintains a stable route between them. This may be the result of several nodes staying within communication range of each other in a single area or possibly moving as a group. Such system can be divided in a set of spatially defined and disjoint clusters. Nodes within the same cluster rarely moves with respect to each other, but different clusters always move with respect to other clusters. Examples of this kind of scenario are the Personal Area Networks (PANs): in such particular cases, a single cluster can be identified with a person wearing one or more mobile wireless devices. The different devices attached to a given person seldom move with respect to other devices in the same cluster, but the same person moves in an environment where other people (clusters) are present and move.

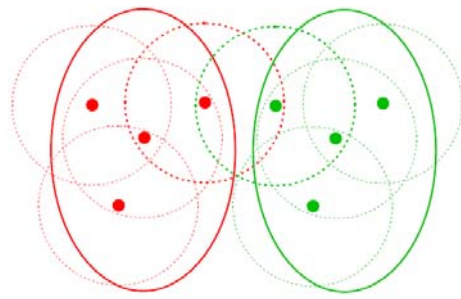


Figure 1.3 Example of remote-to-remote communication in MANETs

3) Dynamic Traffic. This occurs when nodes are moving around. Routes must be reconstructed. This results in a poor connectivity and network activity in short bursts.

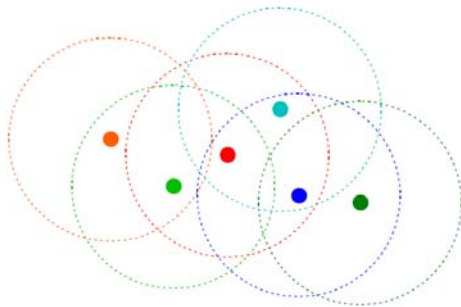


Figure 1.4 Example of mixed communication in MANET

To summarize the differences that exist between wired and wireless multicast communication issues, the selective qualitative comparison of wired and wireless multicast can be described in the following Table 1.1.

Table 1.1 Selective Qualitative Comparison of Wired and Wireless Multicast

Qualitative comparison of wired and wireless multicast			
Issues	Current “wired” multicast	Wireless and mobile multicast	Possible ways to support wireless and mobile multicast
Type of links	Symmetrical and fixed characteristics.	Possibly asymmetrical and/or unidirectional links.	Handle route asymmetry and unidirectional links using possible prediction scheme.
Bandwidth	Plentiful.	Limited and variable amount.	Adaptive membership and routing updates according to bandwidth availability and user mobility.
Topology	Fixed.	Fixed in infrastructure-based, dynamic in ad hoc networks.	Protocols for both fixed and changing topology by “sensing” topological changes.
Routing	Fixed routing structure throughout the multicast session.	Routing structure subject to change due to user mobility.	Dynamically adaptive routing to current structure and available resources.
Quality of Service	Individual routes can use RSVP.	Due to user mobility, RSVP may cause excessive overhead.	Design of new protocols for “soft” QoS under varying link conditions and mobility.

1.5 Common Flaws in Performance Study in MANET

Many existing performance studies in MANET also suffer from three common flaws:

a. Simplistic mobility models: Many existing studies use either a Uniform mobility model [21] or the Random Waypoint model [22] [32]. It is well known that because these models utilize random, independent movements, they do not reflect realistic usage patterns.

b. Low density. Many evaluations use only 50 mobile nodes in a $1000m^2$ field and a 250m radio range, which often lead to a density of less than 10 nodes within radio range [21][32][23]. However, there are many common scenarios in which a network may have many more users in a small area – any situation in which there is a planned gathering or a crowd. This density will likely result in congestion and packet loss, thus some protocols use packet loss as an indicator for mobility they may react in precisely the wrong way.

c. Low traffic load. Current evaluations generally employ a very low data rate of 2 to 20 kbps.

1.6 Contributions

The work in this thesis concentrates on multicast routing in mobile ad hoc network. Two protocols were proposed and evaluated. The QoS-disabled multicast routing protocol was an on-demand routing that aimed at improving the packet delivery ratio and elongating the life time of the network. In addition, the QoS-enabled protocol took QoS requirement as bandwidth constraint into consideration and proposed a modified QoS-based scheme of the QoS-disabled counterpart.

In QoS-disabled scheme proposed, each intermediate node maintains a soft state forwarding probability, called as *Packet_Delivery_Prediction* (PDP), for the next hop. This PDP is calculated using the *mobility prediction* model and *remaining* battery of this very node (Chapter 4). It is continuously updated based on data packets it received or sent and/or the

preset time span. The nodes use this prediction to decide which next hop to use to forward the data packets and which node to use as back-up node to the receiver in case certain data packets are missing. Since initially the network uses modified On-Demand Multicast Routing Protocol (ODMRP) to build the underlying mesh structure in mesh construction stage, every node that participates in it will have one or more routes to the receiver stored in the routing table. During data forwarding stage, the forwarding nodes can select the most appropriate next hop node from its routing table using data forwarding scheme introduced in Chapter 4. In this way, a data forwarding tree is built on the mesh structure. Less control or data packets will be flooded in the network thus it saves the limited bandwidth and battery power that are key constraint in MANET.

In QoS-enabled protocol, every node participating in packet forwarding monitors the bandwidth consumption of itself and neighboring nodes. Real-time traffic flows may be either admitted or rejected according to bandwidth availability of the nodes along the route. The protocol guarantees higher packet delivery ratio and lower average end-to-end delay of the admitted real-time traffic compared to QoS-disabled one.

The organization of this thesis is as follows:

Chapter 2 provides an overview on the mobility models that are always used in the study of routing protocols in MANET. The most basic multicast routing protocols that have been studied in the past are described. Four protocols, including flooding, tree-based scheme, mesh-based scheme and hybrid scheme are presented.

Chapter 3 takes a look at the most popular evaluation metrics that have been used in routing protocol and the methods used to improve these metrics. It gives hint on how to solve the existing performance limitation, but also reveals how each metrics may affect each other under different circumstances.

Chapter 4 describes the proposed multicast protocol that aims at improving the packet delivery ratio and elongating the life time of the network. The operation of the protocol, the message used in it and the data structures are presented in details.

Chapter 5 gives the simulation result using n2-2.28 and compares performance of the proposed scheme in different scenarios with MAODV [32] and ODMRP [20]. It also analyzes the result and summarizes the strength and weakness of the scheme.

Chapter 6 takes QoS bandwidth constraint into consideration and modifies the proposed scheme so that it can meet QoS requirement of the real-time traffic. The modified operation, message and data structures are described.

Chapter 7 simulates and compares the results of the QoS-supported scheme and QoS-disabled scheme.

Chapter 8 concludes the thesis through giving a summary of the achieved work and discusses the directions for future research in this topic.

Chapter 2

Basic Multicast Routing Protocols

In an ad hoc wireless network, nodes may move freely within the field. For a pair of nodes to communicate, a route must be formed between intermediate nodes. For this type of network, it is very important to model nodes' positions and movement, as transmitting range is generally fairly small compared to the size of the field.

In this chapter, the most basic multicast routing protocols that have been proposed in the past year are described. Before proceeding to the routing schemes, some popular mobility models are introduced.

2. 1 Mobility Models

Dynamic topology changes in wireless multi-hop network will cause low network connectivity and/or low network performance. To capture the nature of mobility of nodes in a Mobile Ad Hoc network (MANET), different mobility models have been proposed. The mobility models used in simulations can be roughly divided into two categories: independent entity models and group-based models. In the independent entity models, the movement of each node is modeled independently of any other nodes in the simulation. In the group mobility models, there is some relationship among the nodes and their movements throughout the cells or field.

In order to thoroughly simulate a new protocol for an ad hoc network, it is imperative to use a mobility model that accurately represents the mobile nodes that will eventually utilize the given protocol. Only in this scenario is it possible to determine whether or not the proposed protocol will be useful when being implemented. A mobility model should attempt to mimic the movements of real mobile nodes. Changes in speed and direction must occur in a

reasonable manner. For example, mobile nodes may not travel in straight lines at constant speeds throughout the entire simulation because real mobile nodes would not travel in such a restricted manner. Several mobility models are presented here.

2.1.1 Entity Models

Movements of mobile nodes considered in this class are completely uncorrelated. Each mobile node follows an individual independent mobility scenario. Random Walk, Random Waypoint, Random Direction, Modified Random Direction, Brownian Motion, and Gauss-Markov are examples of this class. Some geographic models, like map-based, city section, Manhattan, also belong to this class.

2.1.1.1 Random Walk Model

The simplest entity mobility model used is known as the Random Walk Model. The Random Walk Mobility model was developed to mimic extremely unpredictable ways. In this model, a mobile node moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from pre-defined ranges, $[\text{speed_min}, \text{speed_max}]$ and $[0, 2\pi]$ respectively. Each movement in Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d , which means the course is followed until either a predefined time or distance is moved or elapsed, and at the end of it new direction and speed are calculated. If a mobile node which moves according to this model reaches a simulation boundary, it “bounces” off the simulation border with an angle determined by the incoming direction. The mobile node then continues along this new path. The Random Walk Mobility Model is a widely used mobility mode (e.g. [37] [38]).

The Random Walk Mobility Model is memory-less mobility pattern because it retains no knowledge concerning its past locations and speed values [39]. The current speed and direction of a mobile node is independent of its past speed and direction [40]. This characteristic can generate unrealistic movements such as sudden stops and sharp turns.

2.1.1.2 Random Waypoint (RWP) Model

The Random Waypoint Mobility Model includes pause times between changes in direction and/or speed. A mobile node begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the mobile node chooses a random destination in the simulation area and a speed that is uniformly distributed between [speed_min, speed_max]. The mobile node then travels toward the newly chosen destination at the selected speed. Upon arrival, the mobile node pauses for a specific time period before starting the process again.

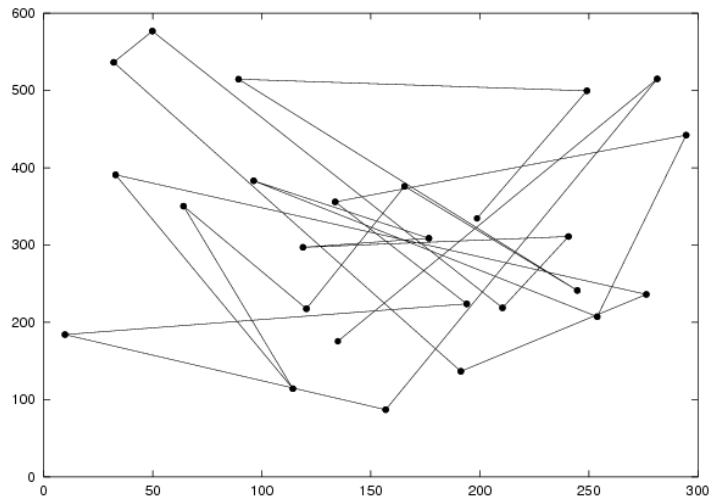


Figure 2.1 Traveling pattern of a mobile node using the Random Waypoint Mobility Model
Figure 2.1 shows an example traveling pattern of a mobile node using Random Waypoint Mobility Model starting at a randomly chosen position (133,180); the speed of the mobile node in the figure is uniformly chosen between 0 and 10m/s. The movement pattern of a mobile node using Random Waypoint Mobility Model is similar to the Random Walk Mobility Model if pause time is zero, as well as minimal speed and maximum speed is the same respectively.

In most of the performance study that use the Random Waypoint Mobility Model, the mobile nodes are initially distributed randomly around the simulation area. This initial random distribution of mobile nodes is not representative of the manner in which nodes distribute

themselves when moving. Study in [40] shows that there is high variability in average mobile node neighbor percentage when using Random Waypoint Model. A neighbor of a mobile node is a node that is within the mobile node's transmission range; the average mobile node neighbor percentage is the cumulative percentage of total mobile nodes that are a given mobile node's neighbor. The nodes will have non-uniform distribution after some run, with maximum node density in the center region and minimum node density on the boundary. This high variability in average mobile node neighbor percentage will produce high variability in performance result, unless the simulation results are calculated from all simulation runs. The nodes will experience sharp turn, sudden acceleration and sudden stop when using this mobility model.

There is also a complex relationship between node speed and pause time in the Random Waypoint Mobility Model. For example, a scenario with fast mobile nodes and long pause time actually produces a more stable network than a scenario with slower mobile nodes and shorter pause times [41].

2.1.1.3 Random Direction (RD) Model

The Random Direction Mobility Model [42] was created to overcome density waves in the average number of neighbors produced by Random Waypoint Mobility Model. A density wave is the clustering of nodes in one part of the simulation area. In the Random Waypoint Mobility Model, the probability of a mobile node choosing a new destination that is located in the center of the simulation area, or a destination which requires traveling through the middle of the simulation area, is high. Thus the mobile nodes appear to converge, disperse, and converge again.

In order to alleviate this type of behavior and promote a semi-constant number of neighbors throughout the simulation, the Random Direction Model was developed [42]. In this model, mobile nodes choose a random direction in which to travel similar to the Random Walk Mobility Model. A mobile node then travels to the border of the simulation area in that

direction. Once the simulation boundary is reached, the mobile node pauses for a specific time T_{pause} , chooses another angular direction and continues the process.

This model maintains uniform node distribution; however, it also has unrealistic moving behaviors, like sharp turn and sudden stop.

2.1.2 Group Model

In an ad hoc network, there are many situations where it is necessary to model the behavior of mobile nodes as they move together. For example, a group of soldiers in a military scenario may be assigned the task of searching a particular plot of land in order to destroy land mines, or simply work together in a cooperative manner to accomplish a common goal. In order to model such situations, a group mobility model is needed to simulate this cooperative characteristic.

2.1.2.1 Reference Point Group (RPGM) Model

The Reference Point Group Mobility (RPGM) model represents the random motion of a group of mobile nodes as well as the random motion of each individual mobile node within the group [40]. Group movements are based upon the path traveled by a logical center for the group. The logical center for the group is used to calculate group motion via a group motion vector, say, \overline{GM} . The motion of the group center completely characterizes the movement of its corresponding group of mobile nodes, including their direction and speed. Individual mobile nodes randomly move about their own pre-defined reference points, whose movements depend on the group movement. As the individual reference points move from time t to $t+1$, their locations are updated according to the group's logical center. Once the updated reference points, $RP(t+1)$, are calculated, they are combined with a random motion vector, \overline{RM} , to represent the random motion of each mobile node about its individual reference point.

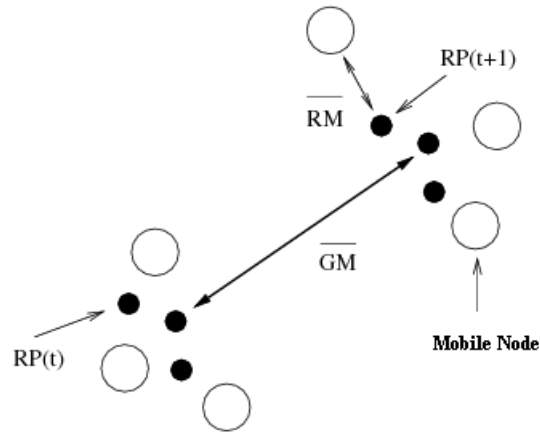


Figure 2.2 Movements of three mobile nodes using RPGM

Figure 2.2 gives an illustration of three mobile nodes moving with RPGM model. The figure illustrates that, at time t , three black dots exist to represent the reference points, $RP(t)$, for the three mobile nodes. As shown in the figure, the RPGM model uses a group motion vector

\overline{GM} to calculate each mobile node's new reference point at time $t+1$, $RP(t+1)$; as stated, \overline{GM} may be randomly chosen or predefined. The new position for each mobile node is then

calculated by summing a random motion vector, \overline{RM} , with the new reference point. The

length of \overline{RM} is uniformly distributed within a specific radius centered at $RP(t+1)$ and its direction is uniformly distributed between 0 and 2π .

In this model, the reference point is predefined during the simulation. The group leader determines the group's motion behavior. Each node is assigned with a Reference Point that follows the mobility of the group leader.

If the group leader uses the mobility pattern of RWP model, RPGM model will inherit all flaws of RWP model. It is difficult to generate different levels of spatial dependency between group members and their leader in the RPGM model.

2.1.3 Mobility Model Summary

The performance of an ad hoc network protocol can vary significantly with different mobility models. Even using the same mobility model, different model parameters can also affect the performance of the ad hoc network protocol.

The Random Walk Mobility Model with a small input parameter (distance or time) produces Brownian motion. A large input parameter (distance or time) is similar to the Random Waypoint Model (RWP) without pause times. The main difference between these two mobility models is that mobile nodes are more likely to cluster in the center of the simulation area with Random Waypoint Mobility Model (RWP) [40].

The Random Direction Mobility Model is an unrealistic model because it is unlikely that people would spread themselves throughout an area. In addition, it is unlikely that people will only pause at the edge of a given area [40].

The Reference Point Group Mobility Model (RPGM) is a generic method for handling group mobility. An entity mobility model needs to be specified to handle both the movement of a group of mobile nodes and the movement of the individual mobile nodes within the group.

2.2 Basic Routing Protocols

To assist group-oriented communication in MANET, various multicast protocols have been proposed to perform multicast in ad hoc networks. If we use global data structure to forward multicast packets, there are three major categories of multicast routing protocols: the tree-based protocols, like Multicast Ad-hoc On-Demand Distance Vector (MAODV) [32] [43], the mesh-based protocols, like On-Demand Multicast Routing Protocol (ODMRP) [20] [44], and hybrid protocol, like Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR) [45]. As in fixed multicast routing, tree-based protocols build a tree over which multicast data is forwarded. To achieve robustness in the presence of universal mobility and frequent

node outage, mesh-based protocols build a mesh for forwarding multicast data with path redundancy inherent to mesh.

Both tree-based protocol and mesh-based protocol have their pros and cons. The tree-based approaches is bandwidth-efficient, and provide high data forwarding efficiency at the expense of low robustness because in a rapidly reconfiguring environment, the high frequency of branch repairs will increase the risk of packet loss as well as introduce high channel and processor overhead; The mesh-based approaches, on the other hand, may have multiple paths between any source and receiver pair, which allow multicast datagrams to be delivered to the receivers even if links fail, however, the forwarding overhead and network load may increase due to the periodically limited flooding. Thus, hybrid multicast approaches, like Multicast Core Extraction Distributed Ad Hoc Routing (MCEDAR), are proposed to achieved the advantage of both tree and meshed-based approaches.

2.2.1 Flooding

One straight forward way to provide multicast in a MANET is through flooding. When a node receives a packet, it broadcasts the packet except if it has seen that packet before. Nodes keep a cache of recently received packets; older packets are replaced by newly received ones. A node only rebroadcasts a packet if that packet is not in the node's cache [30]. With this approach, data packets are sent throughout the MANET, and every node that receives this packet broadcasts it to all its immediate neighbor nodes exactly once. It is suggested that in a highly mobile ad hoc network, flooding of the whole network may be a viable alternative for reliable multicast. However, this approach has considerable overhead since a number of duplicated packets are sent and packet collisions do occur in a multiple-access-based MANET.

2.2.2 Tree-Based Protocol: Multicast Ad-hoc On-Demand Distance Vector (MAODV)

Multicast Ad-hoc On-Demand Distance Vector (MAODV) routing protocol is intended for use by mobile nodes in an ad hoc network. It enables dynamic, self-starting, multihop routing

between participating mobile nodes wishing to join or participate in a multicast group within an ad hoc network. The membership of the multicast group is free to change during the lifetime of the network. MAODV enables mobile nodes to establish a tree connecting multicast group members [43].

MAODV uses sequence number to identify multicast group. Each multicast group has its own sequence number, which is initialized by the multicast group leader and incremented periodically. Using these sequence numbers ensures that the routes to multicast groups are always the most current ones available. This means that if a node that requests routes to the multicast group has two routes, it always selects the one with greatest sequence number.

Each multicast group is organized by using a tree structure, composed of the group members and several routers, which are not group members but have to exist in the tree to connect the group members. The group members and routers are all tree members and belong to the group tree. The group member that first constructs the tree is the group leader for that tree and is responsible for maintaining the group tree by periodically broadcasting Group-Hello (GRPH) message in the whole network. The group leader also maintains the group sequence number, which is propagated through the GRPH. Here, GRPH contains extensions that indicate the multicast group IP address and sequence numbers (incremented every Group Hello) of all multicast groups for which the node is the group leader. Each node may maintain three tables, Unicast Route Table, Multicast Route Table and Group Leader Table. Unicast Route Table is used to record the next hop for routes to other destination for unicast communication (destination). Multicast Route Table is used to list the next hops for the tree structure for each multicast group. Every node that belongs to that group tree should maintain such entries, with its own identity as group leader, group member, or router. Every next hop is associated with direction either downstream or upstream. Group Leader Table is used to record the currently-known multicast group address with its group leader address and next hop towards that group leader when the node receives a periodic GRPH message.

MAODV follows directly from unicast protocol, Ad-hoc On-demand Distance Vector (AODV). It discovers multicast routes on demand using a broadcast route discovery mechanism employing route request (RREQ) and route reply (RREP) messages that exist in the unicast AODV protocol. As long as the multicast group members remain connected within a ‘multicast tree’, MAODV does not play any role. When a source node wants to join or participate in a multicast group and it is not a member of that group yet, simplified operations are described as follows:

a. Generating Route Request (RREQ) [43]

A node sends a RREQ either when it determines that it should be part of a multicast group, and it is not already a member of that group, or when it has message to send to the multicast group but does not have a route to that group.

If the node wishes to join the multicast group, it sets the “J” flag in the RREQ (RREQ-J); otherwise, it leaves the flag unset. The destination address of RREQ is always set to the multicast group address. If the node knows the group leader and has a route to it, the node may place the group leader’s address in the multicast group leader extension, and unicast the RREQ to the corresponding next hop to the destination; Otherwise, if the node does not have a route to the group leader, or if it does not know who the multicast group leader is, it broadcasts the RREQ and does not include the multicast group leader extension field.

After transmitting RREQ, the node waits for the reception of a route reply (RREP). The node may resend RREQ up to RREQ_RETRIES additional times if a RREP is not received. If a RREP is not received after RREQ_RETRIES additional requests, the node may assume that there are no other members of that particular group within the connected portion of the network. If it wanted to join the multicast group, it then becomes the multicast group leader for that multicast group. Otherwise, if it only wants to send packets to that group without actually joining the group, it drops the packets it had for that group and aborts the session.

b. Receiving Route Request (RREQ)

When a node receives a RREQ, the node checks whether the 'J' flag of the RREQ is set. If the 'J' flag is set, the node can only respond if it is a member of the multicast tree for the indicated multicast group, and if its record of the multicast group sequence number is at least as great as that contained in the RREQ. If the 'J' flag is not set, the node can respond if it has an unexpired route to the multicast group and the multicast group sequence number is at least as great as the one stated in RREQ.

If the node does not meet either of these conditions, it replaces the IP address in the IP head of the original RREQ, updates the Destination Sequence Number to the maximum existing Destination Sequence Number in the RREQ and rebroadcasts this new RREQ [43]. This node creates and updates the route table entry for the source node. It also creates a next hop entry for the multicast group in its multicast route table.

c. Generating Route Reply (RREP)

If a node receives a join RREQ for a multicast group, and it is already a member of the multicast tree for that group, the node updates its multicast route table and generate RREP message. The Source and Destination IP Addresses in RREP contains the current sequence number for the multicast group and the IP address of the group leader. It unicasts the RREP back to the node indicated by the Source IP Address field of the received RREQ.

A node can respond to a join RREQ only if it is a member of the multicast tree. If the node receives a multicast route request that is not a join message, it can reply if it has a current route to the multicast tree. Otherwise it continues forwarding the request.

d. Forwarding Route Reply (RREP)

If an intermediate node receives a RREP in response to a RREQ that it has transmitted, it creates a multicast group next hop entry for the node from which it received the RREP. The direction of this next hop is UPSTREAM, and the Activated Flag is left unset. When the

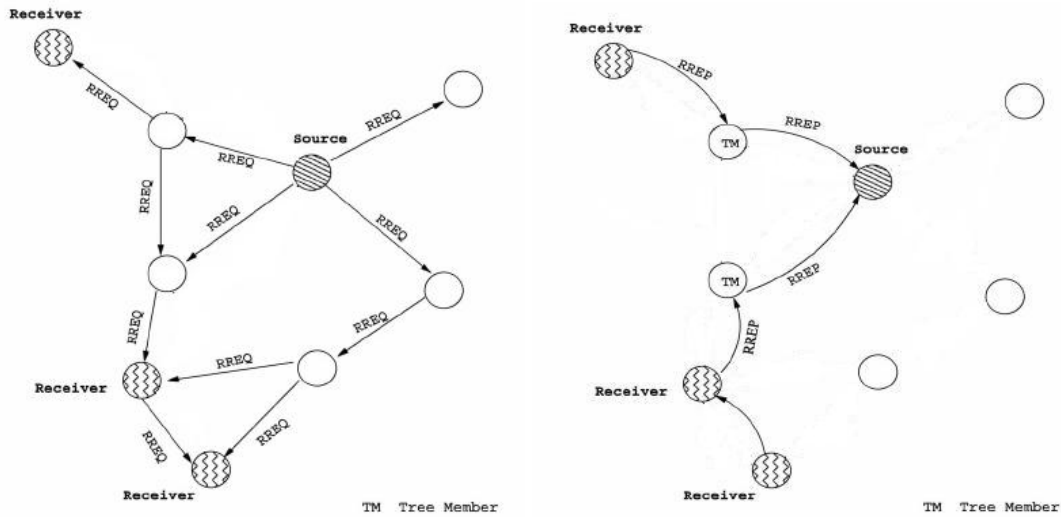
node receives more than one RREP for the same RREQ, it saves the route information with the greatest sequence number, and beyond that the lowest hop count. It discards all other RREPs.

e. Route Activation

When a node broadcasts a RREQ message, it is likely to receive more than one reply since any node in the multicast tree can respond. The RREP message sets up route pointers as it travels back to the source node. If the request is a join request, these route pointers may eventually build a branch onto the multicast tree. Due to the broadcast nature of wireless network, the route to the multicast tree must be explicitly selected and only one of the routes created by the RREP messages. The RREP containing the largest destination sequence number is chosen to be the branch added to the multicast tree, or the path to the multicast tree if the request was a non-join request. If the node received more than one RREP with the same largest sequence number, it selects the first one with the smallest hop count, i.e., the shortest distance to a member of the multicast tree [43]. By sending Multicast Activation (MACT) message, the node selects the route it wishes to use as its link to the multicast tree. The node unicasts this MACT message to the selected next hop, effectively activating the route. It then sets the Activated flag in the next hop Multicast Route Table entry associated with that node. After receiving this message, the node to which the MACT was sent activates the route entry for the link in its multicast route table, thereby finalizing the creation of the tree branch.

When the node receives a MACT selecting it as the next hop, it unicasts its own MACT to the node it has chosen as its next hop, and so on up the tree, until a node which already a part of the multicast tree is reached [43] [32].

MAODV uses destination sequence number for each multicast entry requiring a lot of control message.



(a) Route Request

(b) Route Reply

Figure 2.3 Tree Creation in MAODV

Figure 2.3 (a) shows multicast source requires a route to multicast group by sending join request (RREQ) packets. It broadcasts RREQ packet with join flag set and the destination address set to the multicast group address. Each node, on receiving RREQ (request packet), will update its route table and record the sequence number and next hop information for the source node. Figure 2.3 (b) shows how receivers and tree members send route reply (RREP) packets to source. It follows the criteria described through 2.2.2 c to 2.2.2 d, and is rewritten as:

- A member of the multicast tree with a current route to the destination responds to RREQ with a RREP packet.
- Nonmember will rebroadcast the RREP packet.

2.2.3 Mesh-Based Protocol: On-Demand Multicast Routing Protocol (ODMRP)

On-Demand Multicast Routing Protocol (ODMRP) is a multicast routing protocol designed for ad hoc networks with mobile hosts. It is a mesh-based multicast scheme uses a forwarding group concept, which is only a subset of the nodes forwards the multicast packets via scoped flooding. The mesh structure provides rich connectivity, robustness, and

supplying path redundancy [20]. One of its unique properties is its unicast capability, where ODMRP can efficiently operate as unicast routing protocol and it can also coexist with any unicast routing protocol.

In ODMRP, group membership and multicast routes are established and updated by the source on demand, requiring periodic join/query messages only when sources have data packet to send.

a. Mesh Initialization Phase

a.1 Request Phase

When a multicast source has packets to send but no route and group membership is known, it creates the mesh as follows: each source in the multicast group floods a join query (Query) control packet with data payload piggybacked periodically. This packet is broadcasted to the entire network to refresh membership information and update the routes. When a node receives the Query packet, it stores the source address and unique identifier of the packet to its “Message Cache” to detect duplications. The upstream node address is inserted or updated as the next node for the source node in its “Routing Table”. If the Query packet is not a duplicate and the Time-To-Live value is greater than zero, appropriate fields are updated and it is re-broadcasted [44]. The process continues until reaching the destination (multicast receiver), which broadcasts a join reply (Reply).

a.2 Reply Phase

When a Query packet reaches the multicast receiver, it creates and broadcasts a join reply (Reply) to its neighbors. When a node receives a Reply, it checks if the next node address of one of the entries matches its own address. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group; it sets the FG_FLAG (Forwarding Group Flag). It then broadcasts its own join reply (Reply) built upon matched entries. The next node address field is filled in by extracting the information from its routing table. This way, the Reply is propagated by each forward group member until it reaches the multicast source via

sender node and next node fields in each entry, to establish and update group membership and routes. A node receiving a Reply checks if the next node ID in one of the table's entries matches its own ID, then it considers itself as a forwarding group (FG) node. The reply forwarding process continues until reaching the source through shortest path building a mesh of FG nodes.

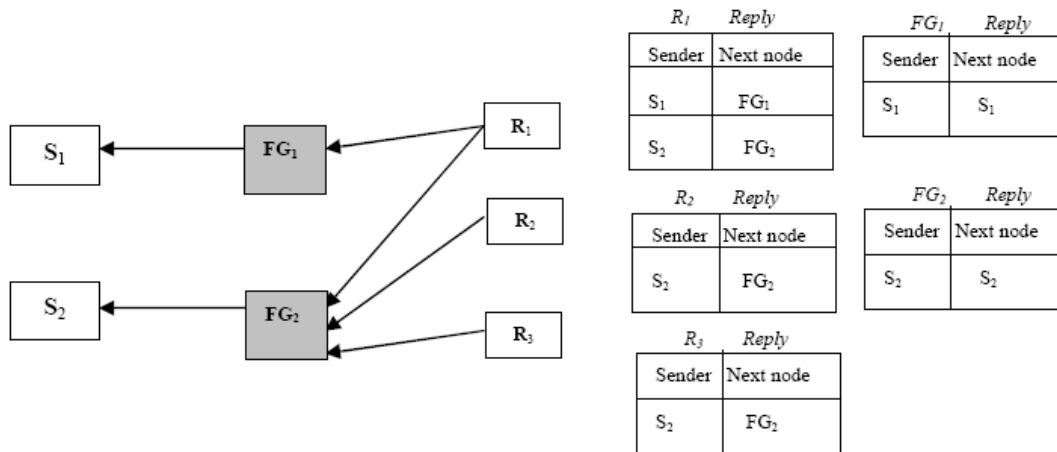


Figure 2.5 Reply Forwarding

b. Mesh maintenance phase

The multicast mesh protects the session from being affected by nodes' mobility through the paths redundancy, as illustrated in Figure 2.6.

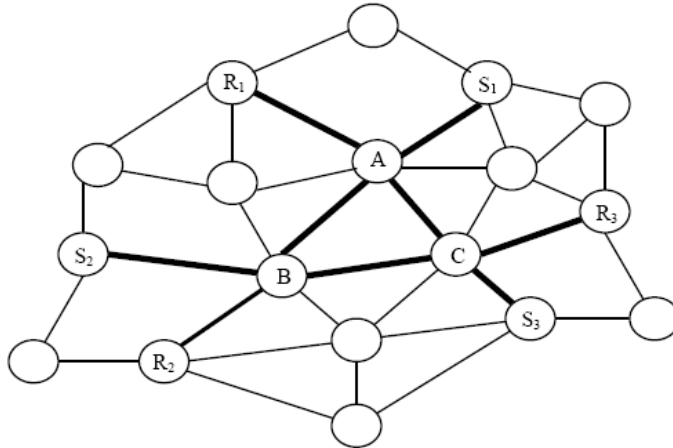


Figure 2.6 Meshing Configuration

For example, during data transmission between S1 and R3, if node B moves, the receiver can still receive data through another path via node C. ODMRP uses a soft state approach to maintain the mesh through employing a mesh refreshment mechanism in which the source periodically floods the Query control packet.

One of ODMRP key advantages is its unicast capability, where a network equipped with ODMRP does not require a separate unicast protocol. Also, the soft state mesh maintenance approach provides robustness but at the expense of high control overhead. Another disadvantage is that the same data packet propagates through more than one path to a destination node, resulting in an increased number of data packets transmissions, thereby reducing the multicast efficiency.

2.2.4 Hybrid Protocol: Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR)

MCEDAR [45] uses a mesh as the underlying mesh infrastructure. Hence every link breakage in the underlying graph does not necessitate a reconfiguration of the infrastructure. At the same time, it uses a forwarding mechanism on the mesh that creates an implicit source based forwarding tree. This ensures that although MCEDAR maintains a mesh routing

infrastructure, the data forwarding occurs only on a source rooted minimum height tree. It decouples the control and data forwarding infrastructure.

2.3 Conclusion

In this chapter, several mobility models and the operation of basic multicast routing protocols in MANET are presented. The drawbacks for the models and the routing schemes are also briefly discussed. The tree-based protocols are bandwidth-efficient, but are not as reliable as mesh-based ones when the topology of the network changes too frequently; the mesh-based protocols are reliable, but the forwarding overhead would be high due to the path redundancy, and are thus not very scalable.

Chapter 3

Evaluation Metrics and Related Protocols

Due to the nature of MANET, under different mobility speed, number of senders, number of receivers, multicast group size and network traffic load scenarios, some of the protocols/schemes will be superior to the others. Useful evaluation metrics for multicast routing protocol include packet delivery ratio, data overhead (number of data packets transmitted per data packet delivered), control overhead (number of control bytes transmitted per data bytes delivered), number of control and data packets transmitted per data packet delivered, delay, etc.

3.1 Data Overhead

Data overhead has a strong impact on the overall scalability and network capacity which can be achieved with a particular routing protocol. Data overhead is related to the cost associated to the use of non-optimal multicast structure. Data overhead reduction is mainly achieved by carefully selecting forwarding nodes in the networks and constructing the data paths [46] [1] [2] [3]. Tree-based as well as mesh-based protocols can use the corresponding methods to reduce forwarding nodes/group. Hybrid protocols can also use the methods to construct the routes.

3.1.1 Data Overhead Reduction for Tree-Based Protocol

The basic concept behind the data overhead reduction for tree-based algorithms is to build some approximation Steiner tree, thus minimal cost tree can be used to forward multicast packets from sender to receivers. The original Steiner tree problem formulation generally does not give an optimal solution in the broadcast medium in multihop wireless networks. It [1] showed that the best approach to reduce the data overhead is reducing the number of forwarding nodes, while increasing the number of leaf nodes.

A proposed distributed approximation algorithm for this approximate Steiner tree works as follows [1]. The source or the root of the sub-tree that the source is in (called source-root) will start flooding a route request message (RREQ). Intermediate nodes, when propagating that message, will increase the hop count. When the RREQ is received by the root of a sub-tree, it sends a route reply (RREP) back through the path that reported the lowest hop count. The nodes in that path are selected as multicast forwarders (MF). In addition, a root of a sub-tree, when propagating the RREQ, will reset the hop count field. This is what makes the process very similar to the computation of the minimal Steiner tree (MST) [49] on the metric closure. Each root of the sub-trees will add to the Steiner tree the path from itself to the source-root, or the nearest root of a sub-tree. The way in which the algorithm is executed from the source-root to the other nodes guarantees that the obtained tree is connected.

The second part of this algorithm is the creation of the cost-effective sub-trees. This part is done locally with just a few messages. Receivers flood a Subtree_Join (ST_JOIN) message only to its one-hop neighbors indicating the multicast group to join. These neighbors answer with a Subtree_Join_Ack (ST_ACK) indicating the number of receivers that covers. This information is known locally by just counting the number of (ST_JOIN) messages received. Finally, receivers send again Subtree_Join_Activation (ST_JOIN_CT) message including their selected root, which is the neighbor who covers a higher number of receivers. This is also known locally from the information in the (ST_ACK). Those nodes that are selected by any receiver, repeat the process acting as receivers. Nodes that already selected a root do not answer this time to ST_JOIN messages.

As the simulation showed [1], distributed version of the algorithm is not as efficient as the centralized one, but offers a good approximation to the centralized scheme. This is because instead of really computing the metric closure in the graph, approximation is used. However, the performance of the distributed approach is still better than the one offered by the Steiner tree.

Simulation [1] also shows that when the number of receivers is low, the proposed schemes do not offer significant differences compared to the Steiner tree heuristic because nodes tend to be very sparse and it is less likely that it is possible to build cost-effective trees. However, when the number of receivers increase, those schemes will build cost-effective trees thus achieve significant reductions in the number of transmission required. The higher the network density, the more reduction in the number of transmissions will occur. This is because for higher densities, it is more likely that several receivers can be close to the same node, which facilitates the creation of cost-effective sub-tree. Also in dense network, the difference between centralized and distributed approaches becomes small, which is because in dense networks, the number of hops between any pair of nodes is also reduced, making the difference between metric closure and its approximation in number of hops be reduced as well.

Of course, using this algorithm to build the minimal cost tree will introduce higher mean path length. This is due to the fact that grouping paths for several receivers makes them not to use their shortest paths.

3.1.2 Data Overhead Reduction for Mesh-Based Protocol

In existing mesh-based protocols, the multicast mesh can consist of shortest path tree plus some backup links [44]. The main advantage of this is that each destination receives multicast data through its best route, which always not only means that the latency from source to each destination is reduced, but also means that the overall number of forwarding nodes will increase, incurring high data overhead. To reduce the number of forwarding nodes, the scheme introduced in [2] builds a forwarding mesh upon Steiner tree, that is, a distributed heuristic based on the epidemic propagation of the number of forwarding nodes. Minimal cost multicast tree is not always the concern; in fact, it may not work well with mobility because of lack of backup mesh. The concern in this scheme is that those minimal cost trees can be used as basis to build a multicast mesh. A forwarding mesh built upon

Steiner trees is expected to have lower data overhead than a forwarding mesh built upon shortest path tree.

This scheme [2] approximates minimal Steiner tree by trying to minimize the number of forwarding nodes required to connect each source and all the receivers. It *adaptively* constructs the multicast forwarding mesh which reduces the number of forwarding nodes when there is enough reliability in the existing mesh. Each ad hoc node will propagate during the path creation, processing information about the number of non-forwarding nodes in its path to the source. Each time a source that wants to discover or refreshes a node initializes the counter to zero. Whenever an intermediate node propagates such a control message, it modifies the counter as follows [2]:

- If node is not forwarding node then increment the counter by one.
- If node is forwarding node then do not increase the counter.
- If node is a receiver then set the counter to one if it is not forwarder, or to zero otherwise.

Thus, a control message with a lower counter value is associated to a route which produces a lower number of forwarding nodes. By selecting those routes, the resulting multicast tree becomes an approximation of a Steiner tree.

When building the multicast mesh, for a larger number of sources (i.e. number of Steiner trees), the redundancy of the mesh is also bigger, although the increasing rate is not clear. So when the number of sources is big, almost no additional redundancy is required and when the number of sources is small, a larger number of additional links might be required to cope with mobility. The *adaptive* control scheme, a probability path selection algorithm, is used at each node: A node will select the shortest path with probability p and the path that minimizes the number of forwarding nodes with probability $1-p$. The design key is to find the proper value of p so that an appropriate amount of redundancy is added. That is, for a lower number of sources, it reduces redundancy moderately as $|S|$ increases, whereas reduces redundancy

very fast for a greater number of sources. To achieve that behavior, [2] selected the value of p as $p = 1 / (1 + |S|^2)$.

The scheme in [2] is been integrated with ODMRP [44], and reduction of forwarding nodes is compared to original ODMRP. Simulation [2] shows that, although the shortest path tree may change quickly, the least cost tree does not change so frequently. Thus the number of forwarding nodes does not increase that much. By reducing data overhead, the proposed approach manages to offer similar performance as ODMRP at a lower cost in term of forwarding efficiency. This lower cost allows the proposed approach to support higher overall traffic load.

3.1.3 Data Overhead Reduction for Hybrid Protocol

Efficient Hybrid Multicast Routing Protocol [3] (EHMRP) is a modified version of ODMRP aimed at scalability issue of ODMRP. In this algorithm, data packets forwarding path is separated from join-query forwarding when multicast routes to destination nodes are unavailable. The rest of the protocol works the same way as ODMRP [44]. Control overhead is reduced for large network size at high network load. Data packet forwarded using Differential Destination Multicast (DDM) [47] and low overhead local clustering from MCEDAR [45], a hybrid protocol, makes this protocol a hybrid one.

In this protocol, the nodes are classified into *core* and *normal* nodes. *Core* nodes are ideally the minimum domination set of the network topology [45]. Each node runs periodically the algorithm to calculate effective degree to help decide whether it is *core* node or *normal* node. Effective degree of a node is the number of neighbors who have chosen the current node as their dominating node. Each node selects the highest degree node with maximum effective degree and lowest node id among one hop neighbors as its dominating node. Each normal node forwards data packets to its dominating node. If a node dominates one or more nodes, it is termed as *core* node. Each normal node selects another core node as its dominator. Each *core* node stores the nodes it dominates, path of its nearby core nodes and dominated nodes

of each core node in the third neighborhood. It also stores certain information for each node it dominates, all one hop neighbors and the dominator of each neighbor node. No core node is aware of the complete core graph. The complete algorithm is based on local computation and hence it is efficient.

When multicast routes to destination nodes are unavailable, *join-query* messages are sent to all nodes in the network and data packets are forwarded by the core nodes to the destination nodes using Differential Destination Multicast (DDM) [47]. DDM is a stateless multicast approach where multicast tree information is appended with each data packet header. The key components of this protocol are as follows:

- (a) Classifying core and normal nodes;
- (b) Separating out data forwarding path using DDM while sending join query request;
- (c) Separating handling of received data packets coming through DDM path;
- (d) Group membership update;
- (e) Normal functionality of ODMRP protocol [44].

Data forwarding path is separated from join query sending path. Data packet is duplicated, and join query is sent with IP packet headers but the data portion of the packet is stripped from the join query request. Data portion of the packet is sent to the members of that multicast group using DDM. With DDM, source node encodes all members of the destination multicast group and attaches them with the data packet by means of a new IP option field defined by this protocol [47]. This packet is sent to the next hop receiver nodes in the same multicast group using single hop broadcast.

Since EHMRP uses DDM for data forwarding when sending join query packets, while ODMRP uses mesh forwarding when sending join query packets, the data-overhead is lower by EHMRP than ODMRP [3].

3.2 Control Overhead

Control overhead is defined as the ratio between the number of control bytes transmitted to the number of data bytes received. In ODMRP (On-Demand Multicast Routing Protocol) [44], control bytes account for Join-Query and Join-Table packets. In MAODV (Multicast Ad Hoc On-Demand Distance Vector) [43], control bytes account for the RREQ, RREP, MACT, Hello and GRP-Hello packets. In flooding, it generates no control traffic but involves redundant retransmissions, thus increases data forwarding overhead.

To reduce control overhead, the main approach can be either constructing the multicast mesh or tree using only selected/limited nodes and decreasing the retransmission of certain control or data packets; or extending the route refreshment interval so that there won't be too many control packets flooded in the network too often [4].

3.2.1 Preemptive Multicast Routing (PMR)

The main idea behind Preemptive Multicast Routing (PMR) [4] is to extend the route refreshment interval. It can be used to detect potential route breaks and, after detection, initiate a new route discovery. That is, the route discovery is initiated only when necessary. Preemptive route maintenance allows a routing algorithm maintains connectivity by preemptively switching to a path of higher quality when quality of the currently used path is deemed suspicious.

When applied to ODMRP [44], preemptive multicast routing can result in better performance because it significantly reduces the number of route refreshment packets by lowering the frequency of flooding. In addition, preemptive route maintenance also reduces the number of collisions of data packets in the original ODMRP.

The preemptive routing algorithm works in two stages:

- (1) Detecting that a route is likely to be disconnected soon;
- (2) Searching for a new path and switching to it.

Detection of potential link breaks is a critical design issue of a preemptive routing protocol. For this PMR, since most route breaks are caused by link failure due to node mobility, it uses signal strength as the measure of path quality [50]. The quality of path can be evaluated based on other factors such as hop count, the age of path, and traffic load. The algorithm in [48] used the signal strength of received packets as the primary criterion with the hop count being the second measure.

Two types of refreshment intervals are defined in PMR: *ExtendedRefreshInterval* and *MinRefreshInterval* [4]. When nodes move slowly, the frequency of flooding the network with JOIN-QUERY packets is reduced and bound by $1/ExtendedRefreshInterval$; when nodes start moving more quickly, the frequency of flooding increases accordingly due to warning messages, and is bound by $1/MinRefreshInterval$, the frequency of flooding in ODMRP.

Two design issues of applying preemptive route maintenance to multicast are identified [4]:

- (1) Feedback implosion;
- (2) High frequency of warning messages at sender.

The most serious problem is feedback implosion, which is best explained using a logical tree structure (although the underlying routing structure can be a mesh, as in the case of ODMRP).

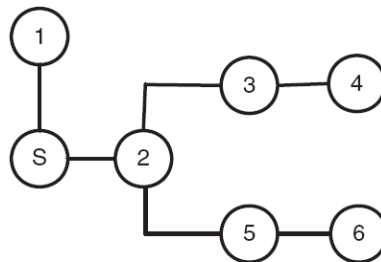


Figure 3.1 A logical tree in the routing mesh

In Figure 3.1, if both links (3,4) and (5, 6) are about to break around the same time, node 4 and 6 would each send a warning message, at probably the same time, or shortly after another, while only one warning from either node suffices to trigger the source “s” to initiate a route refreshment. In this case, the redundancy of warning message will introduce high traffic overhead; it is feedback implosion.

Feedback implosion has two implications when preemptive route maintenance is applied to multicast. The first implication is high traffic overhead caused by redundant warning messages. It may happen that two nodes with no relationship each send a warning message to the source, and the two messages arrive at the source one very shortly after another, as illustrated in Figure 3.1; or several warning messages triggered by the same broken link would cause a sender to flood the network with several JOIN QUERY packets, consuming resources unnecessarily and causing collisions, congestion and delay due to contention, which is the second implication if nothing is done. Feedback suppression mechanism [4] is introduced to overcome this. Different data bits, like *WarningGenerated*, *SentWarning*, *SentJoinQuery*, are used to help suppress the number of warning messages that will be responded by source nodes. Warning messages are unicast to sources using existing routing table constructed by ODMRP.

Simulation in [4] showed that PMR can accommodate multicast applications with both low and high mobility, offering both low control overhead and high delivery ratio.

3.2.2 Neighbor Aware Multicast Routing Protocol (NAMP)

Neighbor aware multicast routing protocol (NAMP) [6] is a tree-based multicast routing protocol which also includes the neighboring concept. The routes are built and maintained using the traditional request and reply messages. A hard state approach is used for multicast group maintenance.

It uses neighbor information of two-hops away for transmitting the packets to the receiver(s). If the receiver(s) is not within this range, it searches the receiver(s) using dominant pruning flooding method and forms the multicast group using the replies along the reverse path.

Each node keeps the information of all of its neighbors of one-hop distance. This information is kept in a neighbor table in each node. A node periodically transmits HELLO packet (containing its own neighbor table information) to all of its neighbors, and gets HELLO or HELLO_REP from its neighbor depending on whether this neighbor is previously included in the neighbor table of both nodes.

When a source wants to send a data packet, it initializes a FLOOD-REQ packet with data payload attached. This packet is flooded throughout the network based on dominant pruning [6]. Dominant pruning approach extends the range of neighborhood information into nodes that are two-hop apart. This two-hop neighborhood knowledge can be obtained by exchanging the adjacent node lists with neighbors. In dominant pruning, the sender node selects adjacent nodes that should relay the packet to complete broadcast. The IDs of selected adjacent nodes are recorded in the packet as a forwarding list. An adjacent node that is requested to relay the packet again determines the forwarding list. This process is iterated until broadcast is completed. The forwarding list should be minimized to decrease the number of transmissions.

When the destination node gets the FLOOD_REQ packet, it sends a REPLY packet to the source backtracking the path through which it has received the FLOOD_REQ packet. It can do so because each intermediate node between source and destination has updated the source address by its own address, and cached the source/upstream address locally. Thus FLOOD_REQ contains only one address as source address and it changes at each hop.

NAMP [6] applies the secondary forwarder list scheme to maintain the multicast tree. In dominant pruning approach, only those nodes are selected for forwarding the FLOOD-REQ

packet, thus can be used to flood the packet throughout the whole network with minimum effort. The nodes that could be selected for flooding the FLOOD-REQ packet but were not selected by dominant pruning approach form the secondary forwarder list (sfl), which can be used when the primary forwarding nodes drift off from their current position or have link failure. Each node forming the multicast tree keeps its own secondary forwarder list (sfl) and also sends the sfl to the selected forwarder. Also, every forwarding node as well as the source can detect any link failure if it finds no information of the downstream node in its neighbor table.

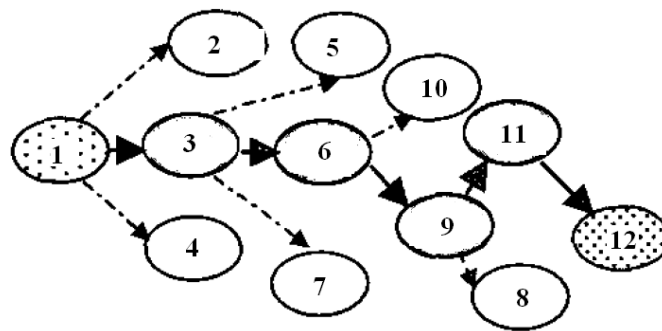


Figure 3.2 Formation of Secondary Forwarder List (SFL)

Figure 3.2 shows how the secondary forwarder list is created. In this figure, node 1 is the source node and node 12 is the destination. The selected forwarders in this route are {3, 6, 9, 11}. Each of these nodes and the source keep the secondary forwarder list (sfl). The sfl of node 1 contains 2 and 4; sfl of node 3 contains 5 and 7; sfl of node 6 contains 10; sfl of node 9 contains 8 and node 11 does not have any secondary forwarder node in its sfl.

NAMP expects that in most cases, the neighbors within the two-hop distance will know how to reach the destination(s). Otherwise, a node from the secondary forwarder list (sfl) is selected as forwarder. The newly selected forwarder will try to communicate with the known next forwarding node to keep the previously established route; otherwise, if this approach fails to recover the link, NAMP will use dominant pruning to create new route using the replies along the reverse path.

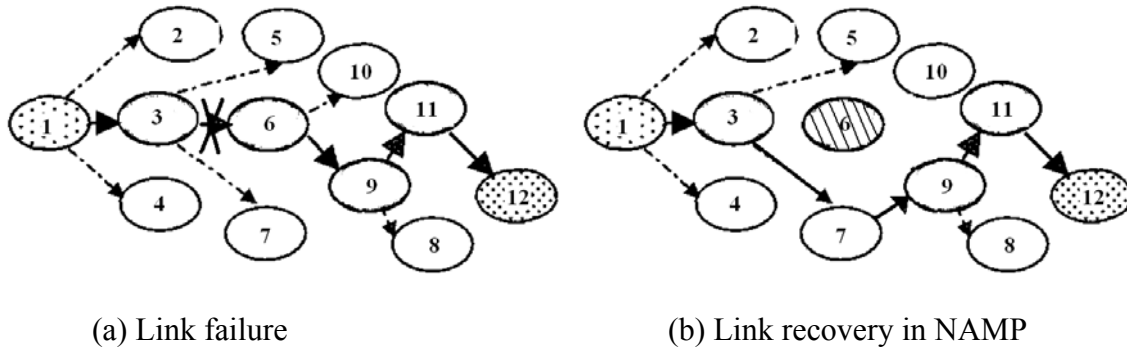


Figure 3.3 Link Failure and Link Recovery in NAMP

Suppose in previous example in Figure 3.2, link between 3 and 6 fails as shown in Figure 3.3 (a). Node 3 tries to find out destination through the neighbor nodes within two hops away. If initial link recovery attempt fails, node 3 checks its own *sfl* to select a new forwarder which can be used for recovery the link. Suppose node 7 is selected. Node 7 knows about the forwarder node 9 and a new link between 7 and 9 is established. The resulting route is shown in Figure 3.3 (b), which includes {3, 7, 9, 11}.

Generally, NAMP [6] functions to minimize the number of forwarding nodes, thus reduces possible control packets as well as decreases the number of transmissions of data packet, improves end-to-end delivery of data packet and reduces control overhead. However, it is not clearly stated how it can implement the protocol to minimize the forwarding nodes (not necessary the shortest path of course). Another concern is when certain forwarding node or link is down, it will select a backup node from *sfl* and try to keep the previously established route unchanged, which will certainly compromise the minimal-forwarding-nodes criteria. Beginning a new round of dominant pruning or selecting backup node from *sfl* really depends on which one fits the current network and results in better performance.

3.3 Packet Delivery Ratio

Packet delivery ratio is the ratio of the number of data packets actually delivered to the receivers versus the number of data packets supposed to be received. In general, three factors are responsible for missed data in ad hoc networks. The first is dropped packets due to

network link contention from control overhead, the second is dropped packets due to contention from data forwarding overhead, and the third is dropped packets due to non-existent links. The first two factors are more common in dense networks, while the third factor occurs increasingly as network node density lessens. The reliability of most basic ad hoc multicast protocols is affected by these three factors, since the protocols rely on initial best-effort data delivery.

3.3.1 Tree-based Protocol: Overlay Multicast based on Heterogeneous Forwarding (OMHF)

Overlay multicast protocols for MANET have been proposed to enhance the packet delivery by reducing the number of reconfigurations caused by non-group members' unexpected migration in tree or mesh structure. However, since data is delivered by using replication at each group member, delivery failure on one group member seriously affects all descendent members' packet delivery ratio. In addition, delivery failure can occur by collision between numbers of unicast packets where group members densely locate. In overlay multicast based heterogeneous forwarding scheme, it reduces influence of delivery failure to enhance the packet delivery ratio.

Overlay Multicast based on Heterogeneous Forwarding (OMHF) [7] tried to improve packet delivery ratio in two ways. One is to construct a new type of overlay data delivery tree, and the other is to apply a heterogeneous data forwarding scheme depending on the density of group members. Here, the former aims to minimize influence of delivery failure on one group member, and the latter intends to reduce excessive packet collision where group members are densely placed.

In order to build a new type of overlay Data Delivery Tree (DDT) [51] with delay bound, a new principle that group members having a higher expecting packet delivery ratio value locate at the upper level on the tree was introduced to minimize the influence of packet delivery failure. A max heap consists only of group members' key value that is measured by

two factors: mobility and remaining battery. This value was denoted as the node's expected packet delivery ratio (pdr). That is, the less dynamic network a host is placed in, and the more available battery remains, the higher a value of pdr is denoted to a group member.

Each group member in the network should maintain the following variables and data structure [7]:

1. Parent and children (if any): its parent and children addresses on overlay DDT.
2. fld (frequency of link dynamics): fld indicates how many neighbor links are broken in every TW (TIME WINDOW).
3. nb: node's available battery amount.
4. pdr: expected node's packet delivery ratio.
5. Delay: end-to-end delay from source to each group member.

In this protocol, a group member starts a group join process by using a modified expanding ring search to locate the closest one among group members. Nodes designated as relay point (RP) rebroadcast JOIN-REQUEST packet with increased TTL instead of joining node that requests join. Nodes at the boundary of the broadcasted TTL range are temporarily designated as RP. That is, nodes become RP if the TTL of the received JOIN-REQUEST packet becomes 0.

When a node wishes to join a specific group, it broadcasts a JOIN-REQUEST packet with the IP header's TTL field set to 1, and then waits an appropriate amount of time. When neighbor nodes receive this JOIN-REQUEST message, they automatically become RP. If there is no reply from any group member, each RP rebroadcasts a JOIN-REQUEST packet with TTL set to 2. This JOIN-REQUEST packet includes a joining node, end-to-end delay bound, as well as measured delay between each RP and joining node. When a group member receives JOIN-REQUEST, it replies joining node with a JOIN-REPLY packet as well as RP

with MEMBER-ACK packet, if end-to-end delay from source to joining node is bounded within a predetermined delay. Otherwise, the JOIN-REQUEST packet is ignored.

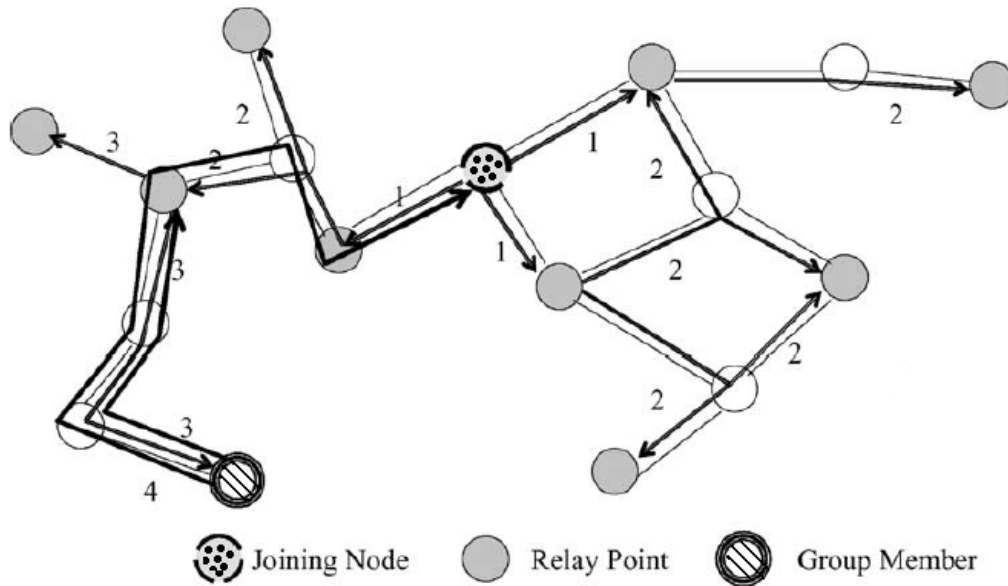


Figure 3.4 Path setup via modified expanding ring search

Figure 3.4 illustrates a path setup procedure via modified expanding ring search. The JOIN-REQUEST packet is delivered to the closest group member in three steps. After a group member receives the JOIN-REQUEST packet, it sends JOIN-REPLY to both relay point (RP) and joining node, as illustrated as the fourth step.

When a joining node receives JOIN-REPLY, on the contrary, it designates group member as its parent relay node (PRN). As an ending step, a joining node sends a JOIN-CEASE-REQUEST message to prevent other RPs from sending a JOIN-REQUEST message.

During all path setup sessions, duplicated JOIN-REQUEST packets with the same sequence number are ignored and immediately discarded. A path on overlay DDT is established with only end-to-end delay.

In OMHF [7], maintenance of overlay DDT consists of three major procedures. The first one is to handle a request for leaving a multicast group. The second procedure is to check route

availability between the PRN and itself, and last one is to reconstruct overlay DDT in a form of max heap with measured pdr value. If a group member forwards a large number of packets than other group member, it will be located near a leaf on overlay Data Delivery Tree (DDT). So packet loss caused by node congestion is limited on a lower level over overlay DDT.

Data forwarding in OMHF is mostly dependent on whether or not there is at least one zone. A flooding zone on-demand is created if members are densely distributed. Once a flooding zone is created, a data packet is broadcasted within a zone. So, it can not only drastically reduce the number of packet collisions, but also increase the packet delivery ratio since duplicated packets can be delivered from different neighboring nodes. Thus, recovery for lost packets is naturally achievable within a zone. If no zone is constructed, all data packets are forwarded along overlay DDT. Here, the zone is temporarily constructed on demand when a group member receives a data packet. Since a zone is not fixed, it is adaptive and flexible. All group members do not attempt to create their own zone. That is, some group members build their zone if they can meet all conditions.

In terms of forwarding scheme, both zone diameter and a sufficient number of neighbor group members to create a zone are very closely related to the performance of OMHF.

In OMHF, packet delivery ratio shows bad performance in a highly dynamic network with small group members [7]. This is because when logical connections over overlay DDT are broken, even though it does not frequently happen, it takes more time to designate a new parent relay node (PRN) with small group members than large group members during the path maintenance phase.

Furthermore, a dense environment is hardly formed with small group members so that the data is transmitted in the form of unicast packets in most situations. The problems existing for this protocol are that how to construct such overlay DDT so that the total hops are minimized; another issue is zone optimization problem.

3.3.2 Robust Multicasting in Ad Hoc Networks Using Trees (ROMANT)

Robust Multicasting in Ad Hoc Networks using trees (ROMANT) [8] is a modified multicast routing protocol of Multicast Ad-hoc On-Demand Distance Vector (MAODV) [43]. Based on simulation result, packet delivery ratio of MAODV is low in scenarios with high mobility or high traffic loads. A multicast tree becomes unstable and needs significant reconstruction activity when a large numbers of RREQ, RREP and MACT packets are sent. Links are assumed to be broken if neighbors do not hear each other's hello packets or hello packets are lost due to collisions in higher traffic load. If more RREQ, RREP and MACT packets are injected in an attempt to fix link breaks, it may lead to more link breaks due to the loss of more hello packets. To overcome this problem, ROMANT proposed a new approach to tree building and maintenance. In ROMANT, each receiver periodically transmits a packet called "join announcement". The join announcement contains the address of the best next-hop towards the group leader. The next-hop considers itself a member of the multicast tree after receiving a join announcement, and begins the periodic transmission of join announcement sent towards its own next-hop to the group leader. If a node doesn't hear the join announcement of its next hop it assumes that the next hop is not in range and deletes the next-hop from its connectivity list. It then sends join announcements to its next best next-hop. In this protocol, only tree members transmit join announcements in ROMANT.

Since all nodes are already aware of their next-hop to the group leader based on their connectivity list, nodes simply forward their data packets to the next-hop to the group leader. If no implicit acknowledge is received within a certain time, the next-hop is removed from the connectivity list and a new next best next-hop is chosen for future data packet forwarding. This process continues, until the data packets reach the first multicast tree member. From there, the data packet is flooded within the multicast tree with a packets cache used to drop duplicate packets.

Simulation [8] showed that this approach fixed the problem of high control overhead and low packet delivery ratio in situations of high mobility, high traffic load and a large number of

members of MAODV. Moreover, it does not perform worse in situations when the performance of MAODV is actually very good.

3.3.3 Mesh-based Protocol: Reliable On-Demand Multicast Routing Protocol (R-ODMRP)

Reliable On-Demand Multicast Routing Protocol (R-ODMRP) [9] increases the overall data packet delivery ratio by adding packet storage and retransmission operations coordinated by the multicast source. Storage responsibilities are assigned based on localized ‘neighborhoods’ of nodes with minimal spanning hop count, within the group.

In R-ODMRP the responsibility for data storage and retransmit is assigned to all receivers of the multicast group, with the source of each data stream coordinating responsibilities. All group members are divided up by the source into sets of local neighborhoods. The source sets the number of nodes per neighborhood, with the option of determining the node’s storage overhead. With each neighborhood member storing a portion of the data packets, each local neighborhood stores a distributed “sliding window” of all transmitted data packets.

In R-ODMRP, when a source initially sends out a Join Query, it becomes a Reliable Join Query (RJQuery) packet. The RJQuery packet has a timeout value attached. Once the RJQuery packet is sent, each node receiving it (whether a receiver node or not) will decrement this timer value by a preconfigured “two hop time” before sending the RJQuery downstream. After the RJQuery timer expires at each node, each receiver node will send a Reliable Join Reply (RJReply) back upstream. If a node with an expiring timer is not a receiver, it will send an RJReply only if it receives other RJReplies from downstream. The end result of the RJQuery/RJReply phase is that the source obtains a full positional listing of all receivers and forwarding group members in the network. RJQuery/Reply operations occur periodically.

The source will then set a number for the “nodes per neighborhood” count, and, with the Network Datapath table as input, partition all receiver nodes into local neighborhoods using its “Source Neighborhood” Algorithm [9]. The source then assigns data packet storage responsibilities such that the set of nodes within any given neighborhood will store the full set of data packets in sliding window fashion.

As nodes leave the group, their storage responsibilities are reassigned on new RJQuery/Reply rounds. However, as more and more nodes join over time, more neighborhoods are created and duplicate storage responsibilities will be assigned. The individual neighborhoods storing the duplicate packets will become smaller and smaller, relative to the overall network.

The second responsibility, data packet retransmission, will be initiated by a receiver node noticing a gap in data packets. It will broadcast a “Resend Request” packet to its local neighborhood, with a local time-to-live scope, listing all packets needed by sequence number. The requestor will give its ID for unicast replies. Upon receiving the packet, neighbor nodes will check their storage for the requested sequence numbers and unicast found data packets back along a single path. If the requesting node receives an incomplete reply or no reply at all, it will retain all gap sequence numbers, sending them out in its next “Resend Request”.

As the group of receivers grows in size, neighborhood partitions and node data storage responsibilities are dynamically reallocated by the source, allowing partitioned neighborhoods to be composed of a diminishing percentage of network receiver nodes that are more closely grouped. As the number of receiver nodes and neighborhoods grow in an ad hoc network, Resend Requests and replies will travel fewer hops, reducing overall network traffic. Scalability is built in to the data storage and retransmitting process.

Results show that R-ODMRP [9] does outperform ODMRP [44] under these conditions in terms of reliability, at an acceptable cost of an increase in routing efficiency and forwarding efficiency.

3.4 End-to-End Delay

In MANET, if the distances between hops are long, the transmission rates will be low. A low transmission rate incurs long transmission time, which results in low throughput and high energy consumption.

Shorter distances allow mobile nodes to be connected for a longer time when they move, increasing reliability of routes. Shorter distances between multicast nodes also permit transmissions at rates higher than the base rate, lowering transmission time, increasing network throughput, and thus reducing energy consumption. However, it does not always mean that long paths would give better performance. It is true that the longer the path, the shorter the links on that path tend to be and the higher the transmission rate that can be achieved. However more hops are needed to reach the destination. Each hop incurs contention delay for accessing wireless channels. Furthermore, the selected path will affect the traffic load of other flows on that path, and the congestion level at the nodes within the interference range of the path.

3.4.1 Rate-adaptive Multicast (RAM)

Rate-Adaptive Multicast (RAM) [10] protocol is multirate-aware. During the process of path discovery, the quality of wireless links is estimated to suggest optimal transmission rates, which are then used to calculate the total transmission time incurred by the mobile nodes on a path. Rate adaptation is the process of dynamically switching data rates to match the channel conditions. Among several considered paths from a source to a destination, RAM selects the path with the lowest total transmission time.

Rate adaptation involves two stages: channel quality estimation and rate selection. Several metrics can be used as indicators of channel quality such as signal-to-noise ratio, signal strength, symbol error rate, or bit error rate. The rate selection procedure then uses the channel quality estimate to select an appropriate rate [35].

RAM protocol considers two main issues: (1) path selection based on path weights; and (2) computing transmission rates (at the physical layer) of the nodes on the selected paths.

JOIN QUERY packets are used for estimating channel conditions, and for accumulating the weights of the links on the paths they traverse (in a field called `pathWeight`). Every node on paths between a sender and a receiver, after receiving a JOIN QUERY, measures the signal strength of the packet, which is then compared against a set of threshold values to suggest a transmission rate, `SuggestedRate`. The `SuggestedRate` is converted to a link weight, which is then added to the `pathWeight` recorded in the JOIN QUERY. After the JOIN QUERY arrives at a receiver node and is processed, field `pathWeight` contains the weight of the source-to-destination path traversed by the JOIN QUERY. The node examines all these JOIN QUERY packets and selects the JOIN QUERY, and hence the path, with the minimum `pathWeight` value.

When a node n receives a JOIN QUERY from an upstream node p , it suggests a `SuggestedRate` based on the signal strength of the JOIN QUERY, and the `SuggestedRate` is stored in the routing table of n . After n receives a JOIN REPLY from downstream (or creates a JOIN REPLY if n is a receiver), it adds the recorded `SuggestedRate` to the JOIN REPLY (in addition to updating the routing information as required by ODMRP), and sends the JOIN REPLY to p . This is the transmission rate p should use to send data to n , as suggested by the signal strength of the JOIN QUERY measured by n . Note however that p may receive several JOIN REPLY packets from different downstream nodes, and the `SuggestedRate` values in these JOIN REPLY packets may be different. Therefore p will select the minimum

SuggestedRate value among those it has seen during that JOIN QUERY consolidation interval, so as to accommodate the downstream link with the lowest quality.

3.4.2 A Priority Scheduler Using Fuzzy Logic

A priority scheduler is used so that it can schedule the packets to reach the destination quickly, which are at the verge of expiry. Without a scheduler, packets will be processed in FIFO manner and hence there is more chance that either more packets may be dropped or may not meet the QoS target. A fuzzy based priority scheduler [11] can be used. It schedules the packets based on its priority index. The priority index is attached to the header of the data packets.

This fuzzy scheduler has three inputs, i.e. Expiry Time, Data rate and Queue length, the output is the Priority index. This priority index, if very low, indicates that packets are associated with very high priority and should be immediately scheduled. If index is high, packets are scheduled after all high priority packets are scheduled.

Study in [11] gave the Priority Rule Bases table, and did the simulation. Expiry Time is variable Time-To-Live (TTL), and a packet with very low TTL will be given the highest priority. The data rate of transmission is normalized with respect to channel bandwidth. Queue length refers to the queue length of the node in which the packet is presented. If it is a highly crowded node, it will suffer excessive delay and get lost. So such a packet is given a higher priority and gets saved.

The priority index is calculated with inputs obtained from the network layer, and then is added to the header associated with the packet. Each queue in the node is sorted based on the priority index and the packet with the lowest priority index (i.e. packet with the highest priority), is scheduled next when the node gets the opportunity to send. By this method of scheduling, the overall performance increases. Simulation in the paper showed that during

low mobility, the average delay is dominated by network congestion due to data traffic. During high mobility, it is dominated by route changes.

3.5 Conclusion

The ability for nodes to form ad hoc networks in the absence of communication infrastructure is a critical area of current research. There are existing communication needs which ad hoc networks can meet, such as military and commercial applications, and the development of ad hoc network technology will enable new classes of applications. With the potential for low cost deployment and high availability, coupled with the dropping costs of wireless transceivers, ad hoc networks are becoming economically and technologically feasible right now.

Multicasting can efficiently support a wide variety of applications that are characterized by a close degree of collaboration, typical for many MANET applications currently envisioned. Within the wired network, well-established routing protocols exist to offer efficient multicasting service. As nodes become increasingly mobile, these protocols need to evolve to provide similarly efficient service in the new environment. Adopting wired multicast protocols to MANETs, which are completely lacking in infrastructure, appears less promising. These protocols, having been designed for fixed networks, may fail to keep up with node movements and frequent topology changes due to host mobility increase the protocol overheads substantially. Rather, new protocols that operate in an on-demand manner are being proposed and investigated. Existing studies [30] show that tree-based on-demand protocols are not necessarily the best choice. In a harsh environment, where the network topology changes very frequently, mesh-based protocols seem to outperform tree-based protocols, due to the availability of alternative paths, which allow multicast datagrams to be delivered to all or most multicast receivers even if links fail. Much room still exists to improve protocol performance (as measured by the packet delivery ratio) while reducing the associated overhead.

Various multicast protocols have been proposed to perform multicasting in mobile ad hoc networks. To find the proper multicast protocol for a given circumstances and scenarios, evaluation metrics such as packet delivery ratio, control overhead, delay, throughput, may be used to help satisfy the network requirements or improve performance parameters concerned.

Multicasting in ad hoc network is more challenging than that in the Internet. Mobility in wireless could cause vulnerabilities such as loss of packet, incorrect routing, and discard multicast packets because nodes in ad hoc networks are battery-power limited. Non-centralized access point or existing infrastructure, transmission range limited, and dynamic connectivity. Furthermore, group membership management is also an issue in wireless network. Because links states in wireless may undergo some inconveniences such as interference, noise, and signal attenuation, it is a problem that how to make sure either query or update information can be received by wireless nodes. Therefore, multicasting in wired network is different from that in wireless network so that it is not suitable to extend the wired multicast routing protocols to wireless.

From the approaches used for multicasting in Mobile Ad-Hoc Networks (MANETs), it can be found that improving certain evaluation metrics will affect the performance of other metric. Generally, it is always a good idea to carefully selecting the forwarding nodes/group members to form the routing structure, like tree, mesh or even hybrid one. And the awareness of neighboring nodes' information may also help when certain route fails or some data packets are missing. The maintenance scheme should be simple and effective as more control packets involved may increase not only the control overhead, but also has the tendency to reduce packet delivery ratio (PDR) in situations like high mobility or heavy traffic load.

To design a multicast protocol, one should focus on optimizing for both mobility and density. Protocol should react to link breaks, but with lower overhead. One should also be aware of the pitfalls associated with high density situations, avoiding problems such as Ack implosion and broadcast storms. While these problems have been seen in wired networks, it is apparent

that these lessons have not always carried over into wireless networking. Protocols should instead take advantage of the density created by group-based mobility patterns.

Routing protocols should not attempt to monitor packet loss and repair routes when loss is high. Sometimes, such loss may in fact be due to congestion and the increased repair traffic can lead to congestion collapse. Rather, loss monitoring should be done by the transport layer, which can then use input from the MAC layer to determine if the loss is due to mobility (and suspend sending any more data until a new route is found).

To achieve high capacity, protocols should use a bandwidth-efficient tree rather than a mesh and should have very low control overhead. Mesh-based multicast protocols increase the number of hops in the multicast tree and hence cannot support high traffic rates. This argues for a simple, end-to-end protocol design, with receivers in charge of joining group and reacting to route changes. To ask multicast forwarders to maintain the tree may result in too much control traffic, particularly when broadcast is used to repair the tree.

Some other design parameters also will affect the performance of the multicast routing protocol. Such parameters include the refresh interval selection, the accumulation delay when a node wants to collect certain information from the other nodes to make route decision, the threshold for link strength when it is used in switching from primary to backup routes. It is a good idea to take the circumstances and scenarios into consideration when designing the multicasting routing protocol as no one single approach can satisfy all situations.

Future multicast protocol evaluations – both in simulations and testbeds – need to be more comprehensive. Evaluation should consider a range of realistic mobility models and should include special cases, such as high density and high traffic rates.

Chapter 4

On-Demand Hybrid Multicast Routing Protocol

So far, to enable group communication in mobile ad-hoc network (MANET), a wide variety of routing algorithms have been proposed in the literature. Due to the dynamic network topology and limited resources of energy (battery) and bandwidth, no single multicast routing protocol can satisfy all scenarios as different scenarios can have different movement pattern, density and traffic rate depending on the environment and the nature of the interactions among the participants. For example, in a search-and-rescue operation, individuals may fan out to search a wide area, resulting in fairly regular pattern of movement, low density, and low traffic rate. In a battle-field scenario, the movement of soldiers may be heavily influenced by the movements of their commander, with higher density, and a higher traffic rate.

Given the constraints typical of the ad hoc network environment, the main requirements for ad hoc network routing protocol design can be summarized as follows:

1. Low overhead. The routing protocol needs to minimize the number and size of control messages it transmits, as well as the number of times it forwards each data packet, in order to conserve bandwidth and battery resources.
2. Adaptiveness. In order to operate efficiently in a wide range of network conditions, the routing protocol needs to be able to adapt to a high dynamic environment in which topology and propagation conditions may vary significantly on a variety of time scales.
3. Resilience to loss. The routing protocol needs to operate efficiently in the presence of (control) packet loss. The likelihood of packet loss in the ad hoc network environment may

be high, especially for multicast and broadcast packets, given that those packets are sent without acknowledgments.

The multicast routing protocol proposed in this chapter is applicable to a mobile ad hoc network that is not too sparse. It is intended for networks of devices with omni-directional antennas where a transmission by one node may be overheard by all nodes within its wireless transmission range. It also assumes that the links in the network are bi-directional.

The main goal for this multicast routing protocol is to increase packet delivery ratio while keeping the other evaluation parameters, like data overhead, control overhead and delay, at an accepted degree. As stated in Chapter 3, the packets will drop due to reasons such as:

- a) Network link contention from control overhead;
- b) Network link contention from data forwarding overhead;
- c) Non existent links.

Thus to increase packet delivery ratio (PDR), the control overhead and/or data overhead should be kept relative low so that the contentions introduced by them would be reduced. The scheme proposed also try to maintain the connectivity by preempting a link or path when the link or path in use encounters problem that would lead to the failure of data transmission.

4.1 Overview

As discussed in Chapter 2, tree-based and mesh-based multicast routing protocols have their advantages and disadvantages. Tree-based protocols are bandwidth efficient and have low routing overhead; however, they are not as reliable as mesh-based protocols under high mobility. The mesh-based protocols are relative reliable but the control packet flooding incurs high overhead; this may be even worse when the mesh updates are performed over relative short interval.

In light of the schemes introduced in Chapter 3, the protocol presented in this chapter will build the mesh underlying structure so that the nodes involved in it can store the neighbors' information in the "Message Cache" and "Routing Table". The data packets will be forwarded via the tree built upon the underlying mesh structure. In this way, only a subset of the nodes on the mesh structure will participate in the data forwarding. All nodes on the mesh will exchange their connectivity information using very small probe packets (hello packet), so that when the quality of link between two nodes on the data forwarding tree is deemed suspicious, the algorithm will switch to the link/nodes' pair with better quality to transmit the data packets to the receivers. By doing so, the underlying mesh structure remains unchanged for a relative longer time. This also implies that the mesh refresh interval would be elongated. Moreover, by choosing appropriate value that triggers link/path preempting, it may avoid depleting some nodes' battery too soon thus leading to partition of the network.

4. 2 Routing Protocol Operation

The routing protocol presented is a hybrid multicast routing protocol for MANET. When a source wants to send data packets to a group of receivers but knows little about the group membership, it will trigger the procedure to build the underlying mesh so that the information like group membership, nodes' neighborhood, connectivity, can be obtained. Once the mesh is built, the nodes on the mesh will have a better idea about their position, and the network is ready to transmit the data packets via selected nodes on the mesh. The scheme to choose paths that form the source-rooted tree is described in data forwarding (4.2.2).

4. 2. 1 Mesh Construction

Firstly, the forwarding mesh is built as traditional ODMRP [44], that is, group membership and multicast routes are established by the source on demand. It is constituted of a request phase and a reply phase.

Request phase: When a multicast source has packets to send but no route and group membership is know, it floods a member advertising packets but no payload piggybacked.

This packet, called “*Join-Query*”, is periodically broadcast to the entire network to refresh the membership information and update route. When a node receives a “*Join-Query*” packet, it stores the source address and the unique identifier of the packet to its “Message Cache” to detect duplicate. The upstream node address is inserted or updated as the next node for the source node in its “Routing Table”. If the “*Join-Query*” packet is not a duplicate and the Time-To-Live value is greater than zero, appropriate fields are updated and it is rebroadcast.

Reply phase: When a “*Join-Query*” packet reaches the multicast receiver, it creates and broadcasts a “*Join-Reply*” to its neighbors. When a node receives a “*Join-Reply*”, it checks if the next node address of one of the entries matches its own address. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the “FG_FLAG” (Forwarding Group Flag) and broadcasts its own “*Join-Reply*” built upon matched entries. The next node address field is filled in by extracting the information from its routing table. This way, the “*Join-Reply*” is propagated by each forwarding group member until it reaches the multicast source via the selected path. This process constructs the routes from sources to receivers and builds a mesh of nodes, the forwarding group. Different from traditional ODMRP, when the node realizes that it is part of forwarding group and propagates “*Join-Reply*” message, it not only sets “FG_FLAG”, but also calculate the *Packet_Delivery_Prediction* (PDP) based on the *mobility* and *remaining battery* of this very node, which is described in 4.2.3 through 4.2.5. This PDP information is attached to “*Join-Reply*” and sent to its uplink nodes. The uplink nodes will record this information along with other information associated with this node, like the *Remain_Connected_Predict_Time* (RCPT) of this uplink node and downlink node pair, which is also obtained from mobility prediction as described in 4.2.3. At the end of reply phase, the forwarding mesh is formed and every forwarding node will have PDP information of its downlink nodes and its own. Other information the node needs to record including the receiver ID and the route PDP from current node to receiver, which is the sum of all nodes’ PDP along this route.

The nodes in the forwarding group are potential nodes that would participate in the data forwarding procedure. In other word, nodes that forward data make up a subset of forwarding group in the mesh.

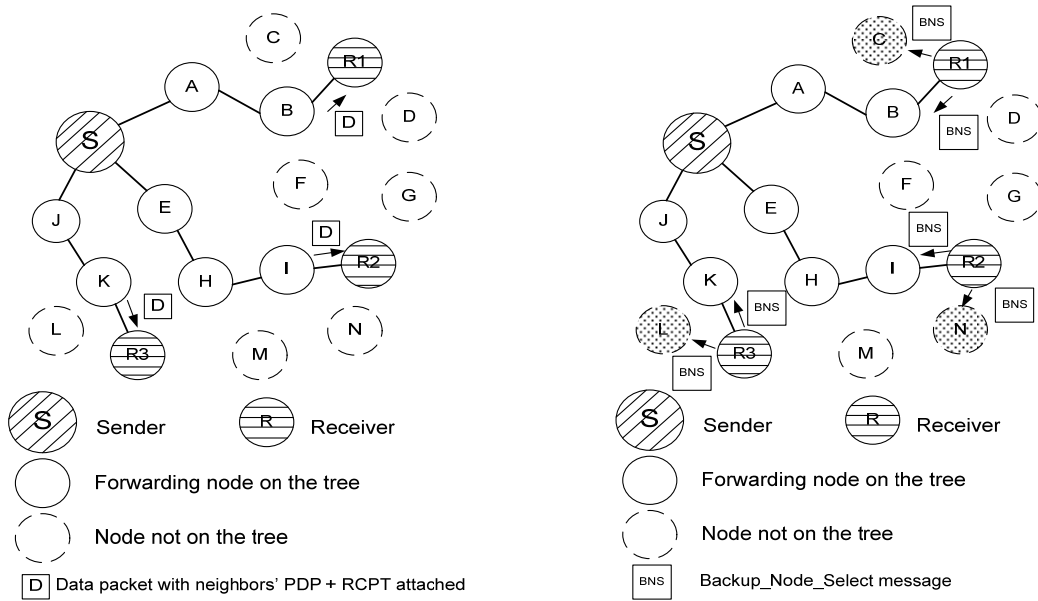
4.2.2 Data Forwarding

The data forwarding part is based on mesh structure built using mesh construction procedure, and is more like a data forwarding tree on mesh. Upon receiving “*Join-Reply*” message, the source will check the routing table it built during mesh construction, select the next hop node according to data forwarding route selection scheme describe in 4.2.5 and 4.2.6 per receiver and send data packet to that node. If multi receiver shared the same next hop node per multicast group IP address, only one copy of packet will be sent. The downlink node will select its own downlink node per receiver based on route selection scheme, just as the source node does, and transmit packets to its downlink nodes. Once a node is selected as part of this data forwarding path, it will set `PATH_SELECT`. This way, a data forwarding tree is built upon the mesh and data packets are forwarded through it.

To make sure that up-to-date information will be used to assist route selection, each node with `PATH_SELECT` flag set will count the number of packets it forwarded during a given period of time Δt so that it can update its *node_battery* value as described in 4.2.4 periodically, and change the PDP value accordingly. At the end of each Δt period, the timer and packet counter will be reset. This PDP information will be broadcasted using small hello packet to its one hop neighbors so that the neighboring nodes on the mesh, including its uplink node, will have the PDP information ready for latest data forwarding route selection (this local broadcasting will increase the link contention).

When the last one-hop node to receiver transmits data packets to receiver, it will piggyback its neighbor information (IP address, PDP, RCPT, etc.) at the end of data packet, and send it to receiver if data retransmission function is enabled. Receiver checks and chooses its preferred common neighbor based on PDP/ RCPT of that neighbor node and the backup

availability of this neighbor. The receiver then sends “*Backup_Node_Select*” packet, which includes neighbor node ID, to its uplink node and the selected neighbor node, if this common neighbor node is available. Its uplink node and the selected backup node will use this packet to set *Backup_Select* bit in the nodes’ data structure and record the uplink/backup node information in the node structure while the other nodes will discard this packet. Next time, when the same uplink node of receiver sends the data packets, the *Backup_Select* bit in the receiver’s data structure will also be set, and the selected neighbor node will also store the data packets (the other nodes that are not selected simply discard the packets). We call this node *Backup Node*.



(a) Uplink nodes start procedure

(b) Receivers select Backup Nodes

Figure 4.1 Procedure to select Backup Node for the data retransmission (optional)

In Figure 4.1, the procedure to select Backup Node for a specific receiver is illustrated. Node B, I and K are the uplink nodes for receivers R1, R2 and R3. When starting sending data packets to the receivers, the uplink nodes also attach their neighbors’ information including PDP and RCPT, which is shown in Figure 4.1 (a). Upon receiving the packet, the receivers respond by sending Backup_Node_Select message to the uplink nodes as well as their

preferred backup neighboring nodes, as shown in Figure 4.1 (b). After this procedure, nodes C, N and L are selected as Backup Node for receivers R1, R2 and R3 respectively.

When *Backup Node* finds that its PDP or RCPT to one-hop uplink node of receiver or receiver falls below certain threshold, it will send “*Backup_Node_Reselect_Request*” packet request to uplink node and receiver, and request them to exchange their neighbor information as described before and selects a new *Backup Node* if possible. Nodes involved in data transmission also have *Remain_Connected_Predict_Time* (RCPT). This RCPT gives the prediction of valid time interval for connected nodes pair.

4.2.3 Mobility Prediction

Mobility prediction is used to help select route used in data transmission. It is used to monitor the connectivity and neighborhood of the node. There are two ways to do mobility prediction [12]. One is to use free space propagation model where the received signal strength of the model solely depends on its distance to the transmitter. Thus if we know motion parameters of two neighbors, e.g. speed, direction and transmission range, we can determine the duration of time that these two mobile remain connected. GPS is needed here so that each mobile node knows its position. Once the mobile’s own position is known, its future location can be determined from the traveling speed and heading, thus we don’t need a frequent data acquisition from GPS.

Another method to achieve mobility prediction is to use received power measurements. Power samples are obtained periodically from packets received via a node’s neighbor. In this way, we can compute the rate of change for a particular neighbor’s power level. When the power level drops below the accepted level, we know that the link between these two nodes is likely to be broken.

The drawback is that, when a node is involved in a route that accelerates the power consumption, or the node changes its direction, the mobility prediction can be outdated.

However, we can have some “critical time” to compensate for this and give a predicted time that the link between certain nodes pair remains reliably connected. In this proposal, the critical time is stated as *Remain_Connected_Predict_Time* (RCPT), which is used to predict the time that two specific node pair will remain connected.

$$\begin{aligned} & \text{Remain_Connected_Predict_Time} \\ = & \begin{cases} \beta * \gamma * P_r + (1 - \beta) * \text{Remain_Connected_Predict_Time}, & P_r \geq P_{r_threshold} \\ 0, & P_r < P_{r_threshold} \end{cases} \quad (1) \end{aligned}$$

Here, P_r is the received power from a certain downlink node, β is the factor we use as smoothing factor ranged between 0 and 1, and γ is the factor we use to transfer received power P_r into something related to connection time.

For a certain node, let *Link_Dynamics_Frequency_Current* (LDFC) represent the frequency of link failure for a particular node during Δt , and *Link_Dynamics_Frequency* (LDF) represent the predicted frequency of link failure of the same node. Then we will have:

$$\text{Link_Dynamics_Frequency_Current(LDFC)} = \frac{\text{No. of link failure observed}}{\Delta t * \text{No. of neighbors}} \quad (2)$$

$$\text{Link_Dynamics_Frequency} = \alpha * \text{LDFC} + (1 - \alpha) * \text{Link_Dynamics_Frequency} \quad (3)$$

Here, α is the factor we use so that it can have memory of node’s previous mobility history. It is ranged between 0 and 1.

LDF indicates how many neighbor links are broken in every Δt . That is, this value is used to check how dynamic the environment is in which a node is currently placed. For example, the lower LDF a node has, the less link failures happened around a node. Similarly, less frequent link failure increases the possibility to receive a packet correctly [7].

4.2.4 Battery Model

Since node’s battery consumption is related to packet transmission and computation such as search for route [7], we will use *node_battery* to denote the effective battery we expect rather than the current available battery amount of the node.

$$\text{node_battery} = \frac{k * \% _ \text{of} _ \text{remaining} _ \text{amount} _ \text{of} _ \text{battery}}{\text{No.} _ \text{of} _ \text{total} _ \text{number} _ \text{of} _ \text{forwarded} _ \text{packets} _ \text{in} _ \Delta t} \quad (4)$$

Here, k is the factor for computation battery consumption ranged between 0 and 1. If every node in the network has almost the same computation load, k can be simplified as 1. Δt is the time interval during which the number of transmitted packets are counted; We also use it as the sampling period for the mobility prediction. In order to avoid infinite value, the initial number of forwarded packets is set to 1.

This model implies that the more packets are forwarded via a node than in previous time interval Δt , the more quickly the battery of the node will be expected to be completely drained out.

4.2.5 Route Selection Model (Packet Delivery Prediction)

As stated in 4.2.2, *Packet_Delivery_Prediction* (PDP) and *Remain_Connected_Predict_Time* (RCPT, 4.2.3) are used to make route selection. *Packet_Delivery_Prediction* is used to predict node's packet delivery ratio. Both are calculated distributedly for every group node, while PDP is also sent to the uplink node so that the uplink node will know the situation of its downlink node. RCPT is more about the connectivity of the link, but it may overlap with PDP as we have the PDP information for the {uplink, downlink} node pair. Both of them represent how dynamic the environment the node is placed.

PDP value is the major factor that a node uses to make decision on which downlink node to choose as next hop node to send the data packet. The node's capability to correctly receive a data packet is defined with two factors: surrounding link dynamics and remaining available battery. We have defined and calculated information on *Link_Dynamics_Frequency* (LDF) in 4.2.3, which is used to define the surrounding link dynamics, and *node_battery* in 4.2.4, which represents the remaining efficient battery. For every group node, PDP is given as:

$$\text{Packet_Delivery_Pr ediction} = \text{node_battery} * \frac{1}{\text{Link_Dynamics_Frequency}} \quad (5)$$

The data forwarding route decision can use either fixed prediction threshold based scheme or adaptive prediction threshold based scheme. Variable *downlink_node (ID)* is used to denote whether there's valid route from node with identified number, *ID*, to the receiver. Once *downlink_node (ID)* is set to *null*, it means there's no possible route for the current node to the receiver, thus this node will inform its uplink node to select a different downlink node, or the whole route from source to current node expire and a totally different data forwarding route will be selected, or the mesh should be re-constructed and new forwarding group will be selected. In this case, the node will send "*Mesh_Refresh_Request*" message backward to source, asking uplink node to build a difference branch, or even forcing source to send "*Join-Query*" message and reconstruct the mesh in the worst case.

4.2.5.1 Fixed Prediction Threshold Based Data Forwarding Route Selection

In fixed prediction threshold scheme, uplink node uses a fixed given prediction to compare against *Packet_Delivery_Prediction (PDP)* value of every possible downlink node and decides which node to use as the next hop node on the route. The fixed prediction threshold based data forwarding route decision is made by:

$$\begin{cases}
 \text{downlink_node}(ID) \\
 \left\{ \begin{array}{l}
 = \text{node_with_max}(\text{Packet_Delivry_Predictio}(i) | \text{Remain_Connected_Predict_Tim}(i) > 0), \\
 \text{for all downlink nodes } PDR(i) > PDP_threshold \\
 \\
 = \text{null}, \\
 \text{for all downlink nodes } PDR(i) < PDP_threshold \text{ or } \text{Remain_Connected_Predict_Tim}(i) = 0
 \end{array} \right.
 \end{cases}$$

According to the scheme, the uplink node will select the node that could remain connected ($RCPT > 0$) and has the greatest *Packet_Delivery_Prediction (PDP)* value. If none of the potential downlink nodes could satisfy the PDP and RCPT requirements, the uplink node will unicast the "*Mesh_Refresh_Request*" message backward to its own uplink node so that it can choose alternative downlink node. In the worst case, this "*Mesh_Refresh_Request*" message

will be unicasted all the way to source and force source to reconstruct the mesh. Details of mobility model and battery model were introduced in 4.2.3 and 4.2.4.

The design parameter in this scheme is the PDP_threshold. It should be large enough to assure that the nodes on the path are reliable enough to transmit the data packets for at least the refresh interval of PDP; it also should be small enough so that the selected path is viable long enough and the mesh reconstruction will not occur too often.

4.2.5.2 Adaptive Prediction Threshold v.s. Fixed Prediction Threshold

The fixed prediction threshold data forwarding route selection worked fine when there were enough candidate routes for the sender-receiver pairs with high Packet_Delivery_Prediction (PDP) and enough Remain_Connected_Predict_Time (RCPT) to transmit the data packets. Once there's no single route with enough Packet_Delivery_Prediction (PDP) between sender-receiver pairs, or some nodes along the route have high prediction value where other nodes along it have comparative low prediction value, it may be better to reconstruct the mesh structure so that new data forwarding tree will contain nodes with relative higher PDP, and no node in the network may be drained out of node battery too fast. Generally, there are two ways to balance the power consumption of transmitting packets through nodes:

The first method is selecting a relative high PDP_threshold (Packet_Delivery_Prediction threshold) so that once there's no route satisfies this threshold constrain from a sender-receiver pair, even though the nodes still have PDP values large enough to maintain the data packet transmission for a long enough reliable time. This will force the underlying mesh structure to reconstruct and find new data forwarding tree with possibly new PDP value of the nodes. The pros are it will choose the routes with highest reliable prediction value to transmit the packets and it helps balancing the energy (battery) consumption of packet transmission throughout the networks. The cons are it is updated more frequently than necessary, and may introduce additional packet loss due to mesh reconstruction if no backup routes from previous data forwarding tree is provided.

The second method is selecting two PDP_threshold, denoted as PDP_threshold_hi and PDP_threshold_lo, for the current mesh structure. The data forwarding route decision is modified as:

$$\begin{cases}
 \text{downlink_node}(ID) \\
 = \text{route_with_max}(\sum \text{Packet_Delivery_Prediction}(i) | \text{Remain_Connected_Predict_Time}(i) > 0), \\
 \quad \text{for all downlink nodes } PDR(i) \geq PDP_threshold_hi \\
 \\
 = \text{node_with_max}(\text{Packet_Delivery_Prediction}(i) | \text{Remain_Connected_Predict_Time}(i) > 0), \\
 \quad \text{for all downlink nodes } PDP_threshold_lo \leq PDR(i) < PDP_threshold_hi, \\
 \quad \text{start to construct new mesh structure} \\
 \\
 = \text{null, all } PDR(i) < PDP_threshold_lo - \text{critical_value, network partitioned}
 \end{cases}$$

When the PDP is greater or equal to the PDP_threshold_hi, the data forwarding route selection chooses the routes with largest PDP summation of the nodes, this way, the most reliable routes are chosen; When PDP of at least one node drops between PDP_threshold_lo and PDP_threshold_hi, the route tries to choose the nodes with largest PDP, which is a little different from when PDP of every node is greater or equal to PDP_threshold_hi; the new mesh is constructed when any node's PDP is very close to PDP_threshold_lo. The old forwarding group information is still stored and used until new forwarding group information is ready. If all downlink nodes' PDP of a certain node is lower than PDP_threshold_lo minus critical_value, it means even with new mesh structure, the route with enough PDP for the intermediate nodes can not be found and network may be partitioned. The pros for this method are the packets are delivered with high probability of successful transmission while the mesh won't be constructed too often; The cons are there will be more store capacity required for each node to ensure that they can store the node information from previous mesh structure, while they must store new node information for the latest mesh structure so that they can switch to new ones. What's more, they may need to have some buffer for data packets to prepare for the possible route "handoff". The *critical_value* is decided by the size of the whole network and the power consumption rate of the node.

4.2.6 Data Retransmission (optional)

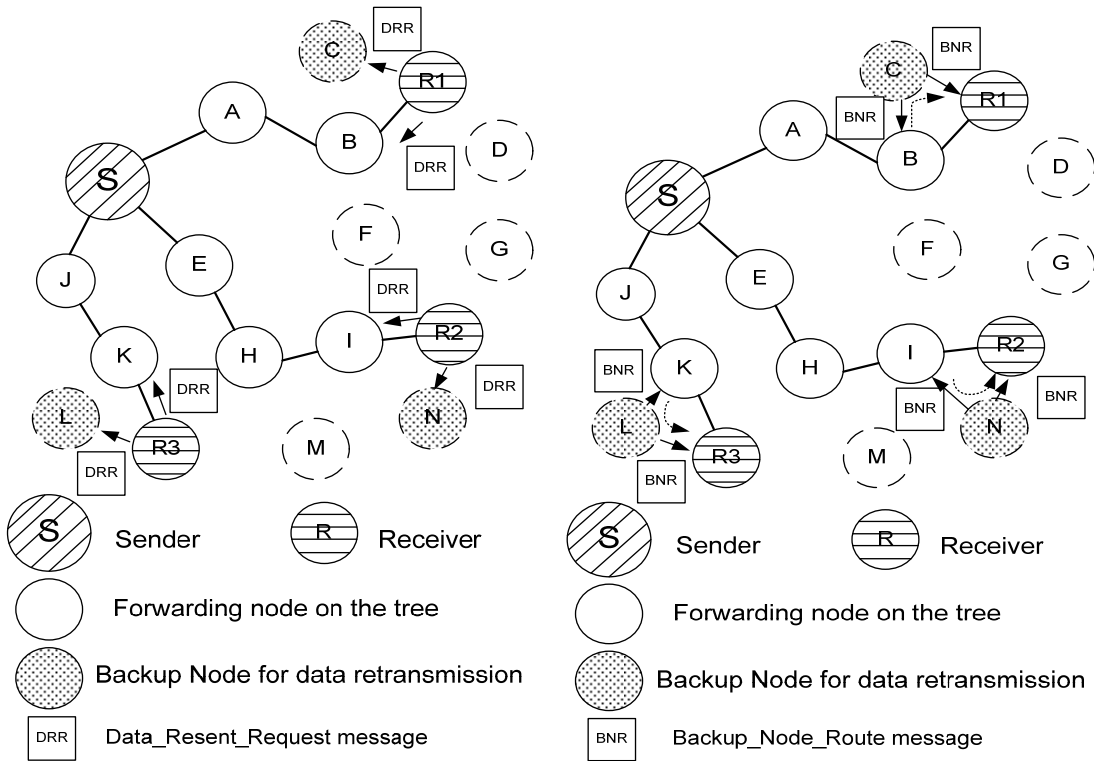
Data retransmission is used when some data packets are missing and the receiver(s) requests resending these missing packets. A mechanism of storing and resending data packets can be used to solve this. By using endpoint delivery of missing data packets, the mechanism introduced here tries to keep the increase in network contention due to new control overhead and data packet resend overhead as minimal as possible.

Neighbor nodes on the mesh but not participate in the data forwarding that can overhear data transmission from the current data forwarding node to the receiver will store the data packets transmitted during a given period and retransmit data packets if the receiver finds out that certain data is missing. The problem here is: If there's no node near the receiver, there could be no data retransmission because of the lack of neighbor nodes; If there's a lot forwarding group members around the current data forwarding node and the receiver, more than enough redundant data store and retransmission may occur. To solve this, receiver and the current one-hop node from receiver on the data forwarding tree can exchange their neighbor information. Once being selected as the data forwarding node with `PATH_SELECT` flag set, the uplink node of the receiver can piggyback the neighbor information at the end of initial data packet, and send it to receiver. Receiver then checks and figures out its preferred common neighbor based on PDP of that neighbor node and the availability of this neighbor, which can not be shared with other receiver. The receiver then sends "*Backup_Node_Select*" message to its uplink node and the selected neighbor node to set the *Backup_Select* bit in their data structure. If no available common neighbor is found, "*Backup_Node_Select*" message will not set any bit in the uplink node's data structure. As stated in 4.2.2, when uplink node of receiver sends the data packets, the selected neighbor node, also known as *Backup Node*, will store all data packets sent from uplink node of receiver to receiver.

Each receiver can have at most one out-of-data-forwarding-tree neighbor that can participate in data retransmission. That is to say, data retransmission is not always available for the

receiver. This is especially true when the node mobility speed is really high or there's not enough neighboring node around. Data retransmission is optional to the proposed multicast routing protocol as it may not provide expected result while it may introduce some forwarding overhead. Even though the simulation in Chapter 5 does not include data retransmission in the proposed protocol, the procedure for data retransmission is described in the following paragraphs.

When the receiver finds out that certain data packet is missing and its "Backup_Select" bit is set in the data structure, it will ask for data retransmission. It sends "*Data_Resent_Request*" message as well as the missing data sequence number to its uplink node and *Backup Node*, and the *Backup Node* will reply by sending "*Backup_Node_Route*" message to both current uplink node of receiver and receiver. The uplink node then sends data packets to this *Backup Node*. The data packets will be queued in the buffer of the *Backup Node* and proper parts will be sent to the receiver by *Backup Node*.



(a) Receivers send Data_Resent_Request (b) Backup Nodes reply by BNR

Figure 4.2 Data Retransmission Procedure (optional)

In Figure 4.2, data retransmission procedure is shown. When receivers miss certain data packets and their “Backup_Select” bit is set in the data structure, they send

“Data_Resent_Request” to both their uplink nodes and Backup Nodes (Figure 4.2 (a)). The

Backup Nodes reply by sending “Backup_Node_Route” message to the uplink node as well as receivers (Figure 4.2 (b)). The data packets will be forwarded via the Backup Node instead

of directly being sent to the receiver. In the figure, the data forwarding paths will be B->C->R1, I->N->R2 and K->L->R3.

When the receiver has received the missing data, it will send “Route_Switch_Back” packet to both Backup Node and original uplink node and reset control bit in their nodes’ structure.

Then the original uplink node will send data packets directly to receiver and they can negotiate to select a new Backup Node if the one that has been used no longer meets the criteria to store and resend packets.

When the Backup Node “*Backup_Node_Reselect_Request*” message will be used when the current Backup Node’s PDP or RCPT for the links connected to the uplink node of receiver and receiver falls below the predefined threshold (4.2.2). This process is the same as when they first select the Backup Node.

It is obvious that, to make sure that most of the data packets will be received by the receiver, each node should have some buffer size so that they can be used in cases like data retransmission.

4.2.7 Mesh Maintenance

If source has data packet to send, it will periodically send “*Join-Query*” packets. This will refresh the forwarding group information and help build updated data forwarding tree. The refresh interval should be chosen so that there won’t be too many control message overflows. Both `MIN_REFRESH_INTERVAL` and `MAX_REFRESH_INTERVAL` will be needed as `MIN_REFRESH_INTERVAL` should be enforced to avoid control messages overflow; while `MAX_REFRESH_INTERVAL` should be set so that the mesh and forwarding nodes’ information are fresh enough.

As described in 4.2.5, when there’s no satisfactory data forwarding route for certain receiver, the uplink node involved in this route selection will send “*Mesh_Refresh_Request*” backward to source and force source to send “*Join-Query*” and reconstruct the mesh even though the time interval between this “*Join-Query*” and last “*Join-Query*” is shorter than original intended refresh interval. This way, the nodes in the forwarding group will have latest information about their neighbors that are on the mesh, and build the new data forwarding routes accordingly. The design parameters, like `PDP_threshold_lo` for the Data Forwarding Route Selection scheme in 4.2.5.2, will affect the mesh reconstruction rate because as long as one intermediate node for certain data route drops below this threshold and no other node above this value, “*Mesh_Refresh_Request*” message is unicasted and mesh reconstruction is

enforced. Other parameters that are involved in calculating PDP and RCPT will also affect mesh reconstruction rate.

If a receiver wants to leave the network, it just doesn't respond to *“Join_Query”*.

If any node wants to leave the network, it also won't retransmit or respond to *“Join_Query”*.

4.3 Message Format

There are three messages for mesh construction: *“Join_Query”*, *“Join_Reply”* and *“Mesh_Refresh_Request”*.

There are five messages for backup node selection and data retransmission:

“Backup_Node_Select” and *“Backup_Node_Reselect_Request”* for Backup Node selection, *“Data_Resent_Request”*, *“Backup_Node_Route”*, *“Route_Switch_Back”* for data retransmission. Some ACK packets can be added to ensure the corresponding packets are received by intended receiver.

Some other possible control message may include *“PDP_Update”* packet to update downlink nodes' PDP and for current data forwarding tree. But this packet can be eliminated by carefully choosing PDP_threshold_hi and PDP_threshold_lo value (4.2.5.2), or by decreasing the recorded downlink node's PDP based on number of packets sending via that node along with other consideration. By eliminating certain control packet, there may be less collision thus packet delivery ratio can increase.

4.3.1 Mesh Construction Message Format

a. “Join-Query”

The packet format for this message is as follows:

“Type + Reserved + Time-To-Live + Hop Count + Multicast Group IP Address + Sequence Number + Source IP Address + Previous Hop IP Address”

Here “Type” defines that it is “Join-Query” packet;

“Reserved” defines that the node ignores all packets;

“Time-To-Live” defines the number of hops this packet can traverse;

“Hop Count” defines the number of hops traveled so far by this packet; it is initialized to 0 by the source of the packet and is incremented by each node forwarding the packet.

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by the source to uniquely identify the packet;

“Source IP Address” defines the IP address of the node originating the packet;

“Previous Hop IP Address” defines the IP address of the last node that has processed this packet.

b. “Join-Reply”

The packet has the following format:

“Type + Count + R + F + Reserved + Multicast Group IP Address + Receiver IP address + Previous Hop IP Address + Sequence Number + Sender IP Address [1] + Next Hop IP Address [1] + Remain_Connected_Predict_Time [1] + Sender IP Address [2] + Next Hop IP Address [2] + Remain_Connected_Predict_Time [2] + ... + Sender IP Address [n] + Next Hop IP Address [n] + Remain_Connected_Predict_Time [n] + Packet Delivery Prediction (PDP)”

Here “Type” defines that it is “Join-Reply” packet;

“Count” defines the number of (Sender IP Address, Next Hop IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“F” defines Forwarding Group flag. This flag is set when the packet is transmitted by a forwarding group node;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Receiver IP address” defines the IP address of the receiver in the multicast group; If this information has to be used, each receiver node will most likely have two IP addresses, one for the multicast group, and the other for individual IP address. If this is not feasible for Mobile Ad Hoc Network, additional information other than this address will be needed so that when building the data forwarding tree, each receiver will be the leaf of the data forwarding tree and can successfully receive data packets.

“Previous Hop IP Address” defines the IP address of the last node that has processed this packet;

“Sequence Number” defines the sequence number assigned by the previous node to uniquely identify the packet;

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group;

“Next Hop IP Address [1...n]” defines the IP addresses of the next nodes this packet is target to;

“Remain_Connected_Predict_Time [1...n]” defines the predicted time the previous hop node will keep connected with current node. This value can be calculated using Mobility Prediction Model stated in 4.2.3. Here for each {uplink node, downlink node} pair, it will have specific Remain_Connected_Predict_Time (RCPT) value. If this value is not easy to extract for every node pair, we can choose and store some predefined RCPT value in the uplink node, and let uplink node decide what the RCPT value should be given to the “link” to a specific downlink node based on its PDP and downlink node’s PDP value;

“Packet Delivery Prediction (PDP)” defines the stability of the previous hop node based on mobility prediction model and battery model in 4.2.3 to 4.2.5.

When the mesh is built, the data packet should be forwarded following the data forwarding route selection scheme in 4.2.5.

c. “Mesh_Refresh_Request”

The packet has the following format:

“Type + Count + R + Reserved + Multicast Group IP Address + Receiver IP address + Previous Hop IP Address + Sequence Number + Sender IP Address [1] + Next Hop IP Address [1] + Sender IP Address [2] + Next Hop IP Address [2] + ... + Sender IP Address [n] + Next Hop IP Address [n]”

Here “Type” defines that it is “Mesh_Refresh_Request” packet;

“Count” defines the number of (Sender IP Address, Next Hop IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“F” defines Forwarding Group flag. This flag is set when the packet is transmitted by a forwarding group node;

“Reserved” defines that the node ignores all packets;

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group that use this node as intermediate node to transmit data packet to receivers.

“Next Hop IP Address [1...n]” defines the IP addresses of the next nodes this packet is target to. The IP address is extracted from the node’s routing table that is used for current data forwarding (PATH_SELECT_Flag set for the routing table);

4.3.2 Backup Node Selection and Data Retransmission (optional)

a. “Backup_Node_Select”

This message is used for Backup Node selection. The packet has the following format:

“Type + count + R + Reserved + Multicast Group IP Address + Receiver IP address + Sequence Number + Sender IP Address [1] + Uplink_node IP address [1] + Backup_Node IP Address [1] + Sender IP Address [2] + Uplink_node IP address [2] + Backup_Node IP

Address [2] + ... + Sender IP Address [n] + Uplink_node IP address [3] + Backup_Node IP Address [n]

Here “Type” defines that it is “Backup_Node_Select” packet;

“Count” defines the number of (Sender IP Address, Uplink_node IP address, BackupNode IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Receiver IP address” defines the IP address of the receiver in the multicast group;

“Sequence Number” defines the sequence number assigned by the receiver node to uniquely identify the packet;

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group;

“Uplink_node IP address [1...n]” defines the IP address of the next hop link address for the data forwarding tree that rooted at Sender IP address [1...n];

“Backup_Node IP Address [1...n]” defines the Backup Node IP address for a given source IP rooted tree that will work as Backup node for the last hop data retransmission.

b. “Backup_Node_Reselect_Request”

This message is used for Backup Node selection. The packet has the following format:

“Type + count + R + Reserved + Multicast Group IP Address + Sequence Number + Sender IP Address [1] + Uplink_node IP address [1] + Receiver IP Address [1] + Sender IP Address [2] + Uplink_node IP address [2] + Receiver IP Address [2] + ... + Sender IP Address [n] + Uplink_node IP address [3] + Receiver IP Address [n]”

Here “Type” defines that it is “Backup_Node_Reselect_Request” packet;

“Count” defines the number of (Sender IP Address, Uplink_node IP address, Receiver IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by the receiver node to uniquely identify the packet;

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group;

“Uplink_node IP address [1...n]” defines the IP address of the next hop link address for the data forwarding tree that rooted at Sender IP address [1...n];

“Receiver IP address [1...n]” defines the IP address of the receiver in the multicast group for a given source IP rooted tree that will choose the node as Backup node for the last hop data retransmission.

c. “Data_Resent_Request”

This message is used for data retransmission. The packet has the following format:

“Type + R + Reserved + Multicast Group IP Address + Sequence Number + Sender IP Address + Uplink_node IP address + Backup_Node IP Address + Missing_Data_Sequence_Number”

Here “Type” defines that it is “Data_Resent_Request” packet;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by the receiver node to uniquely identify the packet;

“Sender IP Address” defines the IP addresses of the sources of this multicast group;
“Uplink_node IP address” defines the IP address of the next hop link address for the data forwarding tree that rooted at Sender IP address;
“Backup_Node IP Address” defines the Backup Node IP address for this given source IP rooted tree that works as Backup node for the last hop data retransmission.
“Missing_Data_Sequence_Number” defines the missing data sequence number that the receiver wants.

d. “Backup_Node_Route”

This message is used for data retransmission. This message is the reply message to “Data_Resent_Request” message initially sent by receiver. It is used to inform both uplink node of receiver and receiver that the data packet will be transmitted via the Backup Node to receiver. The packet has the following format:

“Type + R + Reserved + Multicast Group IP Address + Sequence Number + Sender IP Address + Uplink_node IP address + Backup_Node IP Address + Receiver IP address”

Here “Type” defines that it is “Backup_Node_Route” packet;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by Backup Node to uniquely identify the packet;

“Sender IP Address” defines the IP addresses of the sources of this multicast group;

“Uplink_node IP address” defines the IP address of the next hop link address for the data forwarding tree that rooted at Sender IP address;

“Backup_Node IP Address” defines the Backup Node IP address for this given source IP rooted tree that works as Backup node for the last hop data retransmission.

“Receiver IP address” defines the IP address of the receiver in the multicast group for a given source IP rooted tree that chooses the node as Backup node for the last hop data retransmission.

e. “Route_Switch_Back”

This message is used for data retransmission. It is used to inform the original uplink node of the receiver and Backup Node that the data packet will be transmitted directly from uplink node to receiver. It will help reset certain bit in the corresponding nodes’ data structure. The packet has the following format:

“Type + R + Reserved + Multicast Group IP Address + Sequence Number + Sender IP Address + Uplink_node IP address + Backup_Node IP Address + Receiver IP address”

Here “Type” defines that it is “Route_Switch_Back” packet;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by the receiver node to uniquely identify the packet;

“Sender IP Address” defines the IP addresses of the sources of this multicast group;

“Uplink_node IP address” defines the IP address of the next hop link address for the data forwarding tree that rooted at Sender IP address;

“Backup_Node IP Address” defines the Backup Node IP address for this given source IP rooted tree that works as Backup node for the last hop data retransmission.

“Receiver IP address” defines the IP address of the receiver in the multicast group for a given source IP rooted tree that chooses the node as Backup node for the last hop data retransmission.

“Backup_Node_Route” and “Route_Switch_Back” have the same packet format except the packet type is different.

4.4 Data Node Structures and Routing Table

To make sure that the data packets will be transmitted via selected routes and when certain data packets are missing, data retransmission can be performed through backup nodes, each forwarding group member will maintain the following variables and data structures:

- a. The uplink nodes' and downlink nodes' addresses.
- b. The source IP addresses and receiver Addresses/Multicast IP address.
- c. Timer Δt : used to count packets sent from this node during certain period of time. This can be a global predefined value so that every node will count the number of packets during the same time span. It is counted down from MAX_time to 0 and repeats.
- d. Packet counter: Used to count the number of packets that are sent during timer Δt . The value is reset at the beginning of every time period.
- e. The node_battery value from battery model 4.2.4.
- f. Packet delivery prediction (PDP) based on mobility prediction in 4.2.3 and battery model in 4.2 4.
- g. Remain_Connected_Predict_Time for a specific downlink node.
- h. PATH_SELECT flag bit. If the node is selected in data forwarding route, PATH_SELECT flag bit is set to 1; otherwise 0.
- i. Backup_Select flag bit. If the node is selected by receiver and uplink node of receiver that is currently part of data forwarding tree, Backup_Select bit is set to 1; otherwise 0.
- j. BK_Uplink_Receiver_Addresses. Addresses of the receiver and uplink node of receiver pair that selects the node as backup node for data retransmission. This is valid when Backup_Select_Flag is set to 1.

- k. `Have_Backup_Node_Flag` bit. This bit is set to 1 if a receiver and its uplink node that is currently part of data forwarding tree agree that they will have backup node for data retransmission.
- l. `Backup_Address`. The receiver and its uplink node that is currently part of data forwarding tree will store this `Backup_Address` of backup node so that when data retransmission is needed, backup node can join the route. The data flow is then uplink node -> backup node -> receiver during data retransmission.
- m. `Data buffer size`: it defines the size of the data packets that will be stored in case data retransmission is requested.
- n. `FG_FLAG`: Forwarding Group flag. It is set to 1 if the node is a Forwarding node.

4.5 Summary

In this chapter, a hybrid on-demand multicast routing protocol for multi-hop mobile ad hoc network has been presented. This protocol uses forwarding group nodes to form mesh as underlying structure, and sends data using data forwarding tree, which is constituted only of a sub-set of the forwarding group. The branches on the source-rooted tree are built using Packet Delivery Prediction (PDP) as the major path selection criterion. When the receiver requests the missing data packets, data retransmission is assisted by selected neighbor node known as Backup Node. The performance of this protocol is evaluated in Chapter 5.

Chapter 5

QoS-disabled Protocol Evaluation

This chapter describes the methodology for evaluating protocol in ad hoc network environment, and the performance evaluation results from simulation. The advantage of simulation in studies of protocols is that it allows experimental control: experiments can be designed at will and then rerun while varying an experimental variable and holding all other variables constant. With simulation, it is possible to test the behavior of networks with more nodes than available physical equipment, or networks with equipment that does not even exist yet. The drawback of simulation in studies is that it inherently runs the risk of oversimplification. It is not possible to exactly replicate the entire world inside a computer model. When creating a simulation, some factors must be approximated. The failure to properly capture the behavior of first-order factors can lead to dramatically incorrect results.

5.1 Simulation Tools and Scenario Setup

NS (network simulator) is a discrete event simulator developed by the University of California at Berkeley and the VINT project [56]. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The Rice Monarch Project [57] has made substantial extensions to the ns-2 network simulator that enables it to accurately simulate mobile nodes connected by wireless network interfaces, including the ability to simulate multi-hop wireless ad hoc networks.

5.1.1 Simulating Mobile Ad-hoc Networks

Node model: The model for each node has a position and a velocity, and moves around on a topology that is specified using either a digital elevation map or a flat grid. The position of the node can be calculated as a function of time, and is used by the radio propagation model

to calculate the propagation delay from one node to another and to determine the power level of a received signal at each mobile node.

Each node has one or more wireless network interfaces, with all interfaces of the same type (on all mobile nodes) linked together by a model of a single physical channel. When the model of a network interface transmits a packet, it passes the packet to the appropriate physical channel object. This object then computes the propagation delay from the sender to every other interface on the channel and schedules a “packet reception” event for each. This event notifies the model of the receiving interface that the first bit of a new packet has arrived. At this time, the power level at which the packet was received is compared to two different values: the carrier sense threshold and the receive threshold. If the power level falls below the carrier sense threshold, the packet is discarded as noise. If the received power level is above the carrier sense threshold but below the receive threshold, the packet is marked as a packet in error before being passed to the MAC layer. Otherwise, the packet is simply handed up to the MAC layer.

Medium Access Control: The link layer of the simulator includes a simulation model of the complete IEEE 802.11 standard [58] Medium Access Control (MAC) protocol Distributed Coordination Function (DCF) in order to accurately model the contention of nodes for the wireless medium. DCF is designed to use both “physical carrier sense” and “virtual carrier sense” mechanisms to reduce the probability of collisions due to hidden terminals. Virtual carrier sense attempts to “sense” the presence of carrier near the intended receiver of a packet before transmitting. If either carrier sense mechanism indicates that the wireless medium is busy, the node defers before transmitting, using a binary backoff.

The virtual carrier sense mechanism uses two short packets before the intended data packet to acquire the channel: a “*Request-to-Send*” (RTS) and a “*Clear-to-Send*” (CTS). When a node A wants to transmit a packet to node B, it first transmits an RTS to B, and upon receipt of this packet, node B responds with a CTS to A if both its physical and virtual carrier sense

mechanisms indicate that the medium is idle. When the CTS is received by A, A transmits the data packet, and when B receives the data packet, it returns a link-layer Acknowledgement (ACK) packet to A. All packets sent by 802.11 carry in their headers a *Duration* field that indicates how much longer nodes that receive the packet should consider virtual carrier to be presented, and nodes listen to and obey the virtual carrier implied by the *Duration* field in all packets that they receive, even if not addressed to them.

Simulation Methodology: The basic methodology for simulation is as follows: Each run of the simulation accepts an input *scenario* file that describes the exact motion of each node and the exact sequence of packets originated by each node, together with the exact time at which each change in motion or packet origination is to occur. The detailed trace file created by each run was stored to disk, and analyzed using a script, like “perl” or “python”, to extract information like number of packets successfully delivered and the length of paths taken by the packets. This data was further analyzed in Matlab or Excel to produce the graphs used in this thesis.

5.1.2 Scenario Setup

The performance of proposed protocol is simulated through simulation in a variety of multicast communication scenarios using NS-2 network simulator with mobile NS-2 extension.

The simulation simulates a MANET with 100 nodes randomly distributed over an area of 1000m x 1000m, and each run lasts for 1000 seconds. The radio channel capacity for each mobile node is 2Mb/s, using the IEEE 802.11b DCF MAC layer and a communication range of 250m. The movement model of the nodes in the simulations is the Random Waypoint model. Each node independently starts at a random location in the simulation area and remains stationary for a period of pause time. The node then generates a uniformly distributed new location, which is a random destination inside the simulation area, to move to. At the same time, the node randomly selects a speed value uniformly distributed from a

speed range, say, 0 m/s to 20 m/s in the simulations, which is 10m/s as mean speed, and moves to a new location at the selected speed. This movement pattern is repeated for the duration of the simulation. A pause time of 0 seconds corresponds to a continuous motion whereas a pause time of 1000 seconds corresponds to a static scenario.

The source node generates CBR (constant bit rate) traffic. The above movement scenarios, together with the multicast traffic scenarios, are generated once for each run, and are adopted by all three protocols, ODMRP, MAODV and proposed protocol for a fair comparison. That is to say, each of the approaches has been evaluated over the same pre-generated set of scenarios with varying movement patterns and number of group members (receivers).

Only one kind of traffic load was tested consisting of 1 CBR source for the same multicast group for different mobility speed, and for different receivers for the same mobility speed. The duration of each simulation run is 1000 seconds. The CBR source starts sending data and receivers join the multicast group at a uniformly distributed time within the first 180 seconds of the simulation. The source generates 330 bytes data packets at a rate of 5 packets per seconds (13.2 kb/s).

To assess the effectiveness of the different protocols, the following metrics are used for the evaluation of ODMRP, MAODV, and proposed protocol:

- a. Packet Delivery Ratio: The average number of multicast DATA packets actually received by one destination node over the number of transmitted packets from the source. For example, with five destinations and 1000 generated DATA packets at the source, if 4500 packets are received in total by the 5 destinations, or 900 on average for each node, the delivery ratio is 0.9.
- b. Data Overhead: The average number of DATA packet relays per received DATA packet. This metric represents the efficiency of the multicast forwarding.

- c. Control Overhead: The average number of control packet relays per received DATA packet.
- d. Average end-to-end delay (EED): The end-to-end delay of every packet received at every receiver is recorded; the average over all the packets received is then computed.

Finally, as it may happen that two nodes may be select as next hop to a destination because both of them has the same *pdr*, then a tie resolution method has been proposed. In that case, the node with lowest ID accepts being a forwarding node for the other node, and selects another neighbor towards the destination. To detect this situation, a new field is added to the JOIN_QUERY message which includes the neighbor that the sender of the JOIN_QUERY message selected as its next hop towards the source. This was not a problem in the original specification of ODMRP because alternative routes (different from the shortest path) were detected as duplicate control packets and were not processed.

For ODMRP, the REFRESH_INTERVAL was fixed at 3 seconds and the FG_FLAG timeout was fixed at 3 times the REFRESH_INTERVAL. The maximum number of JOIN_REPLY retransmissions was fixed at 3 and the time that a node waits before sending a JOIN_REPLY (in case it can aggregate several of them in a single message) was 0.025 seconds. In addition, for the modified variant of ODMRP we configured the source aggregation timeout, being the number of seconds to wait for better routes before propagating the JOIN_QUERY to be 0.015 seconds.

For MAODV, the Hello interval was fixed at 2 seconds, the Group-Hello interval was fixed at 5 seconds, the time to wait to receive a MACT was fixed at 20 seconds, re-transmission timeout was fixed at 0.75 seconds, and route expiration timeout was 3 seconds.

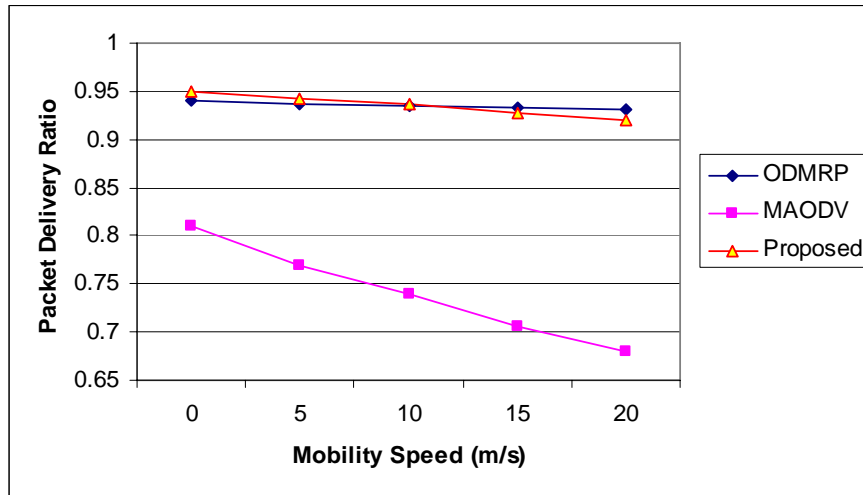
For proposed protocol, the MESH_REFRESH_INTERVAL was set as 30 seconds. MIN_REFRESH_INTERVAL was set to 3 seconds, while MAX_REFRESH_INTERVAL was set to 30 seconds. The Hello interval was fixed at 10 seconds.

5.2 Comparison Evaluation with MAODV and ODMRP

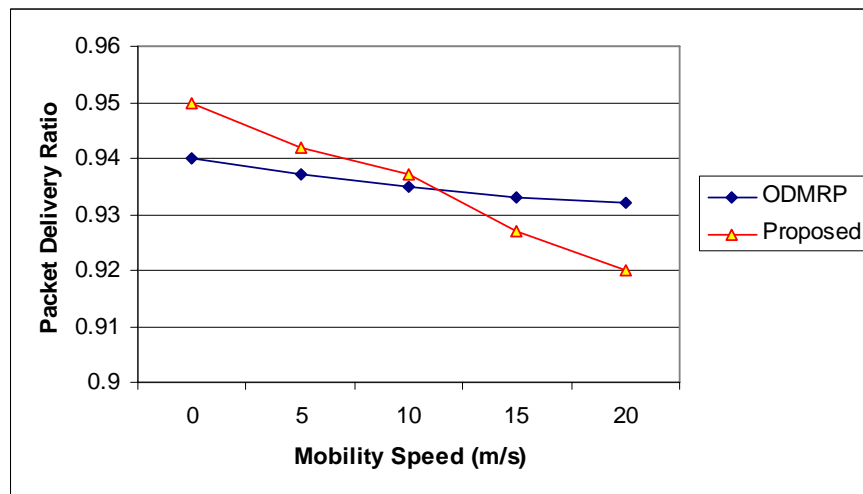
5.2.1 Packet Delivery Ratio

In this section, simulation results from proposed protocol are compared with MAODV and ODMRP in terms of packet delivery ratio. In addition, the simulation of each protocol is evaluated in static group members such that all group members remain members after joining a multicast session.

5.2.1.1 Mobility Speed



(a) Comparison of three protocols

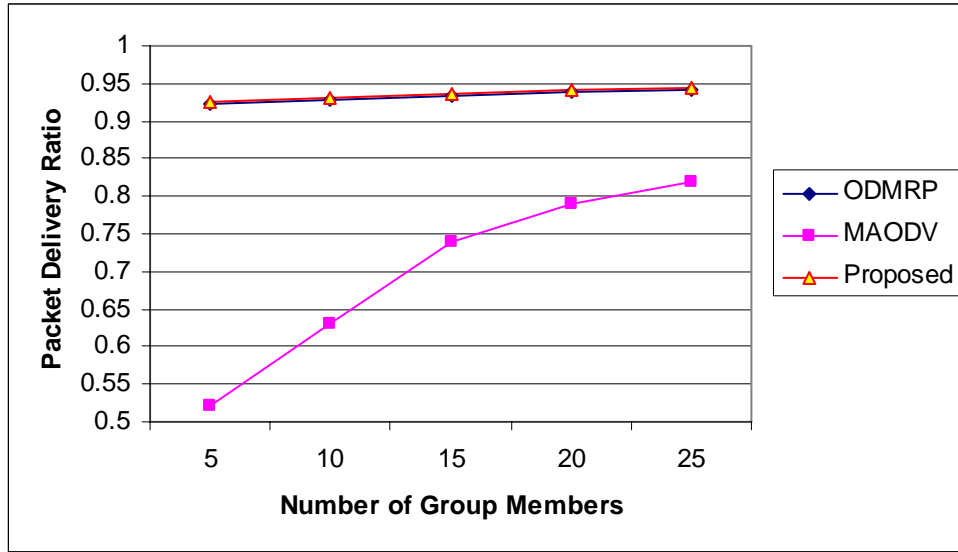


(b) Comparison of ODMRP and proposed protocol (zoom in)

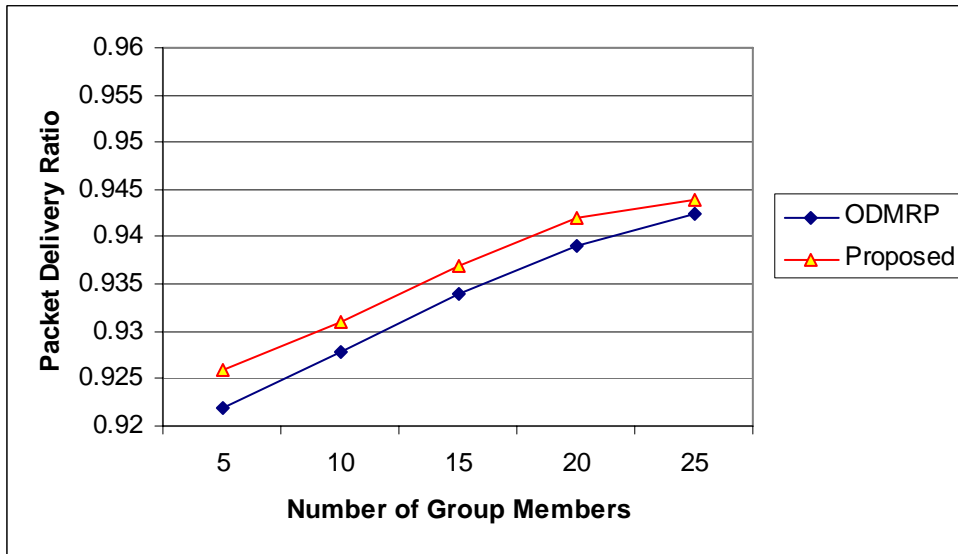
Figure 5.1 Packet delivery ratio as a function of mobility speed

Figure 5.1 illustrates the packet delivery ratio for different protocols as a function of varying movement speed for the three protocols in the 1 source and 15 receivers' scenario. Since ODMRP provides redundant routes with a mesh topology, ODMRP shows good performance even in high dynamic situations. On the other hand, the proposed protocol shows a higher packet delivery ratio to ODMRP when node mobility is not high. The packet delivery ratio of the proposed protocol is around 1.2% lower for the highly mobile scenario (mobility speed = 20m/s), but around 0.2% higher for the medium mobility scenario (mobility speed = 10m/s) and 0.5% higher for the low mobility speed (mobility speed = 5m/s). This is because in highly mobile situation, the link failure will occur more often, which makes packet delivery prediction (pdp) low and more control packets be sent to force the source to reconstruct the mesh structure; also, the proposed protocol does not use duplicated packets, all make the proposed protocol not as effective as ODMRP. However, when the mobility is medium or low, the packet delivery prediction (pdp) is higher and the old mesh structure can last longer and the data forwarding tree built on it helps reduced data overhead. MAODV shows the poorest packet delivery ratio. This is because MAODV builds a shared tree for data delivery, if a single tree link is broken, data cannot be further disseminated until a new tree is reconfigured. At a high mobility ratio, since the tree should be more frequently readjusted, packet delivery ratio rapidly decreases.

5.2.1.2 Number of Group Members



(a) Comparison of three protocols



(b) Comparison of ODMRP and proposed protocol (zoom in)

Figure 5.2 Packet delivery ratio as a function of number of group members

The source is set to 1 and average mobility speed is 10 m/s with pause time 100ms. In Figure 5.2, ODMRP is a little affected by the number of multicast numbers. As the number of group members increases, more nodes may be designated as forwarding nodes, which means more redundant routes may be established and as a result, many alternative paths remain available

even though the primary path is broken. Similar to ODMRP, the packet delivery ratio is improved as the number of group members increases for the proposed protocol. With larger group members, the data forwarding tree is well structured according to pdp values. Additionally, with larger group members, it takes a shorter time to designate a new branch as there are more alternative paths from a certain node to the destination.

5.2.2 Data Overhead

5.2.2.1 Mobility Speed

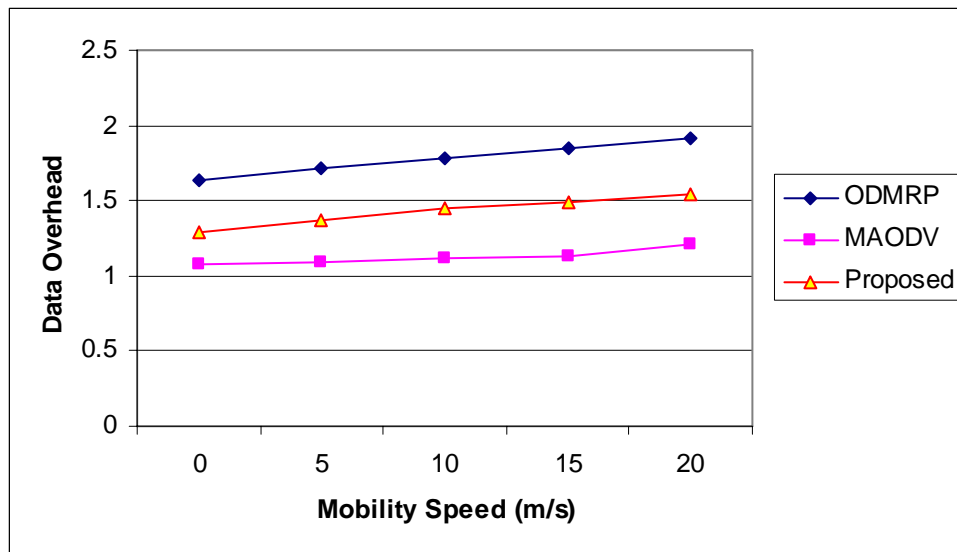


Figure 5.3 Data overhead as a function of mobility speed

Figure 5.3 shows the number of data packets transmitted per data packet delivered for the protocols in the 1 source and 15 receivers' scenario. This ratio drops to a smaller value under low mobility conditions and then increases with high mobility.

ODMRP has a relative high data overhead because there are more redundant data forwarding in the mesh structure; the proposed protocol has relative lower data overhead because it reduces the number of forwarding nodes in the structure and the data forwarding tree is actually used to forward the data. MAODV has the lowest data overhead due to its tree structure.

5.2.2.2 Number of Group Members

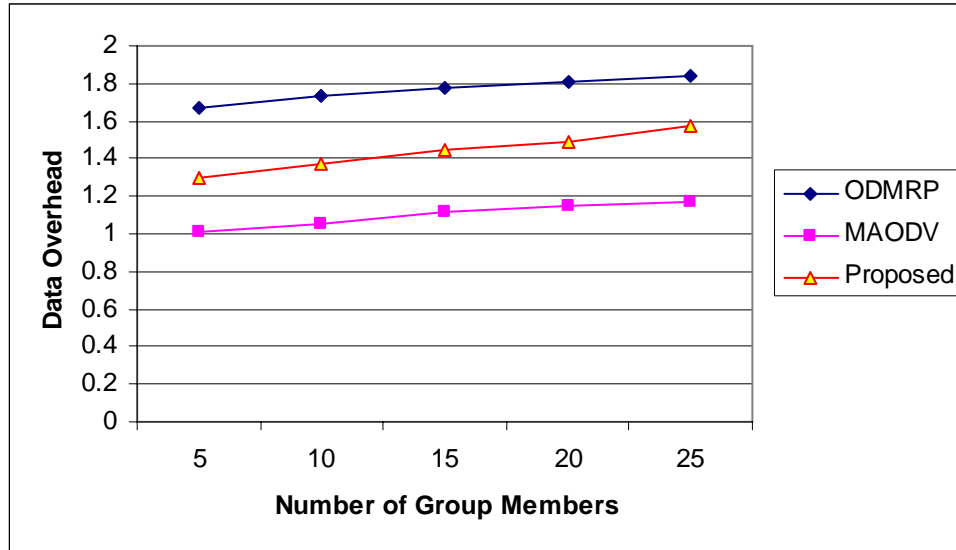


Figure 5.4 Data overhead as a function of number of group members

The source is set to 1 and average mobility speed is 10 m/s with pause time 100ms. The performance results are very similar in the scenarios for different number of receivers. In general, the higher the number of receiver, the lower the differences in the data overhead for ODMRP and proposed protocol. This is because as the number of receivers increase, so does the forwarding nodes which are really needed. Thus, the additional number of forwarding nodes used by ODMRP in a change of the best path with larger pdp that used in proposed protocol is reduced because most of the nodes are already forwarding nodes. This also holds for MAODV.

5.2.3 Control Overhead

5.2.3.1 Mobility Speed

The node moved with predefined speed. The average speeds were varied from 0m/s to 20 m/s. In these experiments, there were one multicast group where 15 nodes were receivers, and one was sender.

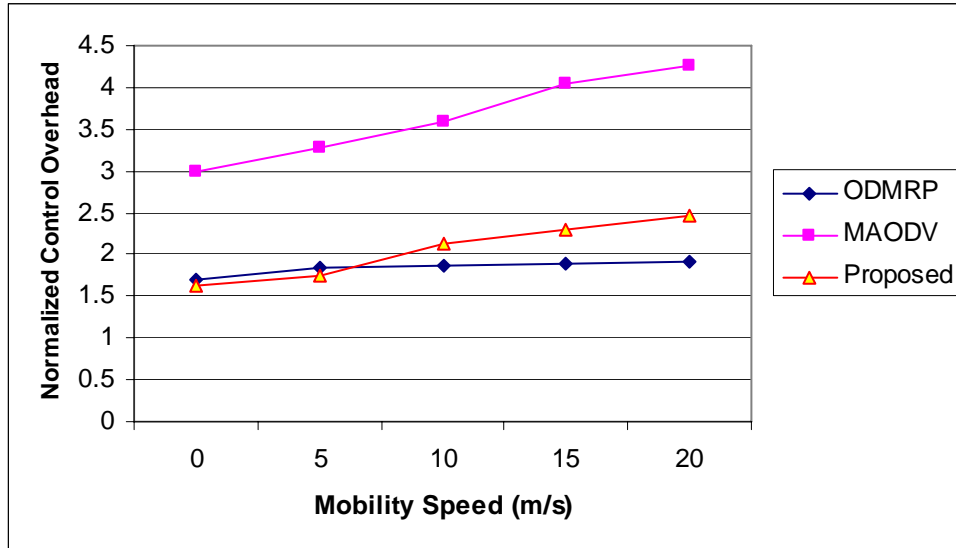


Figure 5.5 Control overhead as a function of mobility speed

From Figure 5.5, in ODMRP, control overhead remains fairly constant with node mobility because the Join-Query interval was fixed at 3 seconds. When the node mobility is low, ODMRP, with a small mesh refresh interval (three seconds), has a higher control overhead than proposed protocol, which has Hello packets that are sent relatively infrequently and routes that are stable. However, with mobility increases, ODMRP has lower control overhead than proposed routing protocol due to the fact that proposed protocol not only has Hello packets but also has Join-Query, Join-Reply and Mesh_Refresh_Request packets sent in the network due to unstable route. MAODV has the highest control overhead due to the flooding of RREQ, RREP, MACT, Hello and GRP-Hello packets in the network.

5.2.3.2 Number of Group Members

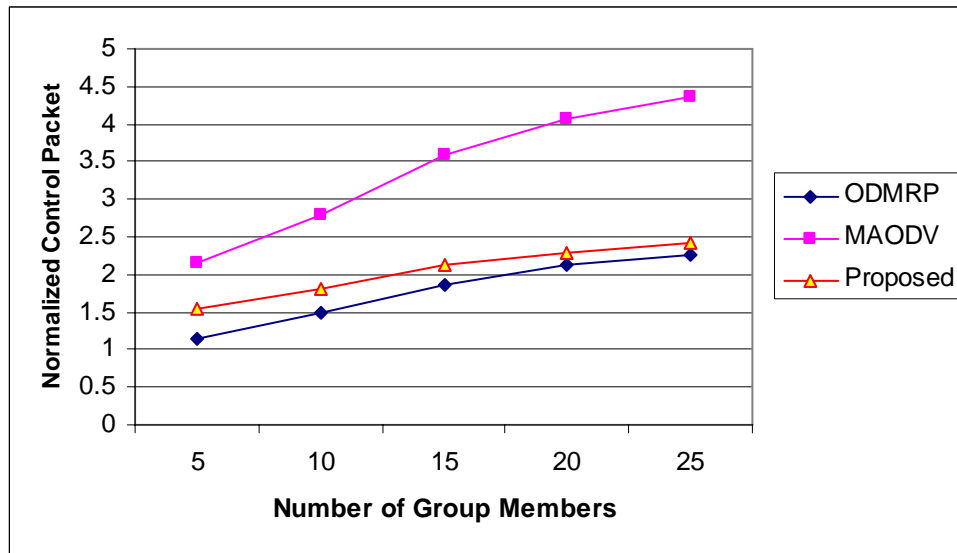


Figure 5.6 Control overhead as a function of number of group members

The number of multicast group members is varied to investigate the scalability of the protocol. The source is set to 1 and average mobility speed is 10 m/s with pause time 100ms. Figure 5.6 demonstrates that as the number of group members increase, control overhead for all three protocols will increase. For MAODV, the increase is the most since more RREQ, RREP, MACT, Hello and GRP-Hello will be sent out. The proposed protocol will also have more control packet because not only the control packets that ODMRP uses will increase, it will have other control packets like Hello packets.

5.2.4 Average End-to-End Delay

5.2.4.1 Mobility Speed

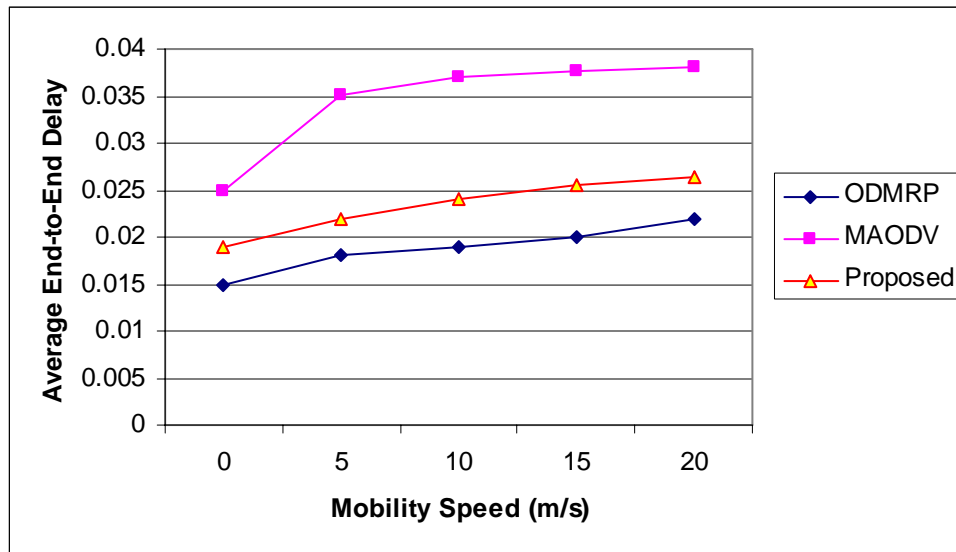


Figure 5.7 Average end-to-end delay as a function of mobility speed

Figure 5.7 shows average end-to-end delay for MAODV, ODMRP and proposed protocol under different node mobility. It shows that ODMRP has smaller average end-to-end delay than MAODV at any node speed. The proposed protocol has the average end-to-end delay that is longer than ODMRP but shorter than MAODV. This is due to that fact that it has less packet collision and retransmission due to less control packet than MAODV, and it does not select shortest path as ODMRP does.

5.2.4.2 Number of Group Members

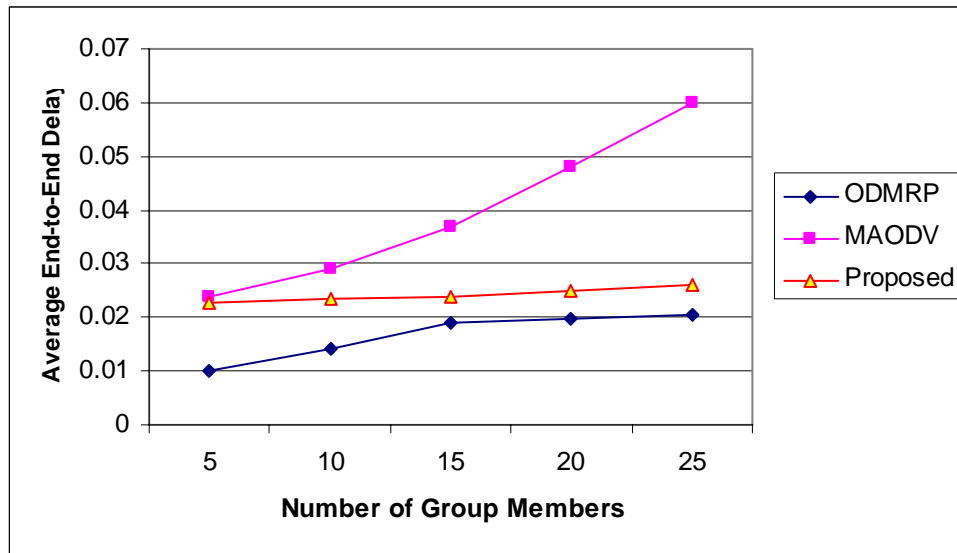


Figure 5.8 Average end-to-end delay as a function of number of group members

Figure 5.8 shows average end-to-end delay for MAODV, ODMRP and proposed protocol for different number of group members. The source is set to 1 and average mobility speed is 10m/s with pause time 100ms. End-to-end delay is highest for MAODV due to the longer paths that data packets have to follow within the shared tree and due to the higher network load caused by the high number of control and data packet transmissions. Network load translates into a busier wireless medium, which causes nodes to have to wait longer before forwarding each packet. ODMRP's end-to-end delay is lowest because, due to its frequent state discovery floods, it uses the shortest forwarding paths among the three protocols. The proposed protocol has average end-to-end delay in between due to its data forwarding tree built upon mesh structure while the paths should satisfy the path selection criteria stated in Chapter 4.

5.3 Summary

In this chapter, simulation and comparisons of three protocols, ODMRP, MAODV and proposed protocol, are discussed. From the simulation result, the packet delivery ratio as a function of the node mobility speed is depicted for three protocols in the 1 source and 15

receivers' scenario. Both ODMRP and proposed protocol deliver over 92% of the traffic even in the highly mobile scenarios. When the node mobility is low, proposed protocol has higher packet delivery ratio (PDR) than ODMRP and MAODV. As the node mobility increases, the original ODMRP version delivers around a 1.2% more of data packets than the proposed protocol. However, to achieve this packet delivery ratio, the original ODMRP version requires use of forwarding nodes to form redundant routes. In addition, the proposed protocol has a lower data overhead. This is achieved by reducing the number of forwarding nodes, the data overhead is also reduced.

The performance results show the difference in the scenarios for 1 source and different number of receivers. In general, for ODMRP and proposed protocol, the higher the number of receivers, the lower the differences in packet delivery ratio and also the lower the differences in data overhead. This is because as the number of receivers increase, so does the number of forwarding nodes which are really needed. Thus, the additional number of forwarding nodes used by ODMRP in a change of the shortest path tree is reduced because most of the new nodes are already forwarding nodes.

Chapter 6

Quality of Service (QoS) Multicast Routing in MANET

The evolution of the multimedia technology and the commercial interest of companies to reach widely civilian applications made Quality of Service (QoS) in mobile ad hoc network (MANET) an area of interest. Quality of Service (QoS) routing requires not only finding a route from a source to a destination, but also the route must satisfy the end-to-end QoS requirement, often given in terms of bandwidth or delay. Future wireless networks will carry diverse multimedia applications such as voice, video and data. In order to provide quality delivery to delay sensitive applications, it is imperative that MANETs provide QoS support in term of bandwidth and delay. QoS provision in MANET is a challenging task since in addition to obeying QoS constraints, we must account also for a dynamic topology and shared wireless medium.

Aside from the problems that exist for QoS in wire-based networks, MANETs have new features that bring more challenges for QoS. These features are:

- a) Dynamic topology of the MANET [15]. Nodes are changing location continually, connecting and disconnecting from the network, making connections unreliable;
- b) Bandwidth constrains [16]. A MANET usually has poor bandwidth resources;
- c) Limited processing and storing capabilities of nodes [62].

Due to these features, we have to keep protocol complexity as low as possible since this may also lead to other problems like excessive power consumption.

QoS in MANET should provide a set of parameters in order to adapt application to the quality of network while routing through the network. Providing strong guarantees in MANET is really hard due to a number of factors such as varying link quality, capacity and mobility. Therefore, it is essential for routing protocols to do their best to control the congestion and manage resources for real-time multicast service in MANETs.

For multicast applications, efficient support of multicast communications is essential to provide service like group audio and video conferencing, dissemination of data to a set of receivers or collaborating of a group of users. With the use of multicast, multimedia applications require Quality of Service (QoS) guarantees. In this chapter, the protocol proposed in Chapter 4 was modified to achieve QoS to accommodate the real-time application requirements when routing the packets through the network.

6.1 Overview

6.1.1 QoS Support in MANETs

There exist several QoS architectures proposed for MANETs. The main idea is to maintain the fewest QoS states as possible because the network topology changes due to mobility make it hard to maintain reservation states over time without high control overhead expenditure [52].

Designed to support adaptive services as a primary goal, INSIGNIA [53] is a QoS model that consists of an in-band signaling protocol with support for adaptive reservation-based service in ad-hoc networks. Adaptive services provide minimum bandwidth assurances to real-time voice and video flows and data, and it allows enhanced levels of service to be delivered when resources become available. It is designed to adapt user sessions to the available level of service without explicit signaling between source-destination pairs. It comprises the following architectural components:

- 1) In-band signaling. It is used to establish, restore, adapt and tear down adaptive services between source-destination pairs.
- 2) Admission control. It is responsible for allocating bandwidth to flows based on the maximum/minimum bandwidth requested.
- 3) Routing. It is used to dynamically track changes in ad hoc network topology, making the routing table visible to the node's packet forwarding engine.

- 4) Packet scheduling. It is used to respond to location-dependent channel conditions when scheduling packets in wireless networks.

In-band information concerning the required resources for the flow is used by intermediate nodes to establish, maintain and restore per-flow soft-state reservations. Admission control is performed hop by hop, and the destination host informs the source node of the result of the reservation using a QoS report mechanism. Reservations are maintained as long as packets associated with a particular flow are periodically received at intermediate nodes to refresh timers. INSIGNIA copes with re-routing by triggering the admission control and resource reservation through the new path, while reservation state along the old path times out.

FQMM [54] was developed for assuring a certain level of service differentiation in ad-hoc networks. It is a hybrid Integrated Services (IntServ)/Differentiated Services (DiffServ) model and supports hybrid per-flow/per-class provisioning. Like DiffServ, FQMM has service differentiation for different traffic class. Due to the bandwidth limitation and other constraints in MANET [16], only a small portion of traffic will be provided with per-flow granularity (IntServ). In other words, the model proposes that highest priority is assigned per flow provisioning and other priority classes are given per-class provisioning. It assumes that not all packets in the network are actually seeking for highest priority.

FQMM [54] defines three kinds of nodes, ingress nodes, interior nodes, and egress nodes, as in DiffServ.

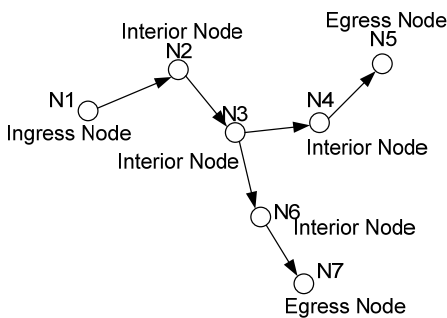


Figure 6.1 Nodes in FQMM

Source nodes are considered as ingress nodes, which use classification, marking, policing and shaping. Traffic conditioning is carried out according to the traffic profile. A relative and adaptive differentiation traffic profile was proposed for FQMM [54], which uses relative percentage of the effective link capacity to keep the differentiation between classes consistent and adaptive to network dynamics. It is used to keep consistent differentiation between sessions (flows/aggregates). Intermediate nodes are nodes that forward data to other nodes and only apply shaping. Egress nodes are destination nodes in the network.

6.1.2 QoS-enabled Multicasting in MANET

QoS-enabled routing plays an important role for providing QoS in wireless ad hoc networks. The goals of QoS routing are in general as follows: selecting routes with satisfied QoS requirement(s), and achieving global efficiency in resource utilization.

Based on application requirements, QoS can be divided into two basic types [6]: resource reservation and prioritization. Some key design considerations in QoS routing are [1]: a) Scalability with respect to network size, communication and storage overhead; b) Guarantee of an initial QoS due to network dynamics caused by node mobility; c) Effectiveness to absorb routing information inaccuracy; d) Preventing starvation of best effort traffic wherein two traffic types coexist.

To provide Quality of Service (QoS) in Multicast MANETs, a tight integration of resource management with multicast protocol seems to be beneficial in order to avoid overhead and minimize network state. Additional signaling packets for reservation protocol should be avoided as this contributes to network congestion, especially in high mobility and traffic-intensive scenarios.

The QoS multicast routing protocol proposed here is aimed at achieving a soft-QoS routing in Mobile Ad Hoc Networks. Soft-QoS support means there exists transient time period when the required QoS is not guaranteed due to factors such as packet loss, congestion, path

breaking, network partition and mobility. For this protocol, it will take the nodes some time at the beginning of mesh construction to build the routes that satisfy the bandwidth requirements later.

In order to handle the bandwidth requirement of the multicast applications, each node is assumed to be able to monitor the available resources (e.g., delay and cost) on each of its outgoing links and decide resource availability. The nodes in the network will periodically broadcast mesh construction packets and measure the bandwidth of outgoing links along the path that the packet has traversed. The information is used to create and maintain the underlying mesh structure. By doing so, the data forwarding paths that satisfy the required bandwidth will be obtained later.

6.1.3 Two Potential Bandwidth-violation Problems

When determining bandwidth-satisfied routes for bandwidth-required requests, [13] pointed out that there are two bandwidth-violation problems in MANET QoS routing/multicasting protocols that would occur.

When a new flow with bandwidth requirement is initiated, a control packet from the source is flooded to determine a route that satisfies bandwidth requirement. Each host may be determined as a forwarder for the new flow if the bandwidth increment will not cause bandwidth violation of it as well as its neighboring hosts. However, bandwidth violation is still possible for its neighbors because it fails to take into account the bandwidth consumption of those hosts which are two hops distance away from it. This introduces a new problem in MANETs, which is referred to as the **Hidden Route Problem (HRP)** [13].

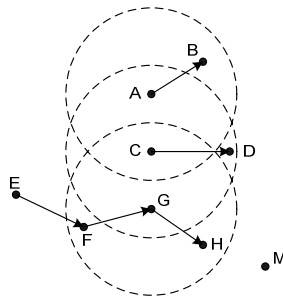


Figure 6.2 An example of HRP problem

In Figure 6.2, two routes A->B and C->D are established and one route E->->M is being constructed. Here “->->” denotes a route of one or more hops. Suppose the capacity of each host is constant, say 11 units, and the bandwidth requirements of A->B, C->D and E->->M are 7, 2 and 3 units respectively. When host G checks to see if it can be an intermediate node in route E->->M, the available capacity of G will be computed. In this case, Node G can be constructed as one of the forwarding node of E->->M. However, the establishment of G as intermediate node increases the bandwidth consumption in the radio coverage range of Node C. The ongoing transmissions from A->B and G->H use $7 + 3 = 10$ units, which makes the total bandwidth needed for Node C be $10 + 2 = 12$ units, and it violates the capacity of Node C. The reason for this violation is that Node G was not aware of an ongoing transmission from A->B when computing remaining capacity in its radio coverage.

Another bandwidth-violation problem, called **Hidden Multicast Route Problem (HMRP)** [13], would mislead the bandwidth reservation for QoS multicast applications. In multicast applications, multiple routes from a server to all clients were determined concurrently, and the bandwidth reservations were activated only when there is real data flowing at the hosts. There is possibility that a host and some of its neighbors might misinterpret to have enough remaining bandwidth at the routing stage and nodes along different routes from server to distinct clients were determined as the forwarders. However, the bandwidth reservation to these hosts will likely violate node's/nodes' bandwidth capacity. On the other hand, they would also trigger HRP.

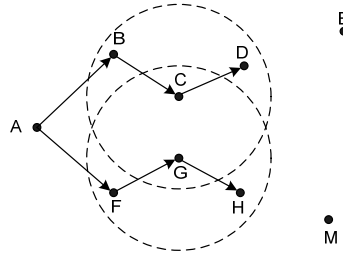


Figure 6.3 An example of HMRP problem

In Figure 6.3, multicast routes from A to E (A->->E) and M (A->->M) are being constructed. Suppose the bandwidth requirement for each of A->->E and A->->M is 3 units. The total bandwidth requirement for Node C is 9 units (B->C, C->D and D->E) if only route A->->E is considered. In this case, Node C has enough capacity, and is constructed as one forwarder for A->->E. When there is no real data flow from A->-> M, the remaining capacity for Node G is still 11 units and Node G can be constructed as one forwarder of A->->M at the same time. However, when there is data flow from A->->E and A->->M, the bandwidth consumption in the radio coverage of Node C will be 12 units, in which there are four forwarders (B, C, D and G) and each forwarder consumes 3 units of bandwidth. That violates the capacity of Node C. The same violation also happens in the radio coverage of Node G.

Both **Hidden Route Problem (HRP)** and **Hidden Multicast Route Problem (HMRP)** may have high possibilities to occur and affect the performance when the network traffic is heavy.

6.2 MAC Layer Modification

Due to network mobility, distributed MAC protocol, 802.11 DCF [59], is used. In my proposed QoS-enabled multicast routing protocol, the ACK will not be used given it is not necessary for multicast traffic. If the MAC-layer backoff increases, the rate at which best effort traffic enters the MAC-layer will decrease at each forwarding nodes. Otherwise, the rate will increase according to Additive Increase Multiplicative Decrease (AIMD) control algorithm [5].

6.3 QoS Operation Modification to On-Demand Hybrid Multicast Routing Protocol

The QoS-enabled multicast routing protocol works in several ways to allow QoS routing. It consists of neighbor discovery and maintenance, mesh construction, packet routing mechanism, bandwidth reservation mechanism and efficient resource releasing upon route adjustment. Bandwidth calculation is used to avoid bandwidth violation and help node decide whether the required bandwidth for a real-time flow should be reserved.

6.3.1 Neighborhood Maintenance

Neighborhood information is very important in the proposed routing protocol since it provides the information about mobility, traffic and local topology. This information is critical for QoS violation avoidance, detection and recovery.

To maintain the neighborhood information, every node in the network is required to periodically send out a “Hello” packet, announcing its existence and traffic information to its neighbors. The “Hello” packet can be sent at a default rate with time to live (TTL) set to 1. Every node in the network receives the “Hello” packets from its neighbors and maintains a neighbor list which includes all its neighbors with their corresponding self-traffic.

6.3.2 QoS Mesh Construction Considering Bandwidth Violation Problems

If the traffic is real-time traffic and it needs QoS multicast routing, the QoS-enabled multicast routing protocol proposed in this chapter will be used. To identify whether the real-time traffic is admitted or rejected, the source node will probe the network to figure out if there's enough bandwidth to accommodate certain traffic flow/flows. This is mainly done during mesh-construction and traffic mark phase. The source will set the Type of Service (ToS) bit in the IP-header of the real-time traffic flow, and probe the network using mesh construction procedure if no mesh has been built.

The mesh construction procedure for the modified routing protocol is on-demand, and is constituted of three phases: request phase, reply phase and traffic mark phase.

The request phase and reply phase for the mesh construction are illustrated in Figure 6.4.

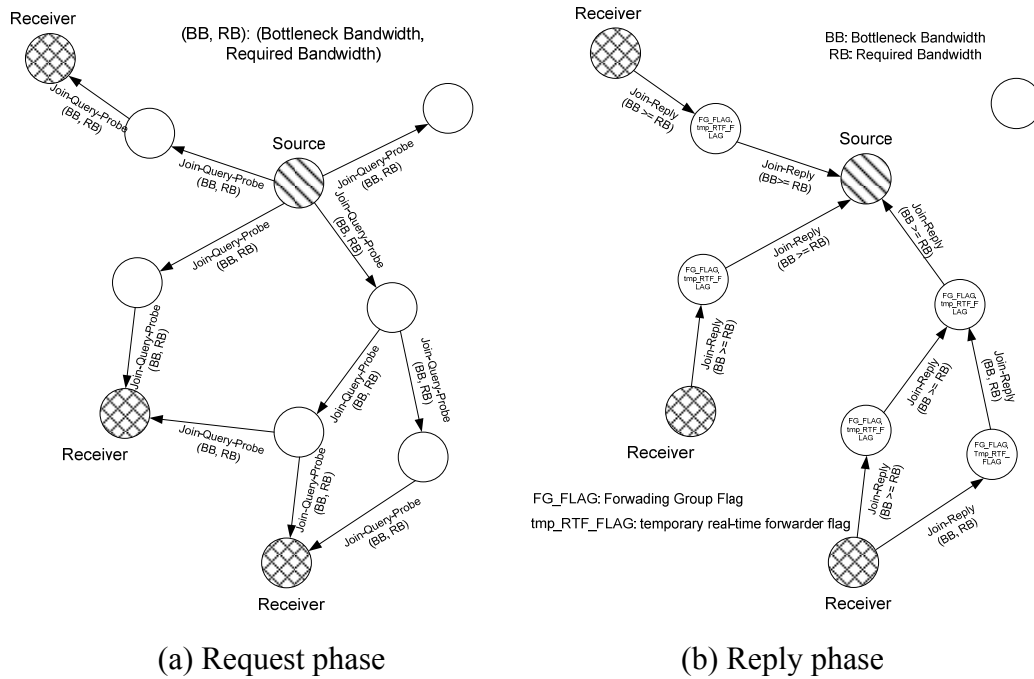


Figure 6.4 Mesh construction: request phase and reply phase

Request phase: When a multicast source has real-time traffic to send but no route or group membership is known, it floods a member advertising packet, called “*Join-Query-Probe*”, which is the “*Join-Query*” (4.3) message with a probing request piggybacked. This probing request contains a bottleneck bandwidth (BB) and a required bandwidth (RB) fields. The “*Join-Query-Probe*” message is periodically broadcasted to the entire network to probe the resources, refresh the membership information and update route. When a node receives a “*Join-Query-Probe*” packet, it stores the *source address* and the *unique identifier* of the packet to its “Message Cache” to detect duplicate. The *upstream node address* is inserted or updated as the next node for the source node in its “Routing Table”. If the “*Join-Query-Probe*” packet is not a duplicate and Time-To-Live (TTL) value is greater than zero, the intermediate node sets pointer to its upstream nodes and modifies the probing request information: if the local bandwidth availability at the given node is less than the bottleneck bandwidth value (BB filed) in the “*Join-Query-Probe*” packet, the bottleneck bandwidth

(BB) field in the “Join-Query-Probe” will be updated. Other appropriate fields, like hop count and previous hop IP address, are updated and the packet is rebroadcast.

Reply phase: When a “Join-Query-Probe” packet reaches the multicast receiver, value in bottleneck bandwidth (BB) field indicates the bottleneck bandwidth along the path. The receiver waits for a small amount of time ($T_{collect}$) to collect enough “Join-Query-Probe” packets that reach the receiver through other paths in the network. The receiver will evaluate if the bottleneck bandwidth (BB) field value is greater than required bandwidth (RB), along with consideration of the link stability information collected along the path. If so, receiver creates “Join-Reply” packet, with “Probe-Response” information like bottleneck bandwidth (BB) and required bandwidth (RB) fields piggybacked, and broadcasts it to its neighbors. The receiver also sets a temporary real-time forwarder flag (“tmp_RTF_FLAG”) bit in its data structure for the given multicast group. The “Join-Reply” is relayed by the intermediate nodes all the way to the source following the pointer set when “Join-Query-Probe” was propagated. When an intermediate node receives a “Join-Reply”, it checks the fields in the packet to see if the next node address of one of the entries matches its own address. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the “FG_FLAG” (Forwarding Group Flag) in its data structure. It also sets its “tmp_RTF_FLAG” bit for the given multicast group and the real-time traffic flow if node’s bottleneck bandwidth (BB) value is larger than required bandwidth (RB). The intermediate node calculates the *Packet_Delivery_Prediction*(PDP) based on the *mobility* and *remaining battery*. It then broadcasts its own “Join-Reply” built upon matched entries with its *PDP* information attached. The next node address fields are filled in by extracting the information from its routing table. The uplink nodes will record the PDP information received via “Join-Reply” sent from every downlink node, and record *Remain_Connected_Predict_Time* (RCPT) of this specific {uplink node, downlink node} pair, which is also obtained from mobility prediction (4.2.3). This way, the “Join-Reply” packet is propagated by each forwarding group member until it reaches the multicast source via the selected path. All uplink nodes will have information of their downlink nodes’ PDP and RCPT for the link

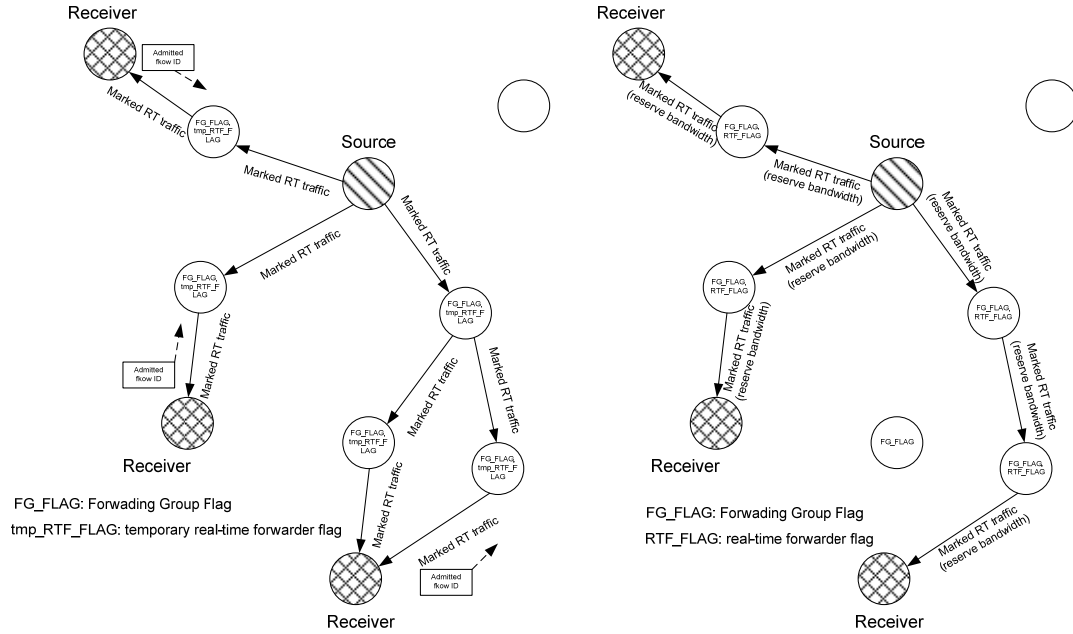
between them. Every intermediate node checks the bottleneck bandwidth of itself, if its bandwidth is greater than real-time traffic's required bandwidth, it will record the required bandwidth and flow ID in its node structure so that it can reserve the bandwidth for the flow in traffic mark phase when the real-time forwarding flag (RTF_FLAG) is set.

Traffic mark phase: For a predefined QoS convergence time $t_{converge}$ duration, real-time traffic is sent via this mesh structure, with only the nodes with both "tmp_RTF_FLAG" bit and "FG_FLAG" bit set forwarding the traffic. During this $t_{converge}$ time, the real-time traffic is marked so that when a node received the duplicated traffic from neighboring node, it will checking equations introduced in 6.4.1 to decide if it has any bandwidth violation problem like **Hidden Route Problem (HRP)** or **Hidden Multicast Route Problem (HMRP)**.

If the node with both "tmp_RTF_FLAG" and "FG_FLAG" set satisfies the real-time QoS bandwidth requirement and doesn't violate bandwidth consumption problem like **Hidden Route Problem (HRP)** or **Hidden Multicast Route Problem (HMRP)**, real-time forwarding flag, "RTF_FLAG" bit, will be set in node's data structure. At the end of QoS convergence time $t_{converge}$, some nodes will have "RTF_FLAG" set, while nodes that violate bandwidth consumption will not have "RTF_FLAG" set even they have set "tmp_RTF_FLAG" previously during Reply phase. The receiver of the real-time traffic flow will send a small packet back to sender to inform sender that the traffic flow with specific flow ID as admitted traffic. This helps sender to set "admitted" bit in the packets belonging to the traffic flow.

After this traffic mark phase and before the next "Join-Query-Probe" packets are triggered, real-time traffic will be forwarded by the nodes with both "RTF_FLAG" and "FG_FLAG" bit set. Bandwidth reservation is made when the real-time flow arrives at the node with "RTF_FLAG" and "FG_FLAG" set. There is also a time interval $T_{reserve}$ used to make bandwidth reservation for a real-time traffic flow effective for a certain period of time.

Details of bandwidth reservation and releasing are presented in 6.4.2 and 6.4.3. Figure 6.5 shows the procedure to set up real-time forwarder group and reserve bandwidth for the requested real-time flow in traffic mark phase.



(a) During initial $t_{converge}$ time (b) After $t_{converge}$ time

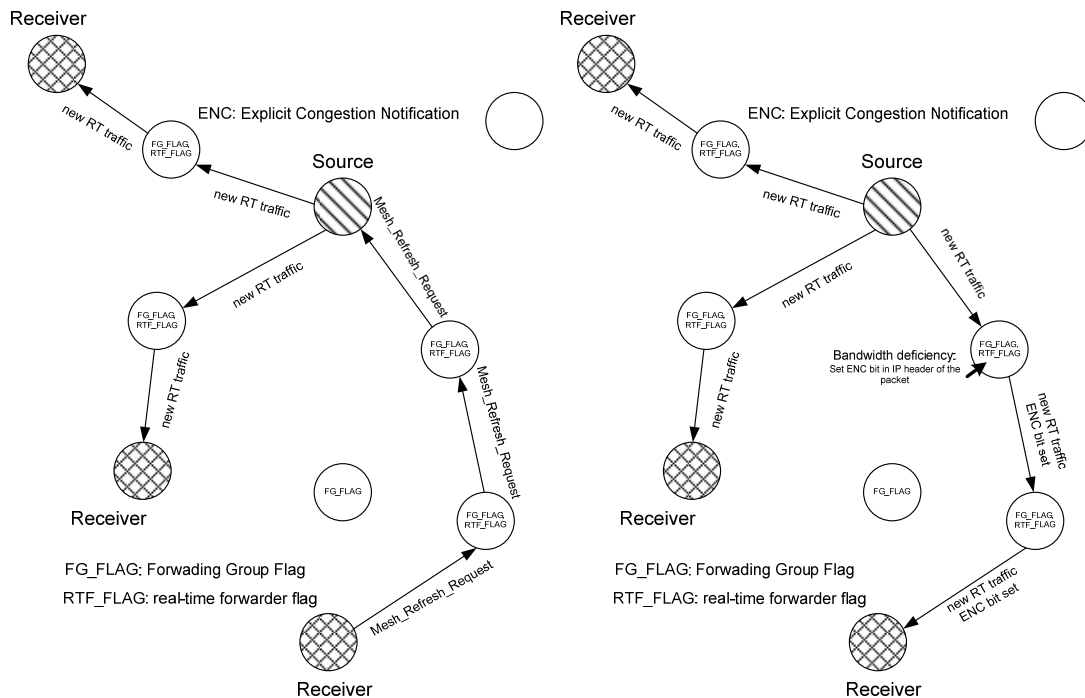
Figure 6.5 Traffic mark phase

The scheme used to select forwarding nodes for real-time traffic builds the mesh structure aimed at satisfying bandwidth requirements and avoiding potential bandwidth consumption violation. Other information each forwarding group node may maintain includes the multicast group IP address and/or individual receiver IP address.

6.3.3 Traffic/Data Forwarding

The traffic/data forwarding scheme is based on whether the traffic is real-time traffic or non real-time traffic. If the network can accommodate the real-time traffic, required bandwidth for the real-time traffic will be reserved for this specific traffic flow at each intermediate node, with unique flow ID as identifier. Source node will send this admitted real-time traffic via MAC-layer broadcast. The tree is built based on bandwidth and $T_{reserve}$ since $T_{reserve}$ represents the stability of the link that can accommodate the real-time traffic flow. Before an

intermediate forwarding node re-broadcasts the packets, the classifier of that node will check if “admitted” bit for the real-time traffic flow is set. If it is set and the node has enough bandwidth, the packets are directly passed to the MAC layer for re-broadcasting. In this way, there will be less effective forwarding nodes in the network when the traffic is heavy, thus reduce the probability that there will be potential bandwidth violation issues like **Hidden Route Problem (HRP)** or **Hidden Multicast Route Problem (HMRP)**. For traffic that is non real-time traffic (best effort, BE) in this QoS-enabled network, the data packets injected into network will be regulated by the rate shaper, which adjusts the rate of best effort traffic after setting aside bandwidth for real-time traffic. MAC layer back-off delay of 802.11b is used as feedback. If the back-off increases, the rate at which best effort (BE) traffic enters the MAC-layer decreases; otherwise it increases according to Additive Increase Multiplicative Decrease (AIMD) control algorithm. The data forwarding scheme introduced in 4.2.5 is used to select the route.



(a) Request mesh rebuilt (b) New RT traffic flow enters: bandwidth deficiency

Figure 6.6 New RT traffic flow enters the network during traffic forwarding phase

Figure 6.6 illustrates two scenarios when a new real-time traffic enters the network. When sender is going to send a new real-time traffic flow, it will start building a new structure for the new flow instead of trying to reuse the old structure for already admitted flows, which is illustrated in Figure 6.6 (a). Every intermediate node will compare the incoming new real-time traffic's bandwidth requirement with the available bandwidth at the node. If there is enough bandwidth to accommodate the new real-time traffic flow, it will be sent to MAC directly; otherwise, the delivery for this real-time traffic will be rejected. If the traffic forwarding routes no longer have enough bandwidth due to mobility or battery shortage, "Explicit Congestion Notification" (ECN) bit will be set in the IP header of the packets of the flow, which is illustrated in Figure 6.6 (b). The receivers monitor the ENC bits and will notify the sender to rebuild the mesh and reserve the resource according to new bandwidth requirement by sending "Mesh_Refresh_Request". The source node waits for a random amount of time and initial reestablishment of the mesh. This can avoid flash-crowd conditions where a number of sources simultaneously initiate the procedure at the same time [4].

6.3.4 Mesh Maintenance

Mesh maintenance is achieved by periodically broadcast "*Join-Query-Probe*" packets. The intermediate nodes in the network then measure resource availability and update bottleneck bandwidth (BB) and/or a required bandwidth (RB) fields in the "*Join-Query-Probe*" message that are used to create the mesh.

6.4 Bandwidth Control

6.4.1 Bandwidth Violation Avoidance

During $t_{converge}$ time in traffic mark phase, the real-time traffic is marked so that when a node received the duplicated traffic from neighboring node, it will use the following equations to check and decide if it has any bandwidth violation problem like **Hidden Route Problem (HRP)** or **Hidden Multicast Route Problem (HMRP)** [13]:

$(B_after_i) = (B_ongoing_i) + (B_required_i)$; for each forwarding node in the mesh
 $\sum((d_{i,j}) (B_after_i)) + \Delta_{bw} \leq B_max_i$; for each forwarding node in the mesh
 $B_ongoing_i > 0$

Here,

$d_{i,j}$: a binary integer $x_i=1$ ($x_i=0$) to denote that forwarding node v_i is (is not) a neighbor of forwarding node v_j . Assume that the transmission range for all nodes are the same and $d_{i,j}=d_{j,i}$ for all pairs of v_i and v_j . $d_{i,i}=1$ since the very node v_i is transmitting traffic.

B_max_i : maximum transmission bandwidth for node i .

$B_ongoing_i$: the sum of bandwidth transmitted from node i for the ongoing flows.

$B_required$: the bandwidth requirement for the requesting multicast group.

B_after_i : the sum of consumed bandwidth transmitted for node i for the ongoing flows and the requesting multicast group.

Δ_{bw} : fluctuation bandwidth that is used in case bandwidth fluctuation in Mobile Ad Hoc Network may affect the forwarding node decision for real-time traffic.

Detailed bandwidth calculation on available bandwidth and added consumed bandwidth introduced by admitting a new real-time flow for a node is presented in 6.4.2 and 6.4.3.

6.4.2 Node's Bandwidth Availability Estimation

In wireless network, the radio channel of each node is shared with all its neighbors. Because of the shared medium, a node can successfully use the channel only when all its neighbors do not transmit and receive packets at the same time. To calculate bandwidth available and consumed, the following assumptions are made: the wireless channel is half-duplex and all the nodes have identical data rate and transmission range. When calculating the available bandwidth of a node, not only the traffic the node is currently transmitting should be counted, the traffic that is carried on by the neighboring nodes and sharing in the same radio channel should also be counted. The available bandwidth for a certain node can be calculated using

the formula introduced here. The traffic that contributes to the consumption of a node's bandwidth can be of the following types:

- a) Traffic through node I, $B_{self}(I)$. It is the total traffic that is transmitted or received by node I. This traffic is carried on between node I and its neighbors.
- b) Neighboring traffic, $B_{neighbor}(I)$. It is the total traffic that is carried on between node I's neighbors. Node I does not directly involve in the transmission and receiving of the traffic. The nodes involving in the traffic are all node I's neighboring nodes.
- c) Edge traffic, $B_{edge}(I)$. The traffic that is carried on between node I's neighboring nodes and the nodes that are outside node I's range.

Taking the above traffic into consideration, the total amount of traffic load that exists in the channel is:

$$B_{load}(I) = B_{self}(I) + B_{neighbor}(I) + B_{edge}(I) \quad (6-1)$$

Figure 6.7 shows an example of the traffic load for a node I. Node I has four neighbors, node O, node P, node Q, node R, and node S is outside of node I's range.

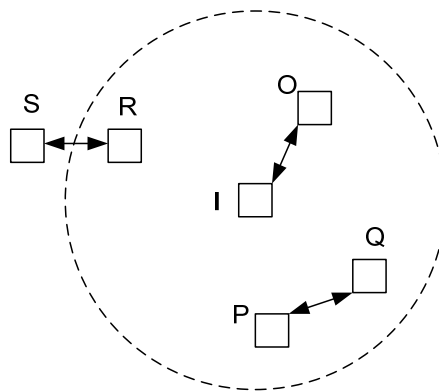


Figure 6.7 Traffic load of Node I

The traffic between node I and node O, denoted as $Traffic_{I-O}$, represents $B_{self}(I)$; Traffic between node P and node Q, denoted as $Traffic_{P-Q}$, represents $B_{neighbor}(I)$; Traffic between node R and node S, denoted as $Traffic_{R-S}$, represents $B_{edge}(I)$. Total load for node I then is:

$$B_{load}(I) = Traffic_{I-O} + Traffic_{P-Q} + Traffic_{R-S} \quad (6-2)$$

By taking edge traffic $Traffic_{R-S}$ into consideration, the hidden terminal problem between node I, node R and node S can be avoided. $B_{load}(I)$ imposes an upper bound of the existing traffic in node I's channel since neighboring traffic and edge traffic can be transmitted at the same time without interference.

To estimate the neighboring traffic, node I has to listen to the channel, however, it may underestimate edge traffic like $Traffic_{R-S}$. Another way to observe the traffic load in node I's channel can be stated as follows:

$$B_{load}(I) = B_{self}(O) + B_{self}(P) + B_{self}(Q) + B_{self}(R) - Traffic_{P-Q} \quad (6-3)$$

Node O, node P, node Q and node R are node I's neighbors. Assume $B_{total_nb_self}(I)$ as the sum of node I's neighbors' self traffic. Equation (6-3) can be written as:

$$B_{load}(I) = B_{total_nb_self}(I) - Traffic_{P-Q} \quad (6-4)$$

$$\text{Here, } B_{total_nb_self}(I) = B_{self}(O) + B_{self}(P) + B_{self}(Q) + B_{self}(R) \quad (6-5)$$

$$\text{or } B_{total_nb_self}(I) = \sum_{J \in Neighbor(I)} B_{self}(J) \quad (6-6)$$

From (6-1), (6-2) and (6-6), we have

$$B_{total_nb_self}(I) = B_{load}(I) + B_{neighbor}(I) \quad (6-7)$$

It can be observed that the sum of node I's neighbors' self traffic bandwidth $B_{total_nb_self}(I)$ is greater than traffic load in node I's channel $B_{load}(I)$ by the amount of neighboring traffic, $B_{neighbor}(I)$. For the traffic shown in Figure 6-3, $B_{neighbor}(I) = Traffic_{P-Q}$. $B_{total_nb_self}(I)$ overestimate $B_{load}(I)$, and it helps establish the lower bound of the available bandwidth of node I when the following formula is used:

$$B_{available}(I) = B_{max} - B_{total_nb_self}(I)$$

$$\text{or } B_{available}(I) = B_{max} - \sum_{J \in Neighbor(I)} B_{self}(J) \quad (6-8)$$

Here, B_{max} is the maximum bandwidth of node. From (6-7) and (6-8), it shows that the more interference traffic in the channel, the more conservative the available bandwidth will be

estimated. In the proposed protocol, to make sure that every node in the network keeps updated available bandwidth in the node's data structure, "Hello" packet is sent periodically to inform the neighboring nodes traffic $B_{self}(I)$ that is reserved at the node. The relation between $B_{self}(I)$ and the reserved bandwidth at node I is defined as:

$$B_{self}(I) = \sum_k B_I(k) \quad (6-9)$$

Here, $B_I(k)$ is the reserved bandwidth for a real-time traffic k at node I.

6.4.3 Flow's Bandwidth Consumption Estimation

To determine whether there is enough bandwidth available for a new flow, we need to know the available link capacity and the bandwidth to be consumed by the requesting flow. In 6.4.2, the available link capacity of a certain node I was calculated. To determine whether there is enough bandwidth available for a new flow, the bandwidth to be consumed by the requesting flow will be calculated here. Each forwarding node in the mesh is supposed to monitor the on-going traffic in the neighborhood so that it has enough bandwidth (required bandwidth, RB) for the admitted traffic. Since the wireless medium is shared by the nodes, the bandwidth that will be consumed by requesting real-time flow in the intermediate node is different from the actually requesting required bandwidth due to the interference of the relaying neighbors.

If there is a new real-time traffic flow k with required bandwidth $B_{required}(k)$, the bandwidth to be reserved at node I along the path can be:

$$B_I(k) = \begin{cases} B_{required}(k) & \text{if source or destination} \\ 2B_{required}(k) & \text{else} \end{cases} \quad (6-10)$$

As discussed in 6.4.2, for node I, new real-time traffic flow k will also add edge traffic to node's traffic load.

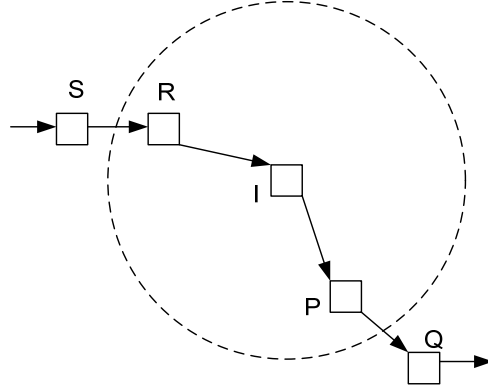


Figure 6.8 Traffic aggregation of new real-time flow

In Figure 6.8, the newly admitted real-time traffic flow k will add $B_{consume}(I, k)$ to the original consumed bandwidth at node I. $B_{consume}(I, k)$ can be written as:

$$B_{consume}(I, k) = Traffic_{S-R}(k) + Traffic_{R-I}(k) + Traffic_{I-P}(k) + Traffic_{P-Q}(k)$$

From (6-10),

$$B_{consume}(I, k) = B_R(k) + B_P(k) \quad (6-11)$$

Here node R is the uplink node of node I and node P is the downlink node of node I. A more general formula for $B_{consume}(I, k)$ can be written as:

$$B_{consume}(I, k) = B_{uplink(I)}(I, k) + B_{downlink(I)}(I, k) \quad (6-12)$$

Here, $B_{uplink(I)}(I, k)$ and $B_{downlink(I)}(I, k)$ can follow equation (6-10). $B_{consume}(I, k)$ puts an upper bound on the actual consumed bandwidth at intermediate nodes along the path for real-time flow k . By comparing $B_{consume}(I, k)$ and $B_{available}(I)$, the node can provide feedback to source node to decide whether the real-time traffic can be admitted or not. Equation (6-12) is good for unicast traffic flow. For multicast traffic flow, the nodes may have more downlink/two-hop downlink bandwidth consumption, which depends on the size of the receiver group and the topology of the network. For simplicity, Equation (6-12B) is introduced to take into account the multicast effect:

$$B_{consume}(I, k) = B_{uplink(I)}(I, k) + B_{downlink(I)}(I, k) + \Delta_{multicast} \quad (6-12B)$$

The procedure for traffic admission and bandwidth reservation is described in 6.4.4.

6.4.4 Traffic Admission/ Bandwidth Reservation

To differentiate admitted real-time traffic and rejected real-time traffic, “admitted” field is added to the packet header. For admitted real-time traffic, “admitted” field is set to 1; otherwise, “admitted” field is set to 0. In the proposed protocol, every node that may participate in traffic forwarding will monitor its available bandwidth and check if there’s any bandwidth violation. However, only source node decides which traffic session should be “admitted” or “rejected” based on the feedback from nodes in the network, like receivers [60]. In other words, admitted or rejected traffic is solely marked at source node (s), which means that all intermediate nodes will use the “admitted” field information assigned by source node (s) instead of adjusting this “admitted” field by themselves.

Once the node reserves the bandwidth for a certain real-time traffic, it adds the identification ID for this particular traffic flow, “flow_ID”, to its node’s structure, and updates its remaining available bandwidth estimate field, “available_bw_est”, which is equal to the total bandwidth for the node minus the reserved bandwidth for the traffic flow as described in 6.4.2. A time period T_{reserve} is also assigned for each real-time flow at every real-time forwarding node so that it can be used in bandwidth releasing and route break detection, which will be discussed in 6.4.5.

6.4.5 Bandwidth Releasing/Route Break Detection

The efficient resource release mechanism is needed in order to free the reserved resources at each node when the existing reserved routes are no longer in use. For the proposed QoS-enabled routing protocol, a time period T_{reserve} is assigned for each real-time flow at every real-time forwarding node. That is, the bandwidth reserved for each real-time flow at each forwarding node will only be effective for a certain time. If the node does not receive any subsequent data packet belonging to this specific real-time flow after T_{reserve} , the reservation becomes invalid. The time duration of T_{reserve} can be giving after traffic mark phase. It would be the inter-packet arrival time of the flow. It can also be a function of the minimum

bandwidth requirement of the flow. According to [61], by setting this reservation period, there is an upper bound on the burstiness, variance of the end-to-end delay of the flow.

T_{reserve} used in resource releasing can be used to signal possible route break. In previous multicast routing protocol without QoS support, *Packet_Delivery_Prediction* (PDP) and *Remain_Connected_Predict_Time* (RCPT) are used to help uplink node decide the stability of the route or if there is possible route to reach the receivers. If T_{reserve} is enough for the QoS-enabled routing protocol, then we may not need to rely on PDP and RCPT as much as what is in QoS-disabled routing protocol.

6.5 Message Modification

The revised packet format for “*Join-Query-Probe*”/ “*Join-Reply*” messages that will be used in this QoS-supported protocol is described here. For other control messages that may be used for real time/non real-time traffic, the packet format remains the same as in 4.3.

There are three messages for mesh construction: “*Join-Query-Probe*”, “*Join-Reply*” and “*Mesh_Refresh_Request*”. For QoS-supported multicast routing, “*Join-Query-Probe*” and “*Join-Reply*” packets have been modified to taking QoS requirement (bandwidth constraint) into consideration.

6.5.1 Mesh Construction Message Format

For QoS-enabled multicast routing protocol, new fields include bottleneck bandwidth (BB) and required bandwidth (RB) for certain real-time flow information in the packets.

a. “*Join-Query-Probe*”

The packet format for this message is based on the original “*Join-Query*” packet format defined in 4.3.1. It is modified as follows:

“Type + Reserved + Time-To-Live + Hop Count + Multicast Group IP Address + Sequence Number + Source IP Address + Previous Hop IP Address + BB + RB+ flowID”

Table 6.1 Join-Query-Probe message format

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Type	Reserved	Time-To-Live	Hop Count

Multicast Group IP Address			

Sequence Number			

Source IP Address			

Previous Hop IP Address			

Bottleneck Bandwidth (BB)			

Required Bandwidth (RB)			

flowID			

Here “Type” defines that it is “Join-Query-Probe” packet;

“Reserved” defines that the node ignores all packets;

“Time-To-Live” defines the number of hops this packet can traverse;

“Hop Count” defines the number of hops traveled so far by this packet;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Sequence Number” defines the sequence number assigned by the source to uniquely identify the packet;

“Source IP Address” defines the IP address of the node originating the packet;

“Previous Hop IP Address” defines the IP address of the last node that has processed this packet.

“BB” defines the bottleneck bandwidth along the path the packet has traversed.

“RB” defined the required bandwidth the application need.

“flowID” defined the real-time traffic flow ID.

b. “Join-Reply”

This message is based on the original “Join-Reply” packet format defined in 4.3.1. It is modified as follows:

“Type + Count + R + F + Reserved + Multicast Group IP Address + Receiver IP address + Previous Hop IP Address + Sequence Number + Packet Delivery Prediction (PDP) + Sender IP Address [1] + Next Hop IP Address [1] + Remain_Connected_Predict_Time [1] + BB_max[1] + RB[1] + flowID[1] + Sender IP Address [2] + Next Hop IP Address [2] + Remain_Connected_Predict_Time [2] + BB_max[2] + RB[2] + flowID[2] + ... + Sender IP Address [n] + Next Hop IP Address [n] + Remain_Connected_Predict_Time [n] + BB_max[n] + RB[n] + flowID[n]”

Table 6.2 Join-Reply message format

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Type										Count										R F										Reserved									

Multicast Group IP Address																																							

Receiver IP Address																																							

Previous Hop IP Address																																							

Sequence Number																																							

Packet Delivery Prediction (PDP)																																							

Sender IP Address [1]																																							

Next Hop IP Address [1]																																							

Remain_Connected_Predict_Time [1]																																							

BB_max [1]																																							

RB [1]																																							

flowID [1]																																							

⋮																																							

Sender IP Address [n]																																							

Next Hop IP Address [n]																																							

Remain_Connected_Predict_Time [n]																																							

BB_max [n]																																							

RB [n]																																							

flowID [n]																																							

Here “Type” defines that it is “Join-Reply” packet;

“Count” defines the number of (Sender IP Address, Next Hop IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“F” defines Forwarding Group flag. This flag is set when the packet is transmitted by a forwarding group node;

“Reserved” defines that the node ignores all packets;

“Multicast Group IP Address” defines the IP address of the multicast group;

“Receiver IP address” defines the IP address of the receiver in the multicast group; If this information has to be used, each receiver node will most likely have two IP addresses, one for the multicast group, and the other for individual IP address. If this is not feasible for Mobile Ad Hoc Network, additional information other than this address will be needed so that when building the data forwarding tree, each receiver will be the leaf of the data forwarding tree and can successfully receive data packets.

“Previous Hop IP Address” defines the IP address of the last node that has processed this packet;

“Sequence Number” defines the sequence number assigned by the previous node to uniquely identify the packet;

“Packet Delivery Prediction (PDP)” defines the stability of the previous hop node based on mobility prediction model and battery model in 4.2.3 and 4.2.4.

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group;

“Next Hop IP Address [1...n]” defines the IP addresses of the next nodes this packet is target to;

“Remain_Connected_Predict_Time [1...n]” defines the predicted time the previous hop node will keep connected with current node. This value can be calculated using Mobility Prediction Model stated in 4.2.3. Here for each {uplink node, downlink node} pair, it will have specific Remain_Connected_Predict_Time (RCPT) value. If this value is not easy to extract for every node pair, we can choose and store some predefined RCPT value in the uplink node and let uplink node decides what the RCPT value should be given to the “link” to a specific downlink node based on its PDP and downlink node’s PDP value;

“BB_max [1...n]” defines the maximum bottleneck bandwidth along the path a specific

“*Join-Query-Probe*” packet has traversed.

“RB [1...n]” defined the required bandwidth the application need.

When the mesh is built, the real-time data packet should be forwarded following 6.3.3 in the QoS-enabled network, and best effort traffic is forwarded after being regulated by rate shaper. If QoS requirement is not required, the packets are forwarded according to 4.2.2.

c. “Mesh_Refresh_Request”

This packet is the same for both real-time traffic and non real-time traffic. It has the following format:

“Type + Count + R + F + Reserved + Multicast Group IP Address + Receiver IP address + Previous Hop IP Address + Sequence Number + Sender IP Address [1] + Next Hop IP Address [1] + Sender IP Address [2] + Next Hop IP Address [2] + ... + Sender IP Address [n] + Next Hop IP Address [n]”

Table 6.3 Mesh_Refresh_Request message format

0	1								2								3																						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Type								Count								R		F		Reserved																			

Multicast Group IP Address																																							

Receiver IP Address																																							

Previous Hop IP Address																																							

Sequence Number																																							

Sender IP Address [1]																																							

Next Hop IP Address [1]																																							

flowID [1]																																							

⋮																																							

Sender IP Address [n]																																							

Next Hop IP Address [n]																																							

flowID [n]																																							

Here “Type” defines that it is “Mesh_Refresh_Request” packet;

“Count” defines the number of (Sender IP Address, Next Hop IP Address) combinations;

“R” defines the acknowledgement request flag. This flag is set when active acknowledgement packet is requested;

“F” defines Forwarding Group flag. This flag is set when the packet is transmitted by a forwarding group node;

“Reserved” defines that the node ignores all packets;

“Sender IP Address [1...n]” defines the IP addresses of the sources of this multicast group that use this node as intermediate node to transmit data packet to receivers.

“Next Hop IP Address [1...n]” defines the IP addresses of the next nodes this packet is target to. The IP address is extracted from the node’s routing table that is used for current data forwarding (PATH_SELECT_Flag set for the routing table);

6.6 Data Node Structures and Routing Table

To make sure that the data packets will be transmitted via selected routes and to satisfy bandwidth constraint for real-time traffic, each forwarding group member will maintain the following variables and data structures:

1. The uplink nodes’ and downlink nodes’ addresses.
2. The source IP addresses and receiver Addresses/Multicast IP address.
3. Timer Δt : used to count packets sent from this node during certain period of time. This can be a global predefined value so that every node will count the number of packets during the same time span. It is counted down from MAX_time to 0 and repeats.
4. Packet counter: Used to count the number of packets that are sent during timer Δt . The value is reset at the beginning of every time period.
5. The node_battery value from battery model in 4.2.4.
6. Packet delivery prediction (PDP) based on mobility prediction in 4.2.3 and battery model in 4.2.4.
7. Remain_Connected_Predict_Time for a specific downlink node.
8. PATH_SELECT_Flag bit. If the node is selected in data forwarding route, PATH_SELECT_Flag bit is set to 1; otherwise 0.

9. Backup_Select_Flag bit. If the node is selected by receiver and uplink node of receiver that is currently part of data forwarding tree, Backup_Select_Flag bit is set to 1; otherwise 0.
10. BK_Uplink_Receiver_Addresses. Addresses of the receiver and uplink node of receiver pair that selects the node as backup node for data retransmission. This is valid when Backup_Select_Flag is set to 1.
11. Have_Backup_Node_Flag bit. This bit is set to 1 if a receiver and its uplink node that is currently part of data forwarding tree agree that they will have backup node for data retransmission.
12. Backup_Address. The receiver and its uplink node that is currently part of data forwarding tree will store this Backup_Address of backup node so that when data retransmission is needed, backup node can join the route. The data flow is then uplink node -> backup node -> receiver during data retransmission.
13. Data buffer size: it defines the size of the data packets that will be stored in case data retransmission is requested.
14. Timer t_converge and t_converge_left: t_converge defines QoS convergence time. It is used for the node to collect enough information about the ongoing traffic that is carried on by itself and its one-hop neighbors. Initially t_converge_left is set as the same value as t_converge. Once a node receive the first data traffic with ToS bit set, the counter t_converge_left starts decreasing until 0. Once t_converge_left hit 0, real-time data traffic will be forwarded via node with RTF_FLAG and FG_FLAG set rather than tmp_RTF_FLAG set.
15. tmp_RTF_FLAG: temporary real-time forwarder flag. This flag is set during traffic mark phase (t_converge time period).
16. RTF_FLAG: real-time forwarder flag.
17. Timer T_collect and T_collect_left: T_collect and T_collect_left have the same value. The receiver will use the time span specified in T_collect field to collect enough Join-Query-Probe. When it received the first Join-Query-Probe, it will decrease the time

- counter $T_{collect_left}$ until it reaches 0. Once $T_{collect_left}$ hits 0, Join-Reply will be sent and $T_{collect_left}$ will be set to $T_{collect}$.
18. B_{max} : Maximum bandwidth the node has.
 19. $B_{ongoing}$: The bandwidth the node consumes.
 20. Δbw : fluctuation bandwidth that is used in case bandwidth fluctuation in Mobile Ad Hoc Network may affect the forwarding node decision for real-time traffic.
 21. FG_FLAG : Forwarding Group flag. It is set to 1 if the node is a Forwarding node.
 22. $flow_ID$: admitted real-time traffic flow identifier.
 23. $available_bw_est$: available bandwidth at the node.
 24. $T_{reserve}$: time period assigned for a specific admitted real-time traffic flow.

6.7 Summary

In this chapter, a modified protocol of Chapter 4 that enables Quality of Service (QoS) support is presented. The distributed resource probing is interwoven with the mesh creation process. The source node uses the mesh construction procedures to probe the bandwidth information along possible routes to the receivers. Traffic mark phase is a temporary phase that is used to validate and help nodes decide whether to reserve the resources for the real-time traffic flow. Local traffic regulation of non real-time traffic is used to assist the QoS guarantee of real-time traffic. Calculation of bandwidth availability estimation and bandwidth consumption for a new real-time flow is also presented to predict whether a real-time flow should be admitted or rejected.

Chapter 7

Quality-of-Service Multicast Routing Performance Simulation and Evaluation

7.1 Scenario Setup

The simulator for evaluation routing protocol is implemented using Network Simulation (ns2) in Linux. The network size is 100 mobile nodes placed randomly over a 1000m x 1000m area. The radio propagation range for each node is 250m, and the channel capacity is 2M bits/s. The movement model of the nodes in the simulations is the Random Waypoint model. The simulation time is 1000 second. Each node independently starts at a random location in the simulation area and remains stationary for a period of pause time. The node then generates a uniformly distributed new location, which is a random destination inside the simulation area, to move to. The mobility speed is uniformly distributed between a certain mobility speed, say 0 m/s and 20 m/s, with a pause time of 0 seconds corresponding to a continuous motion whereas a pause time of 1000 seconds corresponding to a static scenario.

a. Channel and Radio Model

The propagation models in ns2 are free space model, two-ray ground reflection model and the shadowing model [65]. Free space model assumes the ideal propagation condition that there is only one clear line-of-sight path between the transmitter and receiver. The two-ray ground reflection model considers both the direct path and a ground reflection path. The shadowing model consists of two parts. The first part is known as path loss model, and the second part reflects the variation of the received power at certain distance. In the simulation, two-ray ground reflection model is used.

b. MAC Protocol

The IEEE 802.11 MAC protocol with distributed coordination function (DCF) is used as the MAC layer. DCF uses a RTS/CTS/DATA/ACK for all unicast packets. For multicast data packets, they are sent out without using ACK in the sequence.

c. Traffic Pattern

The simulation is used to evaluate the efficiency of transmitting real-time multimedia traffic. Multimedia application, like voice and video application, has different requirements and characteristics compared to data traffic.

Voice traffic consists of small packet with size ranging from 80 bytes to 256 bytes. Frames are sent periodically which lead to constant bit rate traffic (21kbps to 320kbps) [63]. Voice traffic has low tolerance to delay and packet loss. When delay is large, the conversation may be disrupted. A delay of 100-150 ms is considered acceptable. When the delay is above 200ms, it can adversely affect the conversation quality.

Streaming video traffic has a highly variable frame size. Packets are large in size, which usually range from 65 bytes to 1500 bytes [63]. Compared to voice traffic, streaming video traffic needs more bandwidth, and is not highly delay or jitter sensitive as voice traffic. It can tolerate more losses than voice traffic.

In the simulation scenarios, voice and video traffic all use Constant Bit Rate (CBR) traffic for simplicity.

d. Simulation Scenario Summary

The simulation executed 8 scenarios for QoS-enabled multicast routing protocol. The voice traffic is 80 bytes packets at a rate of 30 packets per seconds (19.2Kkbps). Four scenarios were for eight voice traffic flows in the network with maximum nodes speed as 0m/s, 5m/s, 10m/s and 15m/s. Four scenarios were for twelve voice traffic flows in the network with maximum nodes' speed as 0m/s, 5m/s, 10m/s and 15m/s. The receiver group size is 5.

The simulation executed 10 kinds of scenarios for protocol comparison between QoS-enabled and QoS-disabled multicast routing protocols. Five scenarios tested protocol

performance under different mobility speeds with receiver group size set to 3. The maximum speeds are 0m/s, 5m/s, 10m/s, 15m/s and 20m/s. Five scenarios tested protocol performance for different sizes of receiver group as 2, 3, 5, 8 and 10 when the average mobility speed was 10 m/s with pause time 100ms. Forty runs were executed for each scenario per routing protocol. The collected data is the average over forty runs. Table 7.1 shows the summary of the simulation parameters and the chosen value set for each parameters.

Table 7.1 Simulation Parameters

Parameter	Value set
Number of Nodes	100
MAC protocol	802.11
Bandwidth	2Mbps
Propagation model	Two-ray Ground
Mobility model	Random Waypoint
Speed(m/s)	[0, 0], [0, 5], [0, 10], [0, 15], [0, 20]
Group size	2, 3, 5, 8, 10
Pause time	100 seconds
Traffic type	CBR, FTP
Simulation time	1000 seconds

During the comparison simulation, the background traffic is TCP best-effort traffic. There are four TCP flows in all scenarios. All TCP flows are FTP type of traffic with packet size of 128 bytes. There are four voice flow and one video traffic flow active in the scenarios. The voice traffic is 80 bytes packets at a rate of 25 packets per seconds (16Kkbps). The video traffic is 512 bytes packet at a rate of 30 packets per seconds (122.88Kbps).

For the proposed protocol without QoS support, denoted as Ad-hoc Multicast Routing (AMR_NO), the MESH_REFRESH_INTERVAL was set as 30 seconds. MIN_REFRESH_INTERVAL was set to 3 seconds, while MAX_REFRESH_INTERVAL was set to 30 seconds. The Hello interval was fixed at 3 seconds.

For QoS-enabled ad-hoc multicast routing protocol, denoted as AMR_QoS, the MESH_REFRESH_INTERVAL was set as 20 seconds. MIN_REFRESH_INTERVAL was set to 3 seconds, while MAX_REFRESH_INTERVAL was set to 20 seconds. The Hello

interval was fixed at 3 seconds. The resource reservation time T_{reserve} is set to 150 ms for voice traffic, and is set to 800ms for video traffic in the simulation.

7.2 Performance Evaluation

7.2.1 Performance Metrics

To evaluate the effectiveness of the QoS-enabled routing protocol, the following performance metrics were used:

- a) Traffic Admission Ratio: Ratio between the number of data packets sent to the network from source to the number of data packets generated at the source [2]. The metric shows the effectiveness of traffic admission scheme.
- b) Average End-to-End delay: The end-to-end delay between the time when one packet is received at every receiver and time it was sent out at the sender. The delay consists of propagation, transfer times, retransmission delays at the MAC, queuing at the interface queue, buffering during routing. The average over all the packets received is then computed.
- c) Packet Delivery Ratio: The average number of multicast DATA packets actually received by one destination node over the number of transmitted packets from the source.

7.2.2 Performance Analysis

For group size as 5 and different maximum mobility speed as 0m/s, 5m/s, 10m/s and 15m/s, simulations with eight real-time traffic flows and twelve real-time traffic flows were conducted. Simulation result of traffic admission ratio is shown in Figure 7.1.

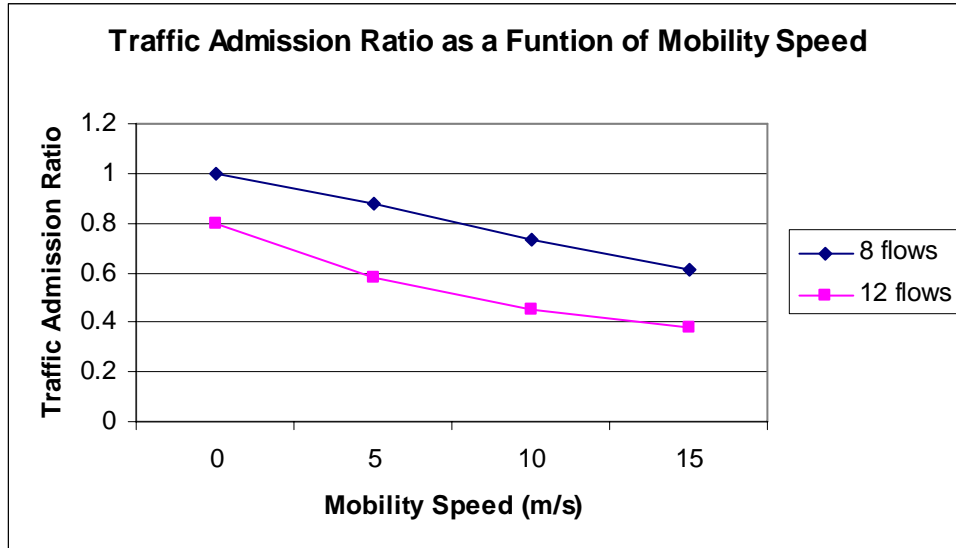


Figure 7.1 Traffic Admission Ratio versus Mobility Speed

From Figure 7.1, it shows when the nodes are stationary and there are eight voice traffic flows in the network, all traffic flows are admitted, while for twelve traffic flows, 80% of the traffic is admitted. As the nodes' mobility speeds increase, the traffic admission ratio decreases. When the node's maximum mobility speed is 5m/s, for eight traffic flows, 87.5% of the traffic is admitted, while for twelve traffic flows, only 58% of the traffic is admitted. This is reasonable because as nodes move around at a higher speed, more links are more unstable and the nodes may tend to reserve resource as they enter a possible route for another real-time traffic flow, which result in the overbook of resources and some real-time traffic flows are rejected due to the deficiency of available resources.

Simulation result of average end-to-end delay for admitted traffic versus mobility speed is shown in Figure 7.2.

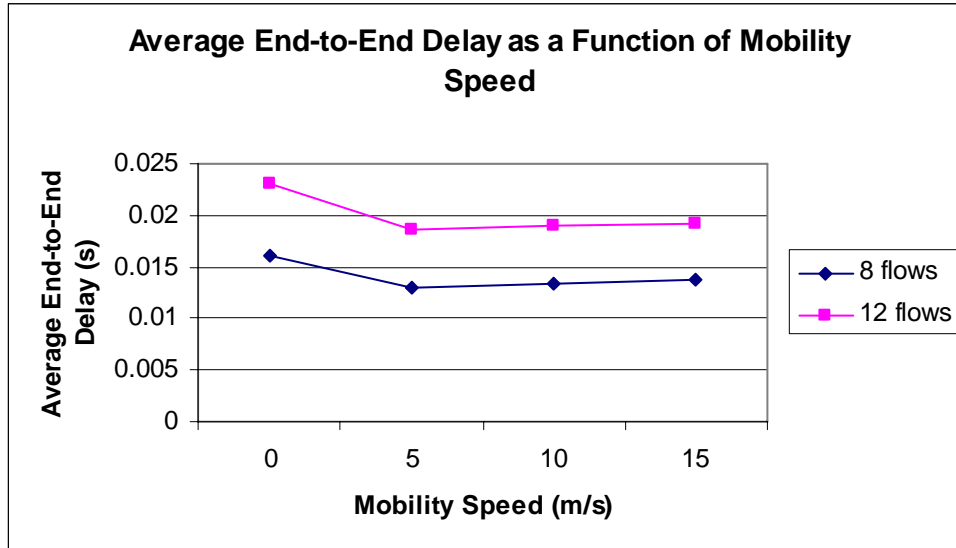


Figure 7.2 Average End-to-End Delay versus Mobility Speed

From Figure 7.2, it shows as nodes' mobility speeds increase, the average end-to-end delay decreases when the maximum mobility rate is 5m/s, and increases when the rate is above 5m/s. This is accompanied by the decreased traffic admitted ratio. It is because as the mobility speed increases from 0m/s to 5m/s, less real-time traffic is admitted in the network. In the 12 flows' plot, when the maximum mobility speed is 5m/s, traffic admission ratio is 58%, which means only 72.5% of the traffic is admitted as compared to the stationary scenario (speed = 0m/s). The network is low in mobility and has less traffic, thus the average end-to-end delay is relative small compared to other scenarios. However, as the mobility increases, the probability of route break and collisions increase. The average end-to-end delay increases slightly when the nodes move at a faster speed.

Packet delivery ratio for admitted traffic versus mobility speed is shown in Figure 7.3.

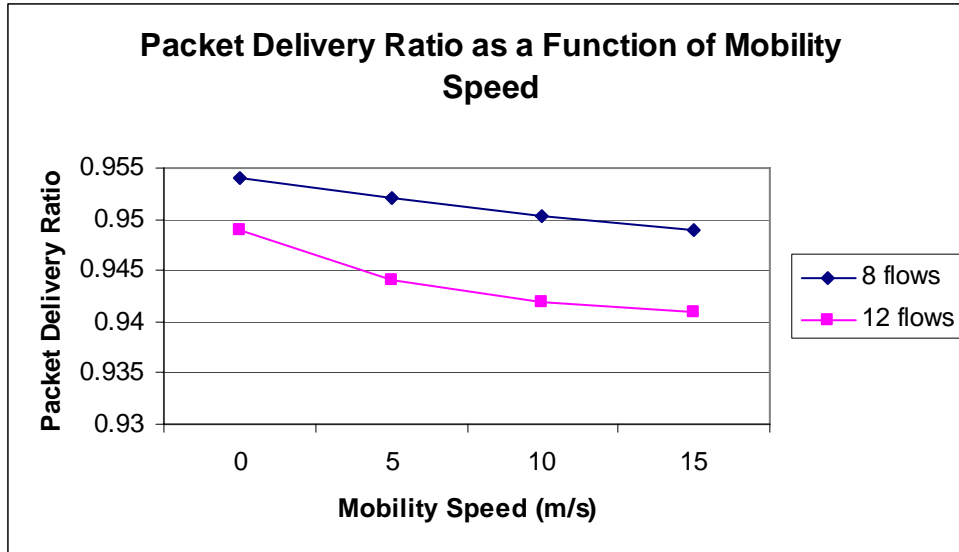


Figure 7.3 Packet Delivery Ratio versus Mobility Speed

From Figure 7.3, it shows as the mobility speed increases, the packet delivery ratio for admitted traffic decreases. For 8 flows' and 12 flows' plots, when the maximum mobility speed is 15m/s, the packet delivery ratios are 94.9% and 94.1% respectively for admitted traffic load. The system is maintaining a relative high packet delivery ratio when the node's mobility speed increases due to decreased real-time traffic load admitted in the system. The QoS-enabled multicast routing protocol can admit and reject real-time traffic based on resource availability and can reliably deliver the admitted traffic.

From above simulation, when the node's mobility speed is low or moderate, the QoS-enabled multicast routing protocol produces desired behavior because of availability of quality links.

7.3 Simulation Comparison of QoS-enabled and QoS-disabled Multicast Routing Protocols

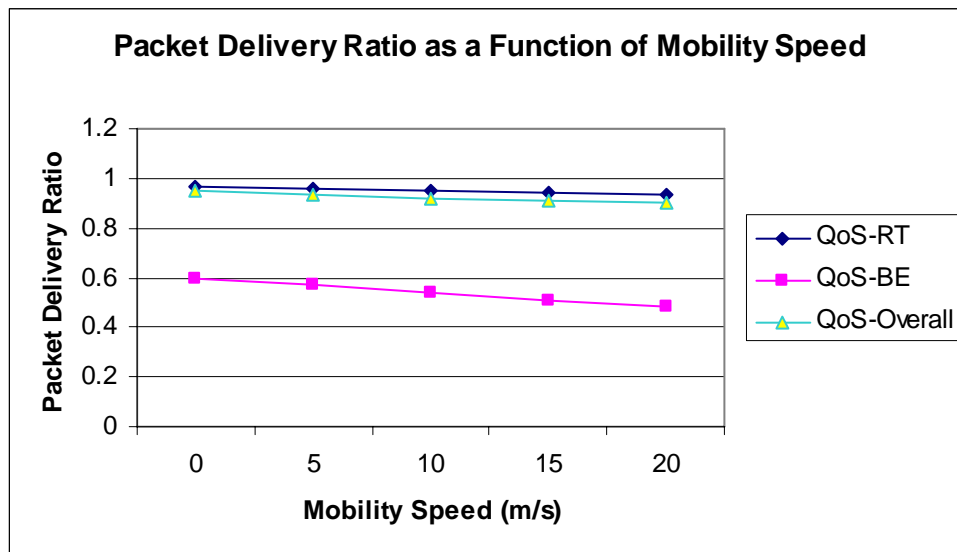
Comparison study of QoS-disabled and QoS-enabled multicast routing protocols was carried out via simulation. Real-time traffic flows like voice and video applications as well as best-effort traffic like FTP traffic was used. The study concentrated on protocols' performance under different mobility speeds and group size. The performance metrics used for the

evaluation of proposed QoS-disabled and QoS-enabled protocols include packet delivery ratio, average end-to-end delay and control overhead.

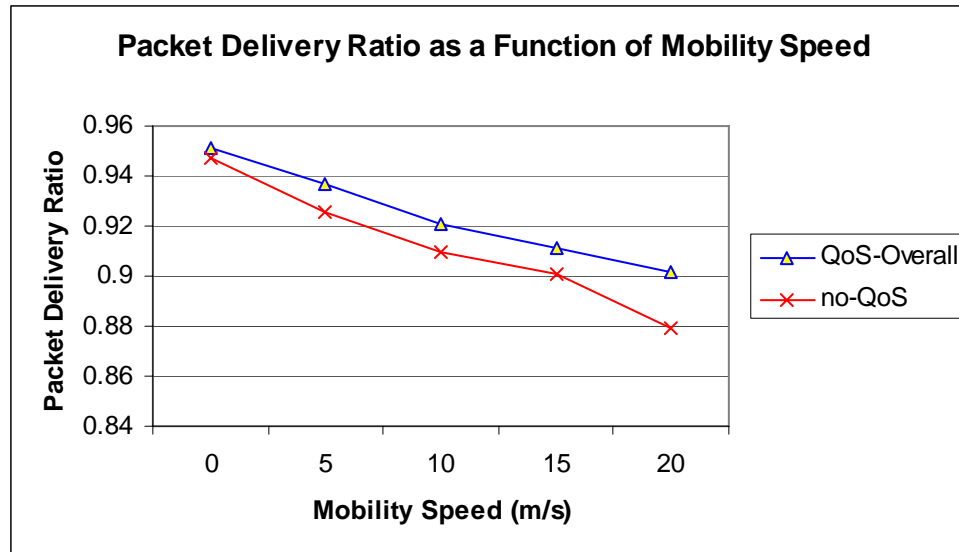
7.3.1 Packet Delivery Ratio

In this section, simulation results from QoS-enabled protocol (AMR_QoS) are compared with QoS-disabled protocol (AMR_NO) in terms of packet delivery ratio. In addition, the simulation of each protocol is evaluated in static group members such that all group members remain members after joining a multicast session.

7.3.1.1 Mobility Speed



(a) Packet Delivery Ratio for QoS-enabled protocol

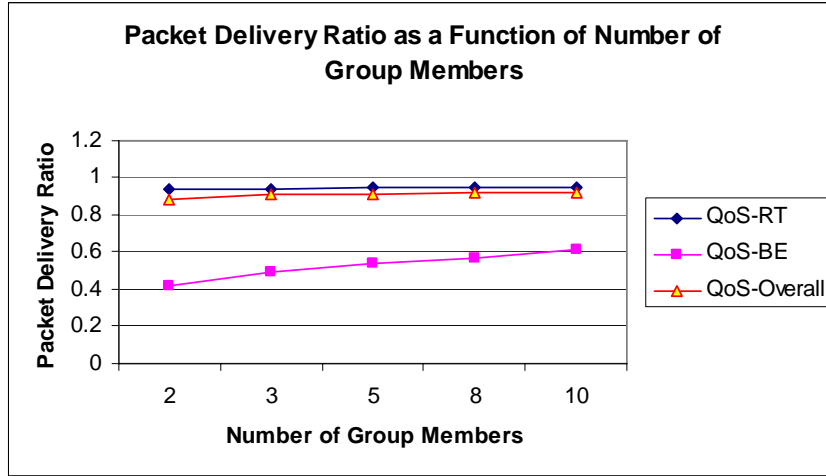


(b) Packet Delivery Ratio for Overall Traffic

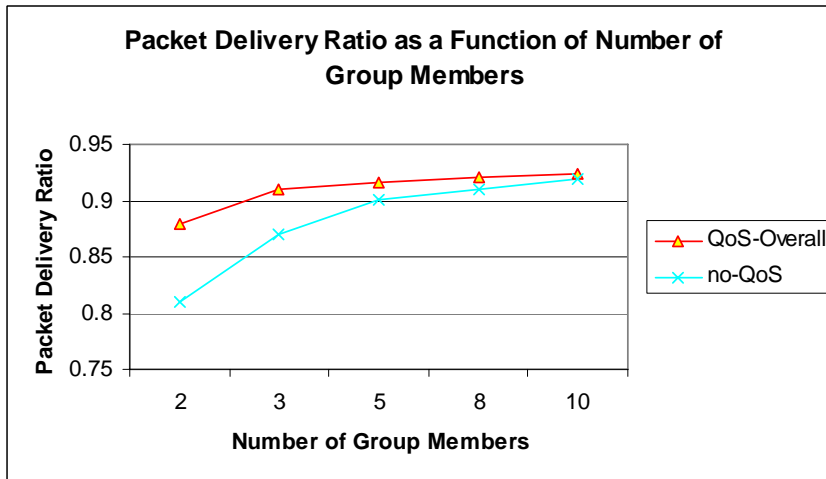
Figure 7.4 Packet Delivery Ratio versus Mobility Speed

From Figure 7.4, the packet delivery ratio for real-time traffic is significantly higher than best-effort traffic. The packet delivery ratio for best-effort traffic when using QoS-enabled routing protocol is lowest among all. This is because the best-effort traffic is regulated by rate shaper and once the shaper queue is full, the best-effort packets will be dropped. However, the overall average packet delivery ratio for QoS-enabled routing protocol is slightly higher than QoS-disabled routing protocol. The control packets are given high priority as admitted real-time traffic and bypass the rate shaper regulation. In this way, the traffic delivery structure for QoS-enabled protocol is more efficient than QoS-disabled one.

7.3.1.2 Number of Group Members



(a) Packet Delivery Ratio for QoS-enabled protocol



(b) Packet Delivery Ratio for Overall Traffic

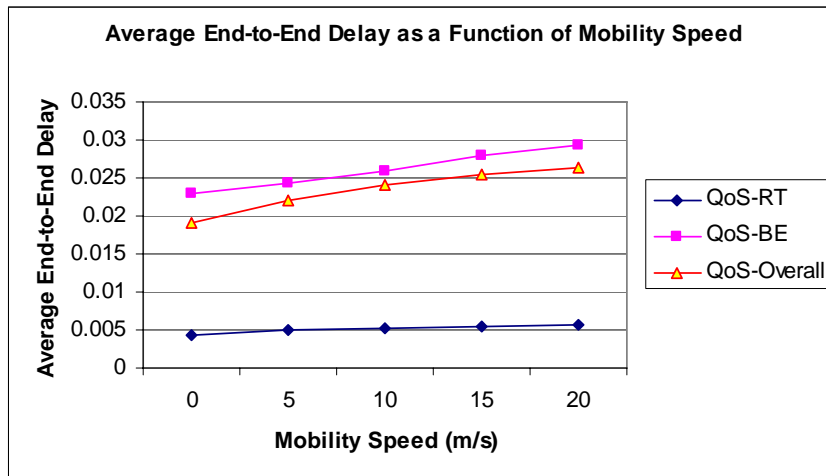
Figure 7.5 Packet Delivery Ratio versus Number of Group Members

The packet delivery ratio for real-time traffic for different group size was shown in Figure 7.5. The source is set to 1 and average mobility speed is 10 m/s with pause time 100ms. The packet delivery ratio for real-time traffic is higher than best-effort traffic when using QoS-enabled multicast routing protocol. As group size increases, less real-time traffic may be admitted if there's not enough bandwidth to reserve for the real-time traffic flow, thus more bandwidth can be used to send best-effort traffic and relative fewer best-effort packets may be dropped due to overflow of shaper queue. The overall packet delivery ratio for QoS-

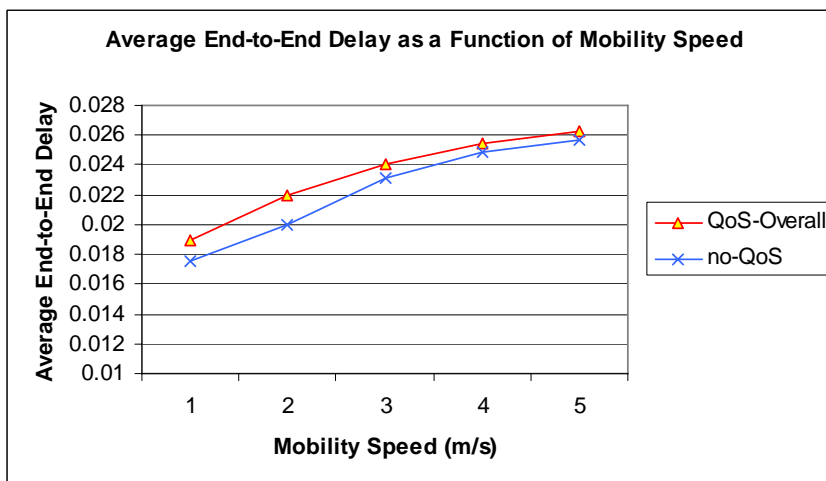
enabled protocol is slightly higher than QoS-disabled protocol. As group size increases, the packet delivery ratio also increases. This is due to that larger number of group member help building a well structured packet delivery structure. Though larger bandwidth may be consumed, the QoS-enabled protocol guarantees that admitted real-time traffic's requirement will be satisfied while best-effort traffic will be regulated.

7.3.2 Average End-to-End Delay

7.3.2.1 Mobility Speed



(a) Average End-to-End Delay for QoS-enabled Protocol

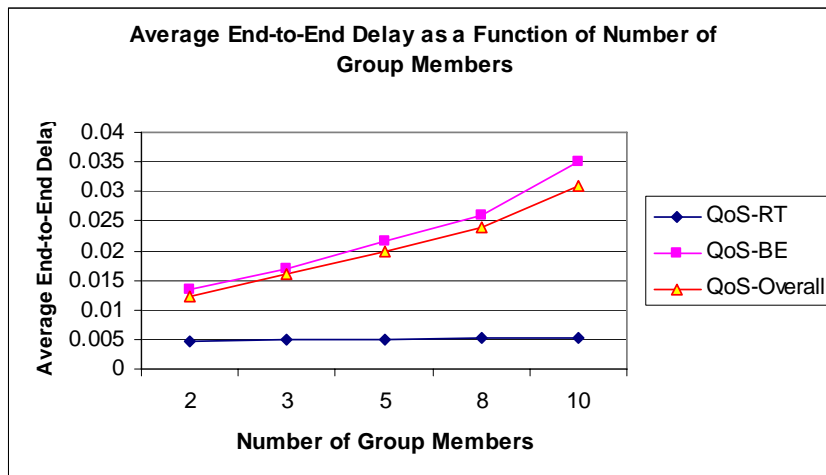


(b) Average End-to-End Delay for Overall Traffic

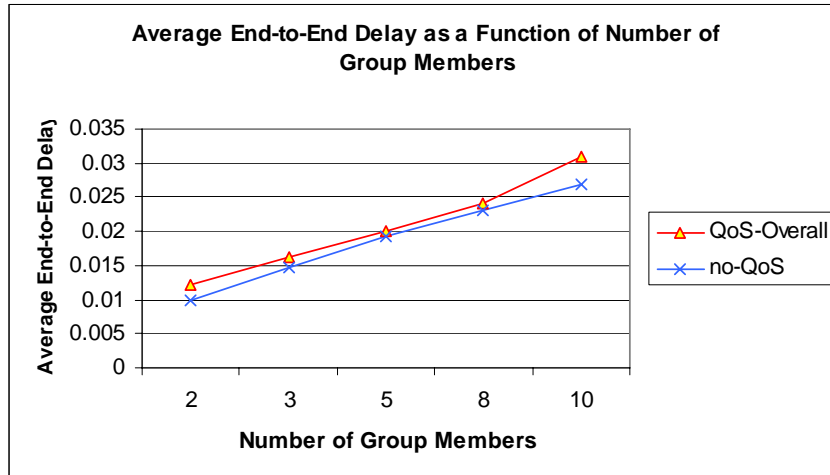
Figure 7.6 Average End-to-End Delay versus Mobility Speed

As shown from Figure 7.6, the average end-to-end delay for real-time packets was controlled by the QoS-enabled multicast routing protocol. As the mobility speed of the node increases, the average end-to-end delay of the real-time packet increases slightly. The increment is bounded because as the mobility speed increase, the best-effort traffic is regulated by the rate shaper. The average end-to-end delay for the best effort traffic is higher than the delay using the multicast routing protocol without QoS support because of longer queue for best-effort traffic. Due to the larger amount of best-effort traffic than real-time traffic in the network, the average end-to-end delay for all packets in the system is higher when using QoS-enabled multicast routing protocol than QoS-disabled routing protocol. However, the QoS-disabled multicast routing protocol can not provide low delay for real-time traffic as it can differentiate real-time traffic from best-effort traffic.

7.3.2.2 Number of Group Members



(a) Average End-to-End Delay for QoS-enabled Protocol



(b) Average End-to-End Delay for Overall Traffic

Figure 7.7 Average End-to-End Delay versus Number of Group Members

Figure 7.7 shows average end-to-end delay for QoS-enabled and QoS-disabled multicast routing protocol for different number of group members. The source is set to 1 and average mobility speed is 10m/s with pause time 100ms. End-to-end delay is low for real-time traffic when using QoS-enabled multicast routing protocol. With the increase of group size, the average end-to-end delay for real-time traffic is a little affected as the QoS requirements of the admitted traffic flow are satisfied. The average end-to-end delay for best-effort traffic is increasing as group size increases. This can be explained by the larger bandwidth consumption due to larger group size in the multicast network. The overall average end-to-end delay is larger for QoS-enabled routing than QoS-disabled routing because of the amount of best-effort traffic in the system, which sets off the overall delay.

7.3.3 Control Overhead

7.3.3.1 Mobility Speed

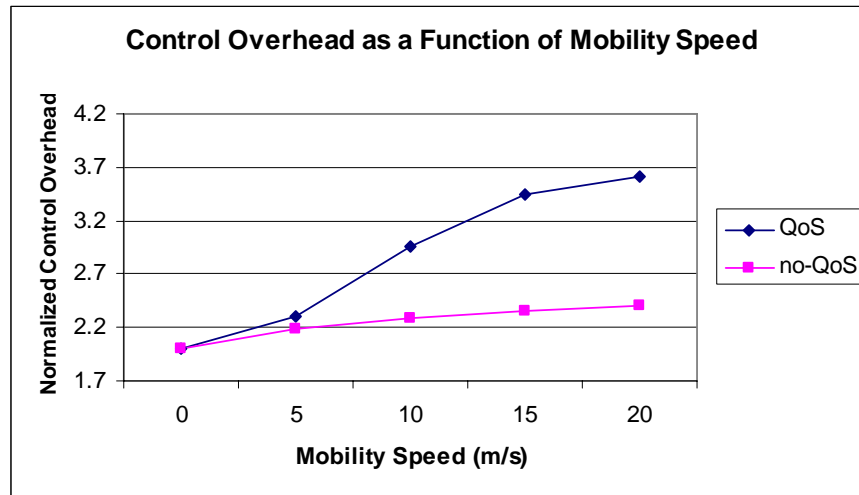


Figure 7.8 Control Overhead versus Mobility Speed

Figure 7.8 illustrates the normalized control overhead under different node mobility. The control overhead for QoS-enabled and QoS-disabled multicast routing protocols increases as the node mobility speed increases. As the nodes move at a faster speed and the links break more frequently, the control packets are sent out more frequently. The QoS-enabled multicast routing protocol has a higher normalized control overhead than QoS-disabled one. This is because as the nodes move around faster, the bandwidth requirement for the real-time traffic flow has the tendency to be more stringent to satisfy than solely using link stability models in QoS-disabled routing protocol.

7.3.3.2 Number of Group Members

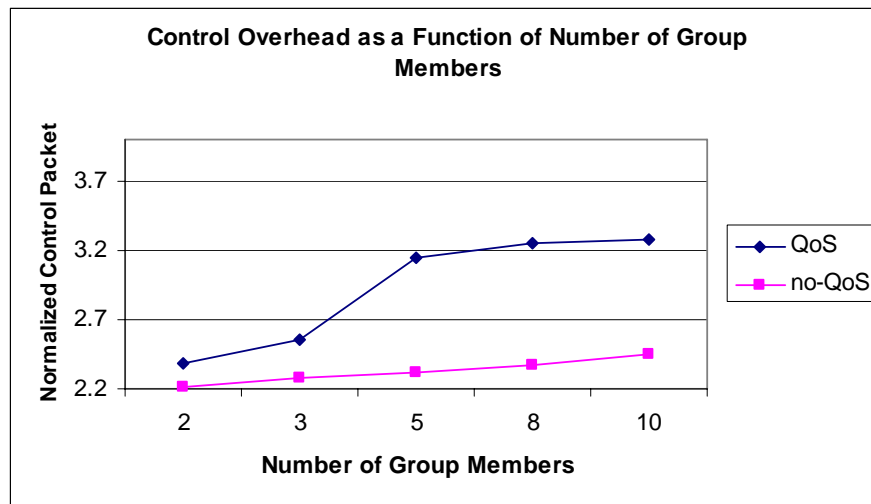


Figure 7.9 Control Overhead versus Number of Group Members

From Figure 7.9, the control overhead for both QoS-enabled routing and QoS-disabled routing increases as the group size increases. This is due to the fact that as more group members in the system, any unstable link or bandwidth fluctuation may introduce the procedure to rebuild the structure in the system, which means more control packets flooded. For the simulation result, when the group size is relative small, the control overhead for QoS-enabled protocol is relative not that large; however, as the group size increases, the control overhead increases dramatically at first and then flattens out. This is because as the group size increases, for real-time traffic flows, more bandwidth needs to be reserved for the multicast application, and the admission control scheme will reject certain real-time traffic flow so that the admitted flows' requirements can be satisfied. As fewer real-time traffic flows in the network, the related control-overhead will be maintained at a certain level under the same mobility speed of the nodes.

7.4 Summary

The simulation performed under different mobility speed for QoS-enabled multicast routing protocol demonstrates that admission control scheme in QoS-enabled routing protocol works.

When there is not enough bandwidth to accommodate certain real-time traffic flow, the flow will be rejected. Simulation shows that when the node's mobility speed is low or moderate, the packet delivery ratio and average end-to-end delay can be maintained at an accepted value.

Simulation and comparisons of the two protocols, QoS-enabled multicast routing protocol (AMR_QoS), which was introduced in Chapter 6, and QoS-disabled multicast routing protocol (AMR_NO), which was presented in Chapter 4, are discussed. Packet delivery ratio, average end-to-end delay and control overhead were studied as functions of the node mobility speed for two protocols in the 1 source and 3 receivers' scenario, and of different group size under same maximum mobility speed. The simulation shows that the QoS-enabled routing protocol can guarantee higher packet delivery ratio, and relative stable average end-to-end delay for real-time traffic, while the best-effort traffic is suppressed with relative low packet delivery ratio and longer average end-to-end delay. Even though the overall average end-to-end delay for QoS-enabled routing protocol is larger than QoS-disabled one, the real-time applications were better served using QoS-enabled one. Simulation also shows that the drawback for QoS-enabled routing protocol is that the control overhead is higher when the real-time traffic is heavier in the network with higher mobility speed. The control-overhead is not that sensitive to group size at the cost of fewer real-time traffic flows being admitted from simulation result. The bandwidth requirement in QoS-enabled protocol is more stringent than only considering the link stability as used in the QoS-disabled counterpart.

Chapter 8

8. Conclusions and Future Works

8.1 Conclusions

The multicast routing protocols for Mobile Ad-hoc Network (MANET) with and without Quality of Service (QoS) support were introduced in Chapter 6 and Chapter 4 respectively. Performance evaluation was conducted using Network Simulator version 2.28 (ns-2.28).

The multicast routing protocol without QoS support builds a mesh structure and forwards data packets using data forwarding tree on the mesh. The mesh structure is updated only when no reliable data forwarding tree can be found to cover all receivers. The reliability of data forwarding tree, or in other word, the reliability of links is defined by Packet Delivery Prediction (PDP), which is used to predict the stability of links in the system. To help predict such stability of links, models like battery model and route selection model were introduced to assist nodes to select more stable links for data transmission. The mesh refresh interval is thus elongated, which means there may be less Join_Query and Join_Reply packets in the network. Simulation showed that data overhead is reduced compared with ODMRP. When the nodes have low to moderate mobility speed, the proposed protocol without QoS support has higher packet delivery ratio than mesh based protocol ODMRP.

The multicast routing protocol with QoS support aimed at providing Quality of Service (QoS) to real-time applications, like voice or video applications. Aside from using the battery model, the routes are selected based on the bandwidth availability at each intermediate node. Real-time traffic can be either admitted or rejected according to available bandwidth of the nodes. The bandwidth is reserved on a per flow basis. Every node monitors the ongoing traffic through itself and the neighboring nodes. The information is used to make resource reservation decision when there is real-time traffic entering the system. Resource release scheme uses time-out approach: each intermediate node will only reserve the bandwidth for a

specific traffic flow for a given time period. The selection of time-out interval has taken into account the instability due to mobility and the packet inter-arrival delay requirement for the real-time application. Simulation shows that the QoS-enabled protocol can admit and reject real-time traffic flows according to nodes' bandwidth availability. When the node's mobility is low or moderate, the average end-to-end delay and packet delivery ratio can meet the QoS performance expectation.

The comparison of the two proposed protocols was studied. From simulation, for QoS-enabled routing protocol, the performance of overall traffic may be degraded compared to QoS-disabled counterpart. However, it shows the QoS-enabled multicast routing protocol can guarantee the average end-to-end delay and packet delivery ratio for real-time traffic flows while QoS-disabled protocol can not differentiate real-time traffic from best-effort traffic flow. Simulation also shows that the control-overhead for QoS-enabled protocol is higher than QoS-disabled one for different mobility speed and group size. However, QoS-enabled protocol can maintain the control-overhead at the cost of reduced admitted real-time flows when group size increases.

8.2 Future Work

Efficient support of multicast communications is essential in order to provide service like group audio and video conferencing, dissemination of data to a set of receivers or collaborating of a group of users. Multimedia applications require Quality of Service (QoS) guarantees. Most of these interactive services have very strong requirements regarding end-to-end delay and bandwidth. QoS-enabled multicast routing protocols should do their best to control the congestion and manage resources for real-time multicast service in MANETs, providing at least soft-QoS guarantees.

The QoS-enabled multicast routing protocol proposed consists of mesh construction and traffic mark phase before the resources are set aside for the real-time traffic flows in the network. That means there's a certain time period that the QoS requirements are not being

satisfied. Also, for the proposed QoS-enabled routing protocol, there's no service differentiation or different service levels for real-time applications. Real-time traffic flow is admitted on a first-come first-serve scheme, and no priority is given to a specific traffic flow or class. The drawback for this scheme is that the more urgent service may be delayed if the less urgent service enters the system earlier. The characteristics of the traffic will also affect the performance of the QoS-enabled multicast routing protocol. Constant Bit Rate (CBR) traffic was used for both voice and video application in the simulation; while in the real world, there may be other types of traffic, e.g., Variable Bit Rate (VBR), which may pose different requirement for the resource reservation and traffic admission schemes.

For future QoS-enabled multicast routing, different types of traffic should be taken into consideration and a prioritized scheme may be implemented so that real-time traffic that is urgent can be taken care more promptly than other less urgent real-time traffic. The study in this thesis is only concentrate on routing part, while a cross-layer QoS support will be more efficient given that the 802.11 used by default here does not really support QoS very well.

BIBLIOGRAPHY

- [1] P. M. Ruiz, A. F. Gomez-Skarmeta, Approximating optimal multicast trees in wireless multihop networks, *Computers and Communications*, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on, 27-30 June 2005, pp.: 686 – 691.
- [2] P.M. Ruiz, A. F. Gomez-Skarmeta, Reducing data-overhead of mesh-based ad hoc multicast routing protocols by Steiner tree meshes, *Sensor and Ad Hoc Communications and Networks*, 2004. IEEE SECON 2004, 4-7 Oct. 2004, pp.: 54 – 62.
- [3] J. Biswas, M. Barai, S. K. Nandy, Efficient hybrid multicast routing protocol for ad-hoc wireless networks, *Local Computer Networks*, 2004. 29th Annual IEEE International Conference on, 16-18 Nov. 2004, pp.: 180 – 187.
- [4] Xing Xiong; Uyen Trang Nguyen, Hoang Lan Nguyen, Preemptive Multicast Routing in Mobile Ad-hoc Networks, *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, 2006. ICN/ICONS/MCL 2006. International Conference on, 23-29 April 2006, pp.: 68 – 74.
- [5] Lin Zhang, Dongxu Shen, Xiuming Shan, V.O.K. Li, Yong Ren, A receiver-initiated soft-state probabilistic multicasting protocol in wireless ad hoc networks, *Communications*, 2005. ICC 2005. 2005 IEEE International Conference on, Volume 5, 16-20 May 2005, pp.: 3365 – 3369.

- [6] A. –S. K. Pathan, M.M. Alam, M. M. Monowar, M.F. Rabbi, An efficient routing protocol for mobile ad hoc networks with neighbor awareness and multicasting, E-Tech 2004, 31 July 2004, pp.: 97 – 100.
- [7] Ki-II Kim, Sang-Ha Kim, A novel overlay multicast protocol in mobile ad hoc networks: design and evaluation, IEEE Transactions on Vehicular Technology, Volume 54, Issue 6, Nov. 2005, pp: 2094 - 2101.
- [8] R. Vaishampayan, J. J. Garcia-Luna-Aceves, Robust tree-based multicasting in ad hoc networks, Performance, Computing, and Communications, 2004 IEEE International Conference on, 2004, pp.: 647 - 652.
- [9] L. Klos, G. G. Richard, III , Reliable ad hoc group communication using local neighborhoods, Networking And Communications, Wireless And Mobile Computing 2005 (WiMob'2005), IEEE International Conference on, Volume 3, 22-24 Aug. 2005, pp.: 361 - 368.
- [10] Uyen Trang Nguyen, Xing Xiong, Rate-adaptive multicast in mobile ad-hoc networks, Wireless and Mobile Computing, Networking and Communications, 2005. (WiMob'2005), IEEE International Conference on, Volume 3, 22-24 Aug. 2005, pp.: 352 – 360.
- [11] C. Gomathy, S. Shanmugavel, Design of a priority scheduler using fuzzy logic and the performance analysis with multicast routing protocols, Personal Wireless Communications, 2005. ICPWC2005. 2005 IEEE International Conference on, 23-25 Jan. 2005, pp.: 451 – 455.
- [12] D. Galatchi, R. Zoican, A Multicast Routing Protocol for Multi-Hop Wireless Networks: ODMRP, Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2005. 7th International Conference, Volume 2, 28-30 Sept. 2005, pp.: 563 – 565.

- [13] Chia-Cheng Hu, E. Hsiao-Kuang, Gen-Huey Chen, Bandwidth-satisfied multicast trees in MANETs, IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), Volume 3, 22-24 Aug. 2005, pp.: 323 – 328.
- [14] Jun-Zhao Sun, Mobile Ad Hoc Networking: an Essential Technology for Pervasive Computing, International Conferences on Info-tech and Info-net, Beijing 29 Oct. – 1 Nov., 2001, Proceedings, ICII 2001, Volume 3, pp.: 316 – 321.
- [15] S. Corson, J. Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC2501.
- [16] B. Adamson, Tactical Radio Frequency Communication Requirements for IPng, RFC 1677, August 1994.
- [17] D.P. Agrawal and Q-A. Zeng, Introduction to Wireless and Mobile Systems, Brooks/Cole, 2003.
- [18] IETF Working Group: Mobile Adhoc Networks (manet).
<http://www.ietf.org/html.charters/manet-charter.html>.
- [19] J. Jubin and J.D. Tornow, “The DAPPA Packet Radio Network Protocols”, Proceedings of the IEEE, vol.75, no. 1, Jan. 1987, pp.: 21-32.
- [20] S. Lee, W. Su, and M. Gerla, On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks, Mobile Networks and Applications, 2002, Volume 6, Issue 7, pp.: 441-453.
- [21] S. Lee, W. Su, M. Gerla, and R. Bagrodia, A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols, Nineteenth Annual Joint Conference of the IEEE Computer

and Communications Societies, Proceedings of IEEE INFOCOM 2000, Volume 2, pp.: 565-574.

[22] J. Jetcheva and D. Johnson, Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks, ACM MobiHoc, October 2001, pp.: 33-44.

[23] Seungjoon Lee, Chongkwon Kim, Neighbor supporting ad hoc multicast routing protocol, Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC, 2000 First Annual Workshop on, 11 Aug. 2000, pp.: 37 -44.

[24] J. Luo, P.T. Eugster, J.-P. Hubaux, Route driven gossip: probabilistic reliable multicast in ad hoc networks, INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Volume 3, 30 March-3 April 2003, pp.: 2229 – 2239.

[25] C. W. Wu, Y. C. Tay, AMRIS: a multicast protocol for ad hoc wireless networks, Military Communications Conference Proceedings, 1999, MILCOM 1999. IEEE, Volume 1, 31 Oct.-3 Nov. 1999, pp.: 25 – 29.

[26] C. de Morais Cordeiro, H. Gossain, D.P. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, IEEE Network, Volume 17, Issue 1, Jan.-Feb. 2003, pp.: 52 – 59.

[27] Se-young Lee, Hyeong Soo Chang, An Ant System Based Multicasting in Mobile Ad Hoc Network, The 2005 IEEE Congress on evolutionary Computation, 2005, Volume 2, 2-5 Sept. 2005, pp.: 1583 – 1588.

[28] C. Jaikao, V. Sridhara, Chien-Chung Shen, Energy Conserving Multicast for MANET With Swarm Intelligence, IEEE International Conference on Mobile Ad hoc and Sensor

Systems Conference, 2005, 7-10 Nov. 2005, pp.: 732 - 740.

[29] F. Sato, T. Mizuno, A Route Reconstruction Method Based on Support Group Concept for Mobile Ad Hoc Networks, 19th International Conference on Advanced Information Networking and Applications, 2005. AINA 2005, Volume 1, 28-30 March 2005, pp.: 84 - 89.

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

[31] C.-K. Toh, Ad Hoc Mobile Wireless Networks, Prentice Hall Inc., 2002.

[32] E. Royer and C. Perkins, Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol, Proc. ACM Mobicom '99 Conf., August 1999, pp.: 207-218.

[33] H. Gossain, C. de Morais Cordeiro, and D.P. Agrawal, Multicast: Wired to Wireless, IEEE Communications Magazine, June 2002, pp.: 116 – 123.

[34] K. Obraczka, G. Tsuduk, Multicast routing issues in ad hoc networks, IEEE 1998 International Conference on Universal Personal Communications, 1998. ICUPC '98, Volume 1, 5-9 Oct. 1998, pp.: 751 – 756.

[35] K. Balachandran, S.R. Kadaba, and S. Nanda, Channel Quality Estimation and Rate Adaptation for Cellular Mobile Radio, IEEE JSAC, Volume 17, Issue 7, July 1999, pp.: 1244 – 1256.

[36] Jun-Zhao Sun, Mobile ad hoc networking: an essential technology for pervasive computing, International Conferences on Info-tech and Info-net, 2001, Proceedings. ICII 2001 - Beijing. 2001, Volume 3, 29 Oct.-1 Nov. 2001, pp.: 316 – 321.

- [37] J.J.Garcia-Luna-Aceves and E.I. Madrga, A Multicast Routing Protocol for Ad-hoc Networks, In Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), Volumn 2, 21-25 March 1999, pp.: 784-792.
- [38] M. M. Zonoozi, P. Dassanayake, User Mobility Modeling and Characterization of Mobility Patterns, IEEE Journal on Selected Areas in Communications, Volume 15, Issue 7, Sept. 1997, pp.: 1239 – 1252.
- [39] B. Liang, Z.J. Haas, Predictive Distance-based Mobility Management for PCS Networks, Proceedings of INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, 21-25 March 1999, pp.: 1377 – 1384.
- [40] X. Hong, M. Gerla, G. Pei, and 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 System (MSWiM), August 1999.
- [40] <http://toilers.mines.edu/papers/psgz/models.ps.gz>.
- [41] B. Karp, Geographic Routing for Wireless Networks, PhD thesis, Harvard University, 2000.
- [42] E. Royer, P. M. Melliar-Smith, and L. Moser, An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks, IEEE International Conference on Communications, 2001, ICC 2001, Volume 3, 11-14 June 2001, pp.: 857 – 861.
- [43] Elizabeth M. Royer, Charles E. Perkins, Multicast Ad hoc On Demand Distance Vector (MAODV) Routing, IETF Internet Draft, draft-ietf-manet-maodv-00.txt, July 2000 (Work in Progress).

- [44] Y. Yi, S.-J. Lee, W. Su, M. Gerla, On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks, IETF Internet Draft, draft-ietf-manet-odmrp-04.txt, November 2002.
- [45] P. Sinham, R. Sivakumar, and V. Bharghavan, MCEDAR: Multicast Core-Extraction Distributed Ad Hoc Routing, IEEE Wireless Communication and Net. Conf., Sept. 1999, pp.: 1313-1317.
- [46] H. Lim and C. Kim, Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks, Proc. MSWiM'00, Boston, MA, August 2000, pp.: 61 -68.
- [47] L. Ji, and M.-S. Corson, Differential Destination Multicast: A MANET Multicast Routing Protocol of Small Groups, Proc. Of IEEE INFOCOM'01, Anchorage, Alaska, April 2001, pp.: 1192-1202.
- [48] Tom Goff, Nael B. Abu-Ghazaleh, Dhananjay S. Phatak, and Ridvan Kahvecioglu, Preemptive Routing in Ad Hoc Networks, Proceedings of the 7th annual international conference on Mobile computing and networking, 2001, pp.: 43-52.
- [49] L. Kou, G. Markowsky, and L. Berman, A Fast Algorithm for Steiner Tree, Acta Informatica, No. 15, vol. 2, 1981, pp.: 141-145.
- [50] Yih-Chun Hu and David B. Johnson, Design and Demonstration of Live Audio and Video Over Multihop Wireless Ad Hoc Networks, Proceedings of IEEE MILCOM, 2002.
- [51] C. Gui and P. Mohapatra, Efficient Overlay Multicast for Mobile Ad Hoc Networks, in Proc. IEEE WCNC, New Orleans, LA, Mar. 2003, pp.: 1118-1123.
- [52] Harald Tebbe, Andreas J. Kessler, Pedro M. Ruiz, QoS-aware Mesh Construction to Enhance Multicast Routing in Mobile Ad Hoc Networks, Proceedings of the first

international conference on integrated internet ad hoc and sensor networks InterSense '06, May 2006.

[53] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Capbell, INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad-hoc Networks, Journal Parallel and Distributed Computing, vol. 60 n°4, Apr. 2000, pp.: 374-406.

[54] H.Xiao, K.Chua, W.Seah and A.Lo, A Flexible Quality of Service Model for Mobile Ad-hoc Networks, Proceedings of Vehicular Technology Conference (VTC), Tokyo, Japan, May 2000, pp.: 445-449.

[55] B. Wang and J. C. Hou, Multicast Routing and Its QoS Extension: Problems, Algorithms, and Protocol, IEEE Network, Vol 14, 01/2001, pp.: 22-36.

[56] Kevin Fall and Kannan Varadhan, editors, ns Notes and Documentation, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, January 1999, Available from <http://wwwmash.cs.berkeley.edu/ns/>.

[57] <http://www.monarch.cs.rice.edu/cmu-ns.html>

[58] IEEE Computer Society LAN MAN Standards Committee, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11-1999, The Institute of Electrical and Electronics Engineers, New York, New York, 1999.

[59] ANSI/IEEE Std 802.11, 1999 Edition [ISO/IEC DIS 8802-11], Wireless LAN medium access control (MAC) and physical layer specifications.

[60] G-S. Ahn, A. T. Campbell, A. Veres, L. Sun, Supporting Service Differentiation for Real-Time and Best Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN), IEEE Transactions on Mobile Computing, Volume 1, September 2002, pp.: 192-207.

- [61] Qi Xue, Aura Ganz, Ad Hoc QoS On-demand Routing (AQOR) In Mobile Ad Hoc Networks, Journal of Parallel and Distributed Computing, Volume 63, Issue 2, February 2003, pp.: 154-165.
- [62] Z.-Y. Demetrios, A Glance at Quality of Services in Mobile Ad-Hoc Networks, <http://www.cs.ucr.edu/~csyiazti/cs260.html>, November 19, 2001.
- [63] Amit Bansal, Mandating QoS in Wireless LANs, NewLogic Technologies, http://www.newlogic.com/products/802_11_wireless_abg/mandating_qos_in_wireless_lans.pdf, Nov 2005.
- [64] G.I. Ivascu, S.Pierre and A. Quintero, QoS Support based on a Mobile Routing Backbone for Ad Hoc Wireless Networks, IWCMC'06, July 3-6, 2006, Vancouver, Canada, pp.: 121-126.
- [65] <http://www.isi.edu/nsnam/ns/doc/node216.html>.
- [66] Aura Ganzm Zvi Ganz and Kittu Wongthavarawat, Multimedia Wireless Networks: Technologies, Standards, and QoS, Prentice hall Ptr, Upper Saddle River, NJ, 2004.
- [67] Jagannathan Sarangapani, Wireless Ad Hoc and Sensor Networks: Protocols, Performance, and Control, Boca Raton, Fla.; London: CRC Press, c2007.