RELIABLE MULTICAST EXTENSION
TO IEEE 802.11 IN AD HOC NETWORKS

by

Meihua Zhan
B.Sc., Beijing Institute of Machinery, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

© Meihua Zhan 2003
SIMON FRASER UNIVERSITY
December 2003

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.
APPROVAL

Name: Meihua Zhan

Degree: Master of Science

Title of thesis: Reliable Multicast Extension to IEEE 802.11 in Ad Hoc Networks

Examining Committee: Dr. Uwe Glaesser
Chair

Dr. Joseph G. Peters, Senior Supervisor

Dr. Qianping Gu, Supervisor

Dr. Lou Hafz, SFU Examiner

Date Approved: December 2, 2003
PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project and extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

Reliable Multicast Extension to IEEE 802.11 in AD HOC Networks

Author: _____________________________

(signature)

Meihua Judy Zhan

(name)

Dec 8, 2003

(date)
Abstract

The IEEE 802.11 standard uses channel reservation schemes and acknowledgements (ACKs) to provide reliable Medium Access Control (MAC) layer unicast services. In contrast, 802.11 does not guarantee the reliability of MAC layer broadcast and multicast transmissions. This lack of reliability extends to ad hoc routing protocols, such as DSR and AODV, which depend on broadcast packets to exchange routing information among nodes.

In this thesis, we introduce an efficient and reliable MAC layer multicast/broadcast protocol called SAM (Sequential Acknowledgement Multicast) protocol. We use simulations to investigate the efficiency and reliability of SAM and to compare the performance of SAM to that of other broadcast/multicast protocols. The foundation of SAM is that the multicast receivers send back Clear to Send (CTS) and ACK frames in a predefined order to avoid collisions at the transmitter. During the retransmission phase of the protocol, the sender only needs to retransmit to nodes that failed to send back an ACK. This method releases those nodes which have been unnecessarily forced to keep silent in a multi-hop ad hoc network.

SAM is efficient, reliable, and easy to implement. Most importantly, it is compatible with the IEEE 802.11 standard.
Acknowledgments

I owe deep debt of gratitude to my senior supervisor, Dr. Joseph G. Peters, for patience, guidance, and being an academic role model. His high standards ensures quality work from all his students, and so I thank him for pushing me in the right direction. This thesis would not have been possible without his strong support to me. I am thankful to my supervisor Dr. Qianping Gu, for his instructions which have widened my angle of views and inspired my thoughts in my study and research. My gratefulness also goes to Dr. Lou Hafer for his valuable input on the thesis and for his time as an examiner.

I offer my deepest thanks to Yuanzhu Peter Chen for his constructive discussions and help me to find out the topic of this thesis.

My special acknowledgement goes to my family, in particular my husband and parents, for their endless support and encouragement.

Finally, I wish to thank all who have helped me during my graduate study and research at Simon Fraser University.
Contents

Approval ii
Abstract iii
Acknowledgments iv
Contents v
List of Tables viii
List of Figures ix
Glossary x

1 Overview 1
  1.1 Introduction ......................................................... 1
    1.1.1 Mobile Ad Hoc Networks ..................................... 1
    1.1.2 Multicast ....................................................... 2
    1.1.3 IEEE 802.11 Specification .................................. 3
  1.2 Motivation and Solution ........................................... 6
  1.3 The SAM Protocol, Evaluation and Results ......................... 8
  1.4 Organization of the Thesis ....................................... 9

2 IEEE 802.11 MAC Layer 10
  2.1 MAC Protocols ..................................................... 11
6.3 Future Work

Bibliography
# List of Tables

2.1 Type and Subtype of the Frame Control field ............................................ 19
4.1 A Normal Neighbor Information Table ...................................................... 30
4.2 MAC address of nodes ................................................................. 31
4.3 Neighbor Information Table of Node A ................................................. 31
4.4 Neighbor Information Table of Node D ................................................. 32
5.1 Trend comparison between static and dynamic network (SDR) ............. 56
List of Figures

1.1 IEEE 802.11 and ISO model .................................. 4

2.1 Examples of hidden terminal problem ........................ 13
2.2 Unicast using virtual carrier sense .............................. 16
2.3 IEEE 802.11 MAC frame format ................................. 18

4.1 Multicast transmission of SAM, BMMM* and BMW ............ 28
4.2 The topology of a simple ad hoc network ....................... 31
4.3 Example of retransmission nodes ............................... 35
4.4 Comparison of frame structure between RTM and RTS ........ 36
4.5 SAM multicast using virtual carrier sense ...................... 39

5.1 Successful delivery rate in a static network .................... 50
5.2 Average packet transmission time in a static network ....... 52
5.3 Average time for last ACK in a static network ............... 53
5.4 Successful delivery rate in a dynamic network ............... 55
5.5 Average packet transmission time in a dynamic network .... 58
5.6 Average time for last ACK in a dynamic network .......... 59
5.7 Effect of beacon interval on SDR .............................. 60
5.8 Average moving distance vs SDR ............................... 61
## Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>APTT</td>
<td>Average Packet Transmission Time</td>
</tr>
<tr>
<td>ATIM</td>
<td>Announcement Traffic Indication Message</td>
</tr>
<tr>
<td>ATLA</td>
<td>Average Time for Last ACK</td>
</tr>
<tr>
<td>BM MMM</td>
<td>Batch Mode Multicast MAC</td>
</tr>
<tr>
<td>BMW</td>
<td>Broadcast Medium Window</td>
</tr>
<tr>
<td>CA</td>
<td>Collision Avoidance</td>
</tr>
<tr>
<td>CCK</td>
<td>Complementary Code Keying</td>
</tr>
<tr>
<td>CF</td>
<td>Contention Free</td>
</tr>
<tr>
<td>CS</td>
<td>Carrier Sensing</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Inter Frame Spacing</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Serve</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>IBSS</td>
<td>Independent Basic Service Set</td>
</tr>
<tr>
<td>IFS</td>
<td>Inter Frame Spacing</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>LLC</td>
<td>Logical Link Control</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Adhoc Network</td>
</tr>
<tr>
<td>MGR</td>
<td>Message Generation Rate</td>
</tr>
<tr>
<td>MPDU</td>
<td>MAC Packet Data Unit</td>
</tr>
<tr>
<td>MSDU</td>
<td>MAC Service Data Unit</td>
</tr>
<tr>
<td>NAK</td>
<td>Negative ACK</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>NIT</td>
<td>Neighborhood Information Table</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical</td>
</tr>
<tr>
<td>RA</td>
<td>Receiving Address</td>
</tr>
<tr>
<td>RAK</td>
<td>Request for ACK</td>
</tr>
<tr>
<td>RL</td>
<td>Responding List</td>
</tr>
<tr>
<td>RTM</td>
<td>Request To Multicast</td>
</tr>
<tr>
<td>RTS</td>
<td>Ready To Send</td>
</tr>
<tr>
<td>SAM</td>
<td>Sequential Acknowledgement Multicast</td>
</tr>
<tr>
<td>SDR</td>
<td>Successful Delivery Rate</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Inter Frame Spacing</td>
</tr>
<tr>
<td>TA</td>
<td>Transmitter Address</td>
</tr>
<tr>
<td>URM</td>
<td>Unicast Realized Multicast</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless LAN</td>
</tr>
</tbody>
</table>
Chapter 1

Overview

1.1 Introduction

1.1.1 Mobile Ad Hoc Networks

An ad hoc network is a local area network (LAN) or other small network. A mobile ad hoc network (MANET) consists of a collection of dynamic nodes with unpredictably changing multi-hop topologies that are composed of relatively bandwidth constrained wireless links. In MANETs, there is no assumption of an underlying fixed infrastructure. Nodes can move around arbitrarily. Each mobile node functions as a router to establish connections between any two nodes. Since each node has a limited transmission range, not all messages may reach all the intended receivers. To provide communication through the whole network, source to destination paths can be relayed through several intermediate neighbouring nodes.

In this thesis, it is assumed that the power of all transmitters in an MANET is the same so that they have the identical transmission range. There are two types of models to present MANETs in the literature. The first one considers the geometry of a network. Each node of the MANET is represented by a point in the plane, and each of those points has a region associated with it, often a circle of a given radius centered at this point. If the point representing node \( v \) is within the region associated with node \( u \), we assume that node \( u \) can reach node \( v \). The second type is a graph
model. Nodes of the graph represent nodes of the network and the existence of an undirected edge \((u, v)\) means that node \(u\) and \(v\) can communicate directly with each other. Two nodes are neighbors if they are connected by an edge or they are within the transmission range of each other. In this thesis, both models are used. It is not difficult to convert one model into the other.

A network node in MANETs acts as either a transmitter or a receiver at any given time. It is called a transmitting node when it is actively transmitting. Otherwise, it is called a receiving node.

The emergence and continual growth of MANETs and other wireless LANs are being driven by the need to lower the costs associated with network infrastructures and to support mobile networking applications that offer gains in process efficiency, accuracy, and lower business costs [16]. The majority of applications for the MANET technology are in areas where rapid deployment and dynamic reconfiguration are necessary and the wired network is not available [11]. These include military battlefields, emergency search and rescue sites, classrooms, and conventions where participants share information dynamically using their mobile devices.

### 1.1.2 Multicast

As in traditional wired networks, MANETs support *multicast*, *unicast* and *broadcast*. They are the three primary methods of passing data between nodes in communication networks.

Unicast transmission, the dominant form of transmission on LANs and the Internet, is to send a piece of information from one network node to another network node. Broadcasting is to transmit a message from one point of the network, the source, to all other points. We can say that a unicast is a one-to-one transmission and broadcasts are one-to-all transmissions.

Multicasting is the networking technique of delivering the same packet (datagram) simultaneously to a group of hosts identified by a single destination address. For example, IP multicast groups are identified by special IP addresses. A multicast is a one-to-many transmission. A multicast datagram is typically delivered to all members
of its destination host group with the same reliability as regular unicast datagrams [13].

Multicasting is intended for group-oriented computing. Unlike broadcast transmission in which the receivers are passive, multicast clients receive a stream of packets only if they have previously elected to do so by joining the specific multicast group address. Membership of a group is dynamic and controlled by the receivers; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a multicast group. A host may be a member of more than one group at a time.

In the Internet (IPv4), multicasting is supported by special routers called multicast routers. These routers in a multicast network learn which sub-networks have active clients for each multicast group and attempt to minimize the transmission of packets across parts of the network where there are no active clients.

The use of multicasting within a network has many benefits. Multicasting reduces the communication costs for applications that send the same data to multiple recipients. Instead of sending via multiple unicasts, multicasting minimizes the link bandwidth consumption, sender and router processing, and delivery delay [29]. Multicast transmission with N receivers may provide significant bandwidth savings up to \((N - 1)/N\) of the bandwidth compared to N separate unicasts. In addition, multicasting provides a simple yet robust communication mechanism whereby a receiver's individual address can be unknown or change in a manner that is transparent to the source [23].

1.1.3 IEEE 802.11 Specification

Various wireless technologies, such as IEEE 802.11, Bluetooth, HomeRF etc., are able to build MANETs at present. Considering that most wireless LAN suppliers now have 802.11-compliant products, we focus on the 802.11 based wireless ad hoc networks in this thesis.

In 1997, after seven years of work, the IEEE published 802.11 [3, 4, 5, 14, 26, 40, 41], the first internationally sanctioned standard for wireless LANs (WLANs). In
September 1999, they ratified the 802.11b "High Rate" amendment to the standard, which added two higher speeds (5.5 and 11 Mbps) to 802.11. With 802.11b WLANs, mobile users can get Ethernet levels of performance, throughput and availability. The standards-based technology allows administrators to build networks that seamlessly combine more than one LAN technology to best fit their business and user needs. Like all other IEEE 802 standards, the 802.11 standards focus on the bottom two levels of the ISO model, the physical layer and data link layer (see Figure 1.1). Any LAN application, network operating system, or protocol, including TCP/IP will run on an 802.11-compliant wireless LAN as easily as it runs over Ethernet [31].

![IEEE 802.11 and ISO model](image)

Figure 1.1: IEEE 802.11 and ISO model
CHAPTER 1. OVERVIEW

PHY Layer

The IEEE 802.11 standard for WLANs focuses on the MAC\(^1\) and physical (PHY) layers for access point (AP) based networks and ad hoc networks. The original standard supported three PHY standards: infrared (IR), frequency hopping spread spectrum (FHSS), and direct sequence spread spectrum (DSSS) [3, 26, 40]. The 802.11b extension of the standard supports DSSS in the 2.4-GHz band with data rates of 1, 2, 5.5, and 11 Mbps. Bit rates of the last two are achieved through complementary code keying (CCK) [5, 26, 40]. Besides 802.11b, another extension, 802.11a, is for a high bit rate orthogonal frequency division multiplexing (OFDM) modulation PHY standard providing bit rates in the range of 6 to 54 Mbps in the 5-GHz band [4, 14, 26, 40, 41]. All the PHY layers support the same MAC layer, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [31]. The 802.11b and 802.11a extensions affect only the physical layer. Their basic architecture, features, and services are defined by the original 802.11 standard.

MAC Sublayer

The goal of the 802.11 MAC layer is to provide access control functions for single shared-medium PHY layers in support of the Logical Link Control (LLC) layer. The MAC layer performs the addressing and recognition of frames in support of the LLC. The 802.11 standard uses CSMA/CA, whereas standard Ethernet uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD). An 802.11 WLAN takes measures to avoid collisions but not to detect them because mobile devices cannot accurately judge collisions from signal power in wireless network environment.

All data packets of the network layer are encapsulated in MAC layer data frames, and network layer addresses (IP addresses) are translated to MAC addresses. MAC layer communications are actually one-hop transmissions because the PHY layer has a single shared channel. Thus, a routing scheme is not necessary at the MAC layer.

\(^1\)MAC is one of the sublayers of the Data Link Layer. We use “MAC sublayer” and “MAC layer” interchangeably in this thesis.
CHAPTER 1. OVERVIEW

Hence, MAC layer unicast, MAC layer multicast, and MAC layer broadcast are transmissions from the sender to one of its neighbors, to a subset of its neighbors, and to all of its neighbors, respectively. Since data cannot be delivered to a node more than one hop away by the MAC layer, it is the responsibility of the network layer to build a route in a multi-hop MANET.

In this thesis, we consider the MAC layer broadcast to be a special case of the MAC layer multicast. Although MAC layer unicast could be treated as another special case of our MAC layer multicast, we keep this concept separate because it is used by the 802.11 MAC protocols to avoid the hidden terminal problem. The hidden terminal problem happens when non-neighboring nodes simultaneously transmit to a common neighbor so that the packets collide at the receiver. The hidden terminal problem is described in Chapter 2.

The reliability of data transmissions at the MAC layer is a big concern because of the hidden terminal problem and the vulnerability to errors of the wireless link. Reliability is defined as follows:

Definition 1.1.1 Reliable Transmission

A transmission is called reliable if a sender knows explicitly that either a receiver successfully received the data, or the receiver failed to receive the data after a certain number of retransmissions.

The IEEE 802.11 standard uses a channel reservation scheme and acknowledgement space (ACK) to provide reliable MAC layer unicast services. However, 802.11 does not guarantee the reliability of MAC layer multicast transmissions. In these cases, the multicast sender blindly sends out data frames when it senses a clear channel. The 802.11 MAC layer protocols are described in Chapter 2.

1.2 Motivation and Solution

IEEE 802.11 MAC protocols for MANETs today do not provide reliable multicast/broadcast service. The network nodes blindly send out multicast/broadcast frames
when they sense a clear channel. Thus, the transmitter does not know whether a neighbor fails to receive the data or not.

However, reliable multicast service at the MAC layer in MANETs is highly desirable. Firstly, most of the routing protocols, including unicast routing [21, 28, 30] and multicast routing [7, 8, 9, 15, 17, 19, 20, 32, 43], depend on successful delivery of broadcast messages, although none of them emphasizes the usage of a reliable broadcast. Routing algorithms could discover a route eventually; however, the optimal one may be missed since the 802.11 MAC broadcast is unreliable. The network performance degrades for this reason.

Furthermore, multihop multicast communication in wireless ad hoc networks needs reliable multicast service from the MAC layer. Recently, many network layer multicast protocols have been proposed. They can be largely classified into two categories: tree-based multicast protocols ([7, 18, 32, 43]) and mesh based multicast protocols ([15, 25]). In fact, both approaches have serious problems if the multicast application requires high reliability. Solely working on the network layer is not sufficient to provide highly reliable multicast for wireless networks in an efficient way, so the approach of gaining multicast reliability support from the MAC layer is worth pursuing.

Besides the above two reasons, the motivation for provision of local recovery at the MAC layer is determined by the characteristics of the wireless links. Unlike wired links where data loss and error rate are sufficiently small so that reliability can be achieved in an end-to-end fashion, wireless links can be down and up unpredictably due to the moving of network stations and vulnerability to interference. All of these factors cause significant data loss and errors and prolong the end-to-end reliable transmission. We have noticed that the local retransmission (recovery) mechanism is successfully deployed in wireless unicast transmissions to increase the network efficiency.

Based on the above observations, we propose a new MAC protocol called Sequential Acknowledgement Multicast (SAM) protocol, which offers reliable multicast service using a channel reservation scheme. SAM can coexist with current IEEE 802.11 MAC multicast/broadcast protocols. Please note that SAM considers a broadcast to be a special case of a multicast.
1.3 The SAM Protocol, Evaluation and Results

Protocol

SAM protocol is an addition to the current IEEE 802.11 standard to provide reliable multicast services at the MAC layer. The goal is to realize reliability with relatively small overhead. In a reliable multicast, the transmitter needs an acknowledgement from each receiver of the multicast group to confirm that all members of the group have received the data frame correctly. The transmitter should be able to identify the members that did not acknowledge so that it can retransmit only to them. Furthermore, a channel reservation scheme is needed to deal with the hidden terminal problem in wireless ad hoc networks.

In SAM, the multicast sender should be aware of the receivers among its neighbors. During a multicast session, a network node could be either a multicast receiver or a multicast router, or both. Only the multicast routers at the network layer become multicast senders at the MAC layer. According to the routing protocols, the multicast router may know the explicit or implicit next-hop nodes which are responsible for relaying messages. If a node maintains the exact next-hop nodes information, it sends out data frames at the MAC layer only to those neighbors. Otherwise, all of its neighbors are the MAC layer multicast receivers.

The transmitter in SAM provides data recovery by using positive acknowledgements. The foundation of SAM is that the multicast receivers send back Clear to Send (CTS) and ACK frames in a predefined order, which is determined by the sender, to avoid collisions at the transmitter. During the retransmission phases of the protocol, the sender only needs to retransmit to nodes that failed to send an ACK back.

Evaluation

Simulation is used to compare the performance of SAM with previous research [34, 37]. To evaluate the performance of SAM and other protocols (BMMM*, BMW, URM), we have conducted two series of experiments under static and dynamic network environments. In a static network, the network topology remains constant. The performance
measured in this setup could be regarded as a theoretical value without considering node movement and network overhead such as management frames. Contrarily, we allow network nodes to randomly move with some maximum speed $v_{\text{max}}$ in the dynamic network environment. We believe that the dynamic network environment is a more realistic one to study.

To compare SAM with other protocols, we use the Successful Delivery Rate, Average Packet Transmission Time, and Average Time for Last ACK. These metrics are investigated under various parameters, including different message generation rates, packet sizes, receiver densities, success thresholds, and node moving speeds. Besides, we also study the relationship between the node moving speed and the beacon interval in the SAM protocol.

Results

By two series of experiments, we investigate four MANET protocol schemes: SAM, BMMM*, BMW and URM. We empirically show that SAM generates the best successful delivery rate while maintaining a short average packet transmission time. SAM exceeds all other protocols in both the static and dynamic network environments.

1.4 Organization of the Thesis

The following chapters are organized as follows. Chapter 2 introduces the 802.11 MAC layer frame structure and protocols, including unicast and multicast. In Chapter 3, a brief description is given of the existing MAC layer multicast protocols designed for IEEE 802.11. The main idea, implementation considerations, and the applications of the new protocol are described in Chapter 4. The simulation setup and experimental results are shown in Chapter 5. We conclude in Chapter 6 and suggest future work.
Chapter 2

IEEE 802.11 MAC Layer

An IEEE 802.11 topology consists of components interacting to provide a WLAN that enables station mobility transparent to higher protocol layers, such as the LLC. The 802.11 standard supports two topologies:

- Independent Basic Service Set (IBSS) networks
- Extended Service Set (ESS) networks

ESS networks are WLANs with infrastructure in which the communications among mobile nodes are relayed by Access Points (APs). There is a distribution system defined in the 802.11 standard to connect the APs. A distribution system is a backbone network that is responsible for MAC-level transport of MAC service data units (MSDUs).

An IBSS network is also called an ad-hoc network where a group of stations communicates without the aid of infrastructure (more specifically without APs). Part of the functionality that would be provided by APs, like beacon generation which we describe later, is performed by the end-user stations. Any station can establish a direct communication session with any other station in an IBSS network.
CHAPTER 2. IEEE 802.11 MAC LAYER

2.1 MAC Protocols

In this section, we introduce the 802.11 unicast and multicast protocols. MAC layer transmissions (unicast, multicast and broadcast) are within a one-hop range. The MAC sublayer uses MAC protocols to ensure that signals sent from different stations across the same channel do not collide.

The basic medium access functionality of 802.11 allows interoperability between compatible PHY layers through the use of the CSMA/CA protocol without ACKs and with random backoff time following a busy medium condition. Additionally, all directed traffic (unicast) uses immediate positive ACKs, with the sender scheduling a retransmission if no ACK is received. The 802.11 CSMA/CA protocol was designed to reduce the probability of collisions among multiple stations accessing the medium at the point where collisions would most likely occur. To resolve and minimize conflicts in medium contention, a random backoff arrangement is introduced.

Carrier Sense

For 802.11 WLAN users, carrier sensing is performed both at the air interface and at the MAC sublayer. The former method is referred to as physical carrier sensing, and the latter one is referred to as virtual carrier sensing. Physical carrier sensing detects the presence of other IEEE 802.11 WLAN users by analyzing all detected packets, and also detects activity in the channel via relative signal strength from other sources.

Virtual carrier sensing is performed by source stations based on reservation information found in the Duration field of all frames. This information announces a station’s impending use of the medium to all other stations. The available information in the duration field is used by other stations to adjust their network allocation vectors (NAVs), which indicate the amount of time that must elapse until the current transmission session is complete and the channel can be sampled again for idle status. A station will update its NAV value to be equal to the duration value when it receives any MAC frames, if that value is greater than the current NAV value. The NAV operates like a timer starting with some value and counting down to zero. The channel is virtually idle for a station if its NAV value is 0; otherwise, the channel is
CHAPTER 2. IEEE 802.11 MAC LAYER

virtually busy. The channel is considered to be busy if either the physical or virtual carrier sensing mechanisms indicate that the channel is busy.

Collision Avoidance

The CSMA/CA protocol avoids the probability of collisions among stations by using a random backoff time if the station’s physical or logical sensing mechanism indicates a busy medium. The period of time immediately following a busy medium is when the highest probability of collisions occurs, especially under high utilization. This is because many stations may be waiting for the medium to become idle and will attempt to transmit at the same time. Once the medium is idle, a random backoff time defers a station from transmitting a frame, minimizing the chance that transmissions will collide.

The random backoff procedure works as follows. A station with a frame to transmit initially senses the channel. If found busy, the station will wait until the channel becomes idle for a Distributed Inter Frame Spacing (DIFS) period, and it then computes a random backoff time. In IEEE 802.11, time is slotted in time intervals of length one slot time, $T_{\text{slot time}}$. The slot time is used to define the Inter Frame Spacing (IFS) intervals and determine the backoff time for stations. The slot time is different for each physical layer implementation. An integer number of time slots corresponds to a random backoff time, i.e. $T_{\text{backoff}} = \text{Random(backoff Range)} \times T_{\text{slot time}}$. Stations decrement their backoff timers only after the medium becomes idle, after a DIFS period, and until the medium becomes busy again or the timer reaches zero. If the timer has not reached zero and the medium becomes busy, the station freezes its timer. When the timer is finally decremented to zero, the station transmits its frame after the channel is idle for a DIFS period. If more than one neighboring station decrements to zero at the same time, a collision can hardly be avoided.

Hidden Terminal Problem

Carrier Sensing (CS) and Collision Avoidance (CA) schemes help to reduce the traffic collisions in IEEE 802.11 WLANs. However, they still suffer from the well-known
hidden terminal problem. The hidden terminal problem happens if a transmitter senses the channel to be idle when the intended receiver is actually busy. These simultaneous transmissions from non-neighboring nodes result in a data collision at the receiver.

The hidden terminal problem occurs commonly in single-channel multi-hop networks. Figure 2.1 shows two examples of the hidden terminal problem. All nodes in the figures utilize CSMA/CA medium access control. The nodes where directed links begin and end are the transmitters and intended receivers, respectively. In Figure 2.1(a) Node A can communicate bidirectionally with Node B and Node C. However, Node B and Node C are out of communication range of each other. Because Nodes B and C cannot hear each other during the “listen” phase, they could both send to A simultaneously. Node A would get corrupted data in this case. We say Nodes B and C are “hidden” from each other. There could still be data collision even if one of the hidden nodes does not explicitly transmit to the common receiver. For example, in Figure 2.1(b), Node A is in the transmission range of B and C; Node B and C cannot directly communicate with each other. Node C senses the channel to be idle while transmitting a packet to A. But the channel is busy at A due to the transmission from B to D, and the packet from C cannot be received by A.

Figure 2.1: Examples of hidden terminal problem
2.1.1 Unicast Protocol

As we mentioned previously, unicast transmission in the 802.11 MAC sublayer uses CSMA/CA with positive ACKs. The sender schedules a retransmission if an ACK is not received. By this mechanism, IEEE 802.11 provides reliable unicast service at the MAC layer.

However, the 802.11 WLAN cannot perform optimally because of the hidden terminal problem. Furthermore, a source station cannot detect a collision during transmission. If a collision occurs, the source will continue transmitting the complete MAC packet data unit (MPDU). When the MPDU is large enough, a lot of channel bandwidth will be wasted due to corrupted MPDUs [3, 4, 5, 12, 26]. A channel reservation scheme is deployed using Request To Send (RTS) and Clear To Send (CTS) control frames to avoid the hidden terminal problem and to minimize the amount of bandwidth wasted when collisions occur.

Channel Reservation with RTS/CTS

After the source station successfully contends for the channel, it transmits an RTS control frame. In the RTS frame, the source announces the destination address and the time it intends to occupy the channel. A station that is addressed by an RTS frame will transmit a CTS frame after a SIFS period if the NAV at the station receiving the RTS frame indicates that the medium is idle. If the NAV at the station receiving the RTS shows the medium is not idle, that station does not respond to the RTS frame. The 802.11 MAC protocol will hold off transmission of any frames until the NAV timer expires even though the physical channel assessment determines there are no transmissions taking place on the medium [16]. The destination specifies the time in its duration field that is needed to complete a transmission.

After the exchange of the RTS and CTS frames, the channel is reserved for the source and destination to use. All the neighbors of the source and the destination keep silent during their transmitting time. The neighbors do the following update on receiving the control frames: After hearing the RTS frame, all the neighbor stations
of the source except the destination read the duration field and set their NAVs accordingly. Similarly, all stations except the source hearing the CTS packet check the duration field and also update their NAVs.

The NAV mechanism reduces the probability of a collision in the receiver's area by a station that is "hidden" from the transmitter for the short duration of the RTS transmission because the station hears the CTS and "reserves" the medium as busy until the end of the transaction. The duration information in the RTS also protects the transmitter's area from collisions during the ACK [1].

Figure 2.2 shows the transmission of an MPDU between two stations, and the NAV settings of their neighbors. When the source node detects an idle channel, it continues to monitor the channel for a DIFS time. If the channel remains idle during this period, the source node sends an RTS frame with the duration field set properly. If the intended receiver observes an idle medium and it remains idle for a SIFS time, it returns a CTS frame with the duration field set appropriately. All neighbors of the source node except the intended receiver will update their NAVs to a period of time indicated in the RTS duration field. The value of the duration field is the amount of time the sending station needs to transmit the data frame, plus one CTS frame, plus one ACK frame, plus three SIFS intervals. Similarly, the neighbors of the destination node excluding the source node will update their NAVs to the time indicated in the CTS duration field. This time is the amount of time from the duration field of the previous RTS frame, minus the time required to transmit the CTS frame and one SIFS interval. If no new updates are received, the NAV timers of all the source's and destination's neighbors will count down to zero after the ACK is sent.

The RTS/CTS operation provides much better performance than the basic access mechanism when there is a high probability of hidden stations. In addition, the performance of RTS/CTS degrades more slowly than basic access when network utilization increases [16].
2.1.2 Multicast Protocol

Like the unicast protocol, the IEEE 802.11 multicast protocol uses CSMA/CA. However, the multicast protocol does not use an RTS/CTS exchange between the transmitter and receivers, and no ACKs are transmitted by any of the recipients of the frame.

The lack of RTS/CTS frames in the 802.11 multicast protocol results in more bandwidth wasted due to the collisions of data frames in a multicast than in a unicast. We know that a sender cannot detect a collision when sending out packets. Even if a collision occurs, the sender will continue transmitting the current MPDU. When the MPDU is large, a lot of channel bandwidth will be wasted [3, 4, 5, 12, 26]. To avoid or minimize the amount of bandwidth wasted when collisions occur, a network node reserves channel bandwidth prior to the transmission of an MPDU by using RTS and CTS control frames in unicast traffic. When compared to the size of a data frame (2346 bytes maximum), the relatively small sizes of RTS and CTS frames (RTS and CTS are 20 and 14 bytes long, respectively) make them ideal to minimize the bandwidth waste.

Another drawback of 802.11 multicast protocol is that no MAC-level recovery
mechanism is provided for failed multicast transmissions. Hence, multicast transmissions are less reliable than unicast ones.

2.2 MAC Frame

2.2.1 MAC Frame Types

To carry out the delivery of MSDUs between peer LLCs, the MAC layer uses a variety of frame types, each having a particular purpose. The IEEE 802.11 specification divides MAC frames into three broad categories that provide management, control, and data exchange functions.

The 802.11 management frames enable stations to establish and maintain communications. One of the management frames we emphasize in this thesis is the beacon frame. The beacon frame provides the “heartbeat” of a wireless LAN, enabling stations to establish and maintain communications. All stations periodically send beacons for synchronization purposes in an ad hoc network. The beacon interval, a specific field in the beacon frame, defines the amount of time between beacon transmissions. However, beacons must be sent using the CSMA/CA mechanism. If another station is sending a frame when the beacon is to be sent by a station, this station must wait. As a result, the actual time between beacons may be longer than the beacon interval.

The 802.11 control frames assist in the delivery of data frames between stations. The common 802.11 control frames include RTS, CTS, and ACK frames. An RTS frame is sent from a station to a particular receiving station to negotiate the sending of a data frame. After receiving an RTS, the destination sends a CTS frame to acknowledge the right for the sending station to send data frames. An ACK frame is sent to the sending station from the destination after receiving an error-free frame to acknowledge the successful reception of the frame.

The main purpose of data frames is to carry information to the destination stations.
2.2.2 MAC Frame Structure

The IEEE 802.11 standard specifies an overall MAC frame format, as shown in Figure 2.3. This frame structure is found in all frames that stations transmit, regardless of frame type. The frame body (MSDU) is a variable length field consisting of the data payload. Source and destination stations are identified in the IEEE 802.11 standard using 48-bit MAC addresses. The two octets for the duration field indicate the time that the channel will be allocated for a successful transmission of a MAC protocol data unit (MPDU). The frame control field carries control information being sent from station to station. The type bits in the frame control field identify the frame as control, management, or data. The subtype bits define the function of the frame. Table 2.1 lists the possible values of type and subtype.

![Figure 2.3: IEEE 802.11 MAC frame format](image-url)
<table>
<thead>
<tr>
<th>Type (b3 b2)</th>
<th>Type Description</th>
<th>Subtype (b7 b6 b5 b4)</th>
<th>Subtype Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Management</td>
<td>0000</td>
<td>Association Request</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0001</td>
<td>Association Response</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0010</td>
<td>Reassociation Request</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0011</td>
<td>Reassociation Response</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0100</td>
<td>Probe Request</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0101</td>
<td>Probe Response</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>0110-0111</td>
<td>Reserved</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>1000</td>
<td>Beacon</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>1001</td>
<td>ATIM</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>1010</td>
<td>Disassociation</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>1100</td>
<td>Deauthentication</td>
</tr>
<tr>
<td>00</td>
<td>Management</td>
<td>1101-1111</td>
<td>Reserved</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>0000-0001</td>
<td>Reserved</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>0110</td>
<td>PS-Poll</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>1011</td>
<td>RTS</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>1100</td>
<td>CTS</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>1101</td>
<td>ACK</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>1110</td>
<td>CF End</td>
</tr>
<tr>
<td>01</td>
<td>Control</td>
<td>1111</td>
<td>CF End + CF-ACK</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0000</td>
<td>Data</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0001</td>
<td>Data + CF-ACK</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0010</td>
<td>Data + CF-Poll</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0011</td>
<td>Data + CF-ACK + CF-Poll</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0100</td>
<td>Null Function (no data)</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0101</td>
<td>CF-ACK (no data)</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0110</td>
<td>CF-Poll (no data)</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0111</td>
<td>CF-ACK + CF-Poll(no data)</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>1000-1111</td>
<td>Reserved</td>
</tr>
<tr>
<td>10</td>
<td>Data</td>
<td>0000-1111</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Table 2.1: Type and Subtype of the Frame Control field
Chapter 3

Related Work

In this chapter we firstly review the network layer multicast routing protocols. Then, we describe the MAC layer multicast protocols previously designed for the 802.11 standard, which are claimed to provide reliability. Even though all of these MAC layer protocols extend the 802.11 multicast/broadcast protocols in some way and provide more reliable services, only a few of them can deliver full reliability.

3.1 Network Layer Multicast Protocols

The Multicast routing problem is another active research area in MANETs. Proposed and developed multicast routing algorithms include AMRIS [43], AMRoute [7], MAODA [32], LAM [20], ODMRP [17], CAMP [15], FGMP [9], DDM [19], and LGT [8]. All of these multicast routing protocols can be classified into 3 strategies: (A) by broadcast, (B) by multicast, and (C) by unicast.

Flooding, an example of Strategy A, is the most straightforward way to provide multicast. In this approach, every network node broadcasts the received data packets to the immediate neighbor nodes exactly once. Only the intended multicast receivers will accept the packets. Every node is a multicast router from this point of view. Another protocol, FGMP, floods the data packets within selected forwarding groups (FG) [9]. The function of FGs is similar to that of the multicast routers.

In Strategy B, a multicast routing protocol either forms a tree (AMRIS, MAODA,
CHAPTER 3. RELATED WORK

LAM), a mesh (ODMRP, CAMP, FGMP), or a hybrid of a tree and a mesh (AM-Route) between multicast senders and receivers. No matter what routes are built, the multicast routers have explicit knowledge of the next-hop nodes for which they are responsible for forwarding the data packets. When a multicast router receives the data packets, it needs to send out the packets only once.

The multicast routing protocols that use Strategy C (DDM, LGT) utilize the unicast routing schemes to avoid the overhead of maintaining the routing tables for multicast. So, the same data packets are sent out more than once at each router. Note that the unicast routing table is still needed at each node.

3.2 Unreliable MAC Layer Multicast Protocols

Reliable MAC layer multicast is an important communication paradigm and is getting increasing attention from the mobile and wireless network research communities. In [6], it is simply suggested that the sender transmits an RTS frame immediately followed by the data frame. This mechanism can reduce collisions, but it still suffers from the hidden terminal problem because the receivers do not use CTS frames to reserve the medium. More importantly, without ACK frames the source node does not know which neighbors have successfully received the data frame.

As mentioned earlier, the IEEE 802.11 standard does not include positive acknowledgements for multicast. To increase reliability, the Broadcast Acknowledgement Scheme in [33] requires that all receivers acknowledge the receipt of a data frame during the DIFS period that immediately follows the data transmission. The 50 $\mu$s DIFS is divided into slots that are several bits long. Each receiver randomly selects a slot and sends a response pattern. When $x$-bit slots are used in an $m$-Mbps wireless LAN, the number of slots is $m(DIFS-SIFS)/x$. Correct implementation of this protocol could be difficult because it requires nodes to transmit response patterns without overlap in short ($x$ bits) consecutive slots. However, even if this implementation problem can be solved, the protocol cannot guarantee reliability because it cannot avoid collisions of response patterns when two or more receivers select the same slot. The Broadcast Acknowledgement Scheme provides a recovery mechanism according to the number of
received response patterns. In the case that the number of received response patterns is less than the number of its neighbors, the sender needs to broadcast again. The sender may fail to collect sufficient response patterns no matter how many rebroadcasts it attempts. A retry threshold is proposed in [33] to minimize the bandwidth wastage. Accordingly, a node will rebroadcast the frame until either the number of response patterns is sufficient or the retry count reaches the threshold.

Several protocols use RTS/CTS mechanisms or similar approaches in addition to ACKs to provide efficient transmission with improved reliability [10, 24, 34, 35, 36, 37, 38].

The Leader-Based Protocol (LBP) [24] involves the election of one of the multicast group members (receivers) as a “leader” or representative for the purpose of sending feedback to the sender. The leader and the other nodes in the multicast group perform complementary actions. When a node receives an RTS and is not ready to receive data, it sends a Not Clear To Send (NCTS) frame if it is not the leader and it does not respond if it is the leader. Similarly, when a data frame is correctly received, the leader sends an ACK and non-leaders do not respond. When a data frame is not received correctly, the leader does not respond and non-leaders send negative acknowledgements (NAKs). Unfortunately, LBP cannot guarantee reliability because it cannot guarantee that every node in a multicast group correctly receives an RTS frame and responds accordingly. Furthermore, the correct operation of LBP requires that all control packets can be transmitted without errors. Suppose that the leader and all but one of the other nodes in the multicast group receive a data frame correctly. The leader will send an ACK to the sender and the node that did not receive the frame correctly should send a NAK to the sender. The ACK and NAK will collide at the sender so that the sender knows that some receiver did not receive the data correctly and retransmits it. However, if the NAK is lost, the sender will incorrectly assume that the data frame was successfully transmitted to all the members of the group.

Like LBP, the Robust Broadcast Protocol (RBP) [38] selects a representative called the collision detector for a multicast group. The collision detector is responsible for detecting collisions and sending this information to the transmitter. The transmitter sends an RTS to the collision detector and relies on its response to determine whether
or not to send a data frame. Since the collision detector’s knowledge of all neighbors of the transmitter may be incomplete, the reliability of the protocol cannot be guaranteed.

**ABROAD** is a *time division multiple access* (TDMA) MAC protocol [10] that uses a CSMA/CA based contention protocol in each slot of a TDMA allocation protocol. In each time slot, *Request to Broadcast* (RTB) and *Clear to Broadcast* (CTB) frames are used to coordinate channel reservations. This new pair of control frames is similar to the CTS/RTS pair used in reliable unicast. ABROAD does not coordinate the transmission of CTB frames by receivers. If two or more nodes send CTB frames simultaneously, they will collide at the sender. The authors of ABROAD assume that a node is capable of detecting an idle channel, a successful packet transmission, or a packet collision. The sender can detect a collision of CTBs, but it is not able to avoid the collisions. So, ABROAD cannot guarantee reliability.

In the *Broadcast Support Protocol* (BSP) [35], a node that receives an RTS immediately returns a CTS if it determines that the channel is idle. It then waits for the data frame. Clearly, this protocol is affected by the hidden terminal problem because CTS frames could collide at the sender. Furthermore, a sender cannot be certain that data frames have been successfully transferred to all of its neighbors because the receivers do not return ACKs or other kind of acknowledgements. The *Broadcast Support Multiple Access* (BSMA) protocol [36] augments BSP with NAK frames to address the problem with acknowledgements. However, BSMA does not coordinate the transmission of either CTS frames or NAK frames, so it is still affected by the hidden terminal problem. Furthermore, if a receiver transmits a NAK frame that is then lost, the sender will not know that it has failed to deliver the data packet successfully to all receivers. Both BSP and BSMA are unreliable because a sender cannot be certain that every intended receiver has received the data [34]. BSP and BSMA are investigated in detail in [34].
3.3 Reliable MAC Layer Multicast Protocols

None of the multicast protocols described in the previous section are reliable because none of them can determine the exact number of neighbors that have received the data frame successfully. The Broadcast Medium Window (BMW) [37] and Batch Mode Multicast MAC (BMMM) [34] protocols provide reliable multicast services by using neighborhood knowledge.

BMW

The BMW protocol uses a reliable unicast protocol, which is a modified version of the 802.11 unicast protocol, to transmit packets to the intended recipients in a round robin fashion. In particular, BMW adds more fields to the RTS and CTS frames to accommodate sequence numbers. This increases the transmission time of RTS and CTS frames in BMW. The main idea of BMW is to send a multicast frame to only one receiver by a unicast. Other intended recipients can “overhear” the unicast and receive the frame. When the sender sends a new multicast frame, it chooses the next receiver in a round robin fashion and unicasts the frame to it. Other recipients still “overhear” and receive the frame. During the unicast, the receiver can tell the sender if any previous frames are missing. If the receiver has missing frames, the sender will retransmit them. When there are no more new frames to send, BMW uses a timer to prevent the round robin process from halting. If the queue of frames to be sent is empty for the time equal to this timer, the next node in the neighbor list will be chosen and the round robin process continues. If all the neighbors are visited in the round robin process and there is still no new packet to transmit, the round robin process stops.

The main disadvantage of BMW is its inefficiency: to receive all acknowledgments, a sender with \( n \) neighbors requires at least \( n \) contention periods and time to transmit the associated control frames (RTS/CTS/ACK). It is very important for routing messages to find out in the shortest possible time which nodes failed to receive a data frame and then to retransmit to it. It is well known that the on-demand network layer routing protocols, such as DSR and AODV, introduce long delays in
the route discovery phase. The route discovery delay can be even longer: when the routing request message is lost at some node that is on a shorter path, a longer route will be chosen. Moreover, since a longer route consumes more power, it is against the general rule of reducing the power usage in a MANET. Further, it is possible that the BMW sender fails to gain control of the channel and thereby interrupts and prolongs the ongoing multicast process [34]. The study in [34] shows that BMW incurs higher overhead and a lower successful delivery rate than BMMM.

**BMMM & LAMM**

BMMM needs one extra control frame, *Request for ACK* (RAK). To avoid the collisions among CTS and ACK frames, the sender provides some simple coordination among the intended receivers. When the sender wins access to the channel, it uses $n$ RTS frames sequentially to instruct each intended receiver to transmit a CTS frame. After the $n$ pairs of RTS/CTS transmissions, the data frame is sent out. After the transmission of the data frame, the sender uses the RAK frames to sequentially instruct each intended receiver to transmit an ACK. Unfortunately, BMMM suffers from implementation issues and cannot be compatible with the current IEEE 802.11 specification. BMMM would work properly if the sender could manage its receivers to transfer $n$ pairs of RTS/CTS frames. The fact is that during the transmission of RTS/CTS between the sender and the first receiver, all other receivers will update their NAV and remain “silent” during the whole process of the broadcast. Hence, the sender will not get any CTS or ACK response from the other $(n-1)$ receivers. If the NAV mechanism is disabled for the broadcasting purpose, the normal unicast cannot function properly because RTS/CTS can no longer reserve the channel. In short, the multiple pairs of RTS/CTS create a dilemma in implementation. So, BMMM can at most guarantee that the first node will receive the data if it follows the IEEE 802.11 standard.

The authors in [34] further proposed a *Location Aware Multicast MAC* (LAMM) protocol based on BMMM. LAMM is still unreliable not only because it is based on an unreliable protocol BMMM, but also because the assumption that it depends on is
doubtful. The assumption is that if the cover set of a sub-set of all the receivers equals the cover set of all the receivers, then all the receivers are guaranteed to receive the data successfully if the sender receives ACKs from all nodes in the sub-set. Based on this assumption, it is only necessary for the sender to coordinate the control frames with the nodes in the sub-set. However, the authors did not consider the situation that some receivers outside the sub-set could not be ready to receive while all the nodes in the sub-set have successfully sent CTSs. In this circumstance, the sender will be fooled.

**BMMM***

In this thesis, BMMM is modified to provide better performance (than the original one does) for the purpose of comparison. We call this modified protocol BMMM*. In BMMM*, upon receiving an RAK frame, a receiver will compare the Transmitter Address (TA) in the last received broadcast packet and the current RAK frame (RAK is also modified to contain the TA address). If two frames originate from the same source node, the receiver responds with an ACK no matter what value the NAV has. This modification makes the following assumption: if the RAK and data frames originate from the same station, the NAV of the receiving station is probably updated by that station too. By deploying BMMM*, the sender could collect more than one ACK frame without retransmissions. However, BMMM* abandons the channel reservation scheme of 802.11. If the NAV timer of a receiver $A$ is not updated by the broadcast sender $S$ but by another station $B$, the ACK frame from $A$ will collide at $B$ since $B$ is either sending or receiving frames. The improvement of BMMM* is achieved by sacrificing the surrounded traffic.
Chapter 4

SAM Protocol and Applications

The goal of our research is to propose a multicast protocol to provide a reliable multicast MAC service with relatively small overhead for IEEE 802.11 based MANETs. This protocol should be able to deliver broadcast frames reliably as well. Since there is a group of receivers in a multicast session, in order to determine whether data frames are correctly received by every receiver, the transmitter requires acknowledgments from each node without collisions. The transmitter should be able to identify unacknowledged nodes so that it can only retransmit data to them. Furthermore, a channel reservation scheme should be utilized to reduce the hidden terminal problem in a MANET. The protocol should be compatible with the current multicast/broadcast approach.

The SAM (Sequential Acknowledgement Multicast) protocol is proposed to satisfy the above requirements. By adopting SAM, a multicast sender is able to coordinate the order in which multicast receivers return CTS and ACK frames. This order coordination avoids collisions of CTS and ACK frames at the sender side. During the retransmission phase, the sender only needs to retransmit to nodes that failed to send back an ACK.
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

4.1 Protocol Design

As mentioned in Chapter 2, the channel for a unicast transmission is reserved by the exchange of RTS/CTS frames between the sender and the receiver. In the design of SAM, we use a similar approach to reserve the channel for reliable multicast services. Before the real data frame is sent over the air, the sender and all of the receivers announce that they are going to occupy the channel for a period of time stated in the duration field, hence other adjacent nodes are kept “silent” during the multicast period. The announcement frame for a sender is RTS while for a receiver it is CTS.

The basic idea of our SAM protocol is to reduce the control frames as much as possible. To initiate a multicast transmission, the sender first broadcasts an RTM (a special kind of RTS), and each receiver of the multicast group responds with a CTS in a predefined order. Similarly, the ACKs from receivers are sent back in the same order. (We will discuss how to determine the transmission order in Section 4.1.2.)

Figure 4.1 compares the transmission processes of SAM, BMMM* and BMW to multicast one data packet to n receivers in the best-case scenario. In BMMM*, the
channel reservation is achieved by exchanging \( n \) pairs of RTS/CTS, one between the sender and each receiver. After transmission of the data, the sender needs to poll each receiver to determine whether it has received the packet, which requires \( n \) pairs of RAK/ACK. In contrast, our protocol SAM only airs 1 RTM (RTS) frame per multicast and only \( n \) CTSs and \( n \) ACKs are needed. Our strategy saves the time to transmit \((n - 1)\) RTSs and \( n \) RAKs. The saving is almost proportional to \( n \). For BMW, if there is only one broadcast packet to be sent out, the transmitter needs to coordinate ACKs from each receiver. According to BMW, the sender receives an ACK from at most one receiver after it transmits a packet. Even though all the destination nodes have received the broadcast packet correctly in the best-case scenario, the sender does not have knowledge of this. To collect the second ACK, the sender has to broadcast the data packet again after the timer times out. So, the total time to obtain \( n \) ACKs consists of the time to transmit one packet \( n \) times plus \((n - 1)\) \( T_{\text{timeout}} \). The transmission times of the above 3 schemes are given as follows:

Let \( T' = T_{\text{contention}} + n(T_{\text{CTS}} + T_{\text{ACK}}) \), then

\[
T_{\text{SAM}} = T' + T_{\text{RTM}} + T_{\text{DATA}} + (2n + 1)T_{\text{SIFS}}
\]

\[
T_{\text{BMW}} = T' + nT_{\text{RTS}} + T_{\text{DATA}} + nT_{\text{RAK}} + 4nT_{\text{SIFS}}
\]

\[
T_{\text{BMW}} = T' + (n - 1)(T_{\text{contention}} + T_{\text{timeout}} + T_{\text{DIFS}}) + n(T_{\text{RTS}} + T_{\text{DATA}} + 3T_{\text{SIFS}})
\]

In Section 4.1.1, we demonstrate the method to broadcast and gather neighborhood information for the determination of the transmission order in a multicast group. In Section 4.1.2, we describe how to modify the RTS frame to contain the transmission order information. Section 4.1.3 shows how to perform a multicast using sequential transmission. A complete SAM protocol is given in Section 4.2. Finally, we discuss the applications of SAM in Section 4.3.
4.1.1 Neighborhood Information

Neighborhood Information Table

In a MANET, mobile nodes usually need to maintain neighborhood information which is important to routing. Provided that all nodes have the same transmission range, if node A is within the transmission range of node B, they are neighbors to each other. In other words, if two nodes can “hear” each other, they are neighbors.

Table 4.1: A Normal Neighbor Information Table

<table>
<thead>
<tr>
<th>Neighbor Sequence</th>
<th>Neighbor’s MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FF-FF-FF-FF-FF-02</td>
</tr>
<tr>
<td>1</td>
<td>FF-FF-FF-FF-FF-03</td>
</tr>
<tr>
<td>2</td>
<td>FF-FF-FF-FF-FF-04</td>
</tr>
<tr>
<td>3</td>
<td>FF-FF-FF-FF-FF-01</td>
</tr>
</tbody>
</table>

The neighborhood information can be stored in a data structure called Neighborhood Information Table (NIT). As shown in Table 4.1, the $i^{th}$ row of the table contains the MAC address\(^1\) of the $i^{th}$ neighbor. We define $i$ to be the Neighbor Sequence. Generally, we have this definition:

**Definition 4.1.1 Neighbor Sequence**

In Node $X$’s NIT, if Node A occupies the $i^{th}$ row, then Node A’s Neighbor Sequence at Node X is $i$.

Node A is called the $i^{th}$ neighbor of Node X.

In our SAM protocol, we add one more attribute to each record: Neighbor($X, i$) (see Table 4.3). Neighbor($X, i$) is the neighbor sequence of Node X in its $i^{th}$ neighbor’s NIT. The formal definition is given below:

**Definition 4.1.2 Neighbor($X, i$)**

\(^1\)MAC addresses are 12-digit hexadecimal numbers (48 bits in length).
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

If Node A is the $i^{th}$ neighbor in Node X's NIT and Node X is the $j^{th}$ neighbor in Node A's NIT (i.e. Node X's neighbor sequence in Node A is $j$), then we define $\text{Neighbor}(X, i) = j$.

Figure 4.2: The topology of a simple ad hoc network

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Node's MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>FF-FF-FF-FF-FF-00</td>
</tr>
<tr>
<td>B</td>
<td>FF-FF-FF-FF-FF-01</td>
</tr>
<tr>
<td>C</td>
<td>FF-FF-FF-FF-FF-02</td>
</tr>
<tr>
<td>D</td>
<td>FF-FF-FF-FF-FF-03</td>
</tr>
<tr>
<td>E</td>
<td>FF-FF-FF-FF-FF-04</td>
</tr>
</tbody>
</table>

Table 4.2: MAC address of nodes

<table>
<thead>
<tr>
<th>Neighbor Sequence ($i$)</th>
<th>Neighbor's MAC address</th>
<th>Neighbor(A, i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FF-FF-FF-FF-FF-02 (C)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>FF-FF-FF-FF-FF-03 (D)</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>FF-FF-FF-FF-FF-04 (E)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>FF-FF-FF-FF-FF-01 (B)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.3: Neighbor Information Table of Node A

Figure 4.2 shows the topology of a small ad hoc network. The MAC addresses of nodes are listed in Table 4.2. The NITs of Node A and Node D are given in Table 4.3 and Table 4.4, respectively. From this example, Node B, C, D and E are the $3^{rd}$,
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

<table>
<thead>
<tr>
<th>Neighbor Sequence (i)</th>
<th>Neighbor’s MAC address</th>
<th>Neighbor(D, i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FF-FF-FF-FF-FF-02 (C)</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>FF-FF-FF-FF-FF-04 (E)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>FF-FF-FF-FF-FF-00 (A)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.4: Neighbor Information Table of Node D

$0^{th}$, $1^{st}$ and $2^{nd}$ neighbors of Node A, respectively. Meanwhile, Node A, C and E are the $2^{nd}$, $0^{th}$ and $1^{st}$ neighbors of Node D, respectively (Node B is not a neighbor of Node D). Since Node A is the $2^{nd}$ node in Node D’s NIT, we have Neighbor(A, 1)=2 in Node A’s NIT.

For neighbors to know about each other’s existence, mobile nodes usually broadcast beacon messages periodically. A beacon message contains the neighbor sequence information in the NIT$^2$ of the broadcaster. When a neighbor B receives a beacon message from A, it performs the following update:

1. If Node A is not in its NIT, Node B adds Node A into its NIT;

2. If the NIT in the beacon message indicates that Node B is the $i^{th}$ neighbor of Node A, Node B updates its NIT for the record corresponding to Node A. Suppose A is the $j^{th}$ neighbor in B’s NIT, B will set the value of Neighbor(B, j) to $i$.

Besides the beacon message, there is another method that allows a node to update its NIT. In a MANET, all network nodes keep track of their neighbors by monitoring frames sent by others. Examples of these frames are RTS, CTS, DATA and ACK. When Node A receives any frame from Node B and Node B is not in A’s NIT, Node B is considered to be a new neighbor to Node A. We call this process joining. As Node B joins Node A’s neighborhood, Node A assigns it a neighbor sequence $k$ and adds a new record for Node B in its NIT. The field Neighbor(A, $k$) is blank at this moment. It will be updated when A receives a beacon message from B.

$^2$The Neighbor(X,i) attribute of a NIT is not sent in the beacon message.
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

Maintaining Neighborhood Relationship

To maintain the neighborhood relationship between two nodes, they must "hear" from each other once in a while. If Node A has not received any beacon messages or other frames from Node B for a time period, \( T_{\text{exp}} \), the record of Node B in A's NIT is out-dated. At this moment, Node A expires Node B's entry in its NIT, i.e. Node A no longer considers B to be its neighbor.

In addition to neighborhood information, we need to keep neighbor sequences up-to-date in all nodes. After neighbor sequences are broadcast in a beacon message, they are valid for a specific time period called a lease period, \( T_{\text{lease}} \). The lease period is renewed after each broadcast if two nodes remain neighbors. If they are out of contact range and the lease period elapses, the neighbor is marked as expired. The neighbors obtain their neighbor sequences in the current node on a First Come First Served (FCFS) basis. A new neighbor is always given the smallest unused neighbor sequence if one is available; the oldest expired neighbor sequence is assigned otherwise. This scheme will never reallocate a neighbor sequence number during its lease period. Generally, we should ensure that the neighborhood relationship expires earlier than the lease period. This strategy avoids assigning the same neighbor sequence to more than one node. It also ensures that we can assign the same neighbor sequence to oscillating nodes which move in and out of the neighbor range frequently. In our experiment, we set \( T_{\text{lease}} = 2T_{\text{exp}} \).

The time interval \( T_{\text{int}} \) for broadcasting beacon messages has an important impact on the NIT. The value of \( T_{\text{int}} \) and \( T_{\text{exp}} \) should be carefully chosen. If \( T_{\text{int}} \) is too long compared to \( T_{\text{exp}} \), neighbor sequences will be improperly marked expired. If \( T_{\text{int}} \) is too short compared to \( T_{\text{exp}} \), frequent beacon updates consume too much network bandwidth. In our experiments, we set \( T_{\text{exp}} = 4T_{\text{int}} \). Another parameter that will affect the NIT is the speed \( v \) at which the mobile node travels. Intuitively, the faster the nodes move, the shorter the broadcast interval should be. Experimental study of the relationship between \( T_{\text{int}} \) and \( v \) is described in Chapter 5.
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

4.1.2 Request To Multicast (RTM) frame

In the preceding section, we discussed an approach to establishing neighbor sequences among neighbors of a node. Our SAM protocol will utilize it for a multicast.

To achieve better performance for a multicast, we can reduce the overhead of control frames occupying the channel. A good method is to let the sender and each of the receivers send at most one control frame to reserve the channel per multicast session. The receivers can use CTS frames to perform the above task. However, an RTS frame is not suitable for the sender to use. If an RTS frame were used by the sender, the destination address in the frame would be either a unicast address (of one of the receivers) or the multicast address. If a unicast address were chosen, only the specific destination would answer with a CTS. To collect all the CTSs from all the receivers, the sender would need to send more than one RTS (e.g. in the case of BMMM*), which violates the goal of reducing overhead. If a multicast address were used, the CTS frames would collide at the sender because all the receivers would answer the RTS simultaneously. Even though the receivers could answer with CTS frames in some predefined order, any retransmissions have to be directed to all of the receivers if any one of the receivers fails to receive the data frame. These unnecessary retransmissions (to the receivers which received the data frame correctly in the first round) increase the bandwidth occupation and degrade the network performance.

Instead of using an RTS frame, a special control frame RTM (Request To Multicast) is introduced in SAM to determine the order in which receivers reply with CTS or ACK frames. On receiving an RTM frame, every neighbor of the transmitter must firstly check whether it is a receiver of this multicast. If it is, it replies with a CTS according to an order specified in the RTM. After the sender has collected all CTSs, it starts to deliver the data frame. Upon receiving the data, receivers send ACKs back in the same order as they sent CTSs. The time to multicast a data frame in an ideal scenario is at least the total time of transmitting all the above-mentioned frames. Additionally, the time includes all of the SIFS intervals. To be compatible with IEEE 802.11, there is a SIFS time interval between any two frames in SAM. (See to Figure 4.1 and Equation 4.1.)
RTM is a newly proposed control frame and its structure follows the IEEE 802.11 control frame specification (See Chapter 2). The multicast sender specifies whether a neighbor is a receiver or not in the RTM frame. One of the advantages of specifying receivers is that the sender can retransmit only to those nodes which failed to return an ACK. This method saves bandwidth compared to retransmitting to all multicast receivers. Moreover, it frees the nodes that are two hops away from the sender to do other transmissions (and they do not need to update their NAV timers). Figure 4.3 gives an example. Node A is a sender of a multicast transmission, and Node B is one of its receivers. During the transmission, Node A receives an ACK back from Node B. If Node B is included in the retransmission (to other nodes), Node C, a neighbour of Node B, will be forced into "silence" during this retransmission period. This mechanism unnecessarily prevents Node C from transmitting or receiving packets. On the contrary, our SAM protocol does not include Node B in the retransmission. Hence, Node C can receive or send packets concurrently with Node A's retransmission.

On receiving an RTM frame, a mobile station responds with a CTS frame with its duration field set accordingly if it is a receiver. Otherwise, it updates its own NAV timer to indicate a busy channel. From this point of view, the RTM frame performs a function similar to that of RTS. It could even share the same format with an RTS frame if the network node degree is no more than 48. As shown in Figure 4.4, a normal RTM frame has the same length as that of an RTS. Their structures are similar except for two different fields: Subtype and RA/RL.
Subtype

The 4 bits of Subtype in the RTS are "1011" while in RTM they are "0001". According to the IEEE 802.11 specification, the bit combination "0001" is not used when the frame control type is "01" (see Table 2.1 on Page 19). By selecting the subtype value of "0001", we achieve compatibility with the existing 802.11 protocol. Besides, from an implementation perspective, it is simple for a mobile node to identify an RTM frame.

Normally, when a mobile node receives a frame, it first checks the receiver address (RA) in the frame. If the RA is not the address of the mobile node or a broadcast address, the node will discard the frame. With the introduction of the RTM frame, mobile nodes are required to check the subtype field of the frame control body. If they have identified an RTM frame which is a signal for a reliable multicast transmission, they will follow the SAM protocol. If the frame is not an RTM frame, the node will check the RA field and decide whether to keep or discard the frame as in the normal process.

RL vs. RA

RTS has a 48-bit RA field containing the receiver's MAC address. In RTM, the corresponding field is called RL (Responding List). The RL field is an important feature of RTM. It allows neighbors of a sender to figure out who the receivers are in a multicast transmission. Furthermore, it defines the order in which receivers should respond with CTSs and ACKs. Specifically, if the $i^{th}$ bit is set to 1, the $i^{th}$ neighbor
of the sender is a receiving node; otherwise, if the $i^{th}$ bit is 0, the $i^{th}$ neighbor is not a receiving node. Among all the receivers, the one with the lowest value $i$ will be the first one to send CTS and ACK. The one with the second lowest value $i$ will be the second one to send, and so on.

In general, the length of the RL field could be longer than 48 bits. It depends on the size of the MANET. A MANET can be viewed as an undirected graph with a maximum node degree of $d$. SAM is able to coordinate up to $w$ receivers where $w$ can be set to the value of $d$, and every node $N_i$ in the MANET has $n_i$ neighbors, where $0 \leq n_i \leq w$. The RL field of such a MANET should contain at least $w$ bits, which allows each node to represent all of its neighbors in its NIT. We define the relationship between RL bits and neighbors as follows:

Consider a MANET that has $w$ bits in its RL field. If a Node $N$ has $n$ neighbors in its NIT, the $m^{th}$ bit of its RL field represents the $m^{th}$ neighbor of $N$, where $0 \leq m < n \leq w$ ($m$ is in fact the neighbor sequence, see Definition 4.1.1). The value of every bit in RL could be set to 0 or 1. With the $m^{th}$ bit set to 0, the $m^{th}$ neighbor is not a receiving node of the multicast and it will not reply with CTS and ACK; instead, it updates its own NAV using the duration value in the RTM (See Figure 4.4). With the $m^{th}$ bit set to 1, the $m^{th}$ neighbor is a receiving node and will reply with CTS and ACK if its NAV indicates an idle channel.

SAM requires every neighbor of the transmitter to check its bit in the RL field upon receiving an RTM frame. From the RL field, each receiving node can obtain its own transmission order for sending CTS and ACK. The transmission order is defined as follows:

**Definition 4.1.3 Transmission Order**

Let the RL bits be $S_0S_1\ldots S_{i}\ldots S_{w-1}$, where $0 \leq i < w$.

Let the number of receivers be $m$, $1 \leq m \leq w$.

---

3A receiving node is one of the receivers in a multicast session.
Let the RL bits of the receiving nodes be \( S_{i_0} S_{i_1} \ldots S_{i_j} \ldots S_{i_{m-1}} \), where \( i_0 < i_1 < \ldots < i_j < \ldots < i_{m-1} \).

The transmission order of receiver \( S_{i_j} \) is \( j \).

As long as the transmitter sends out an RTM frame, the order\(^4\) for its receivers to reply is determined. All receivers should send their CTS/ACK frames according to their order after receiving the RTM/DATA frames so that \( n \) CTSs and \( n \) ACKs arrive at the transmitter sequentially without any collisions. We discuss in detail how to arrange the transmission of nodes in the SAM protocol in the next section.

### 4.1.3 Sequential Transmission

Suppose that Node \( X \) is sending a multicast message to \( n \) receivers, and suppose that they are named as Node \( N_i (0 \leq i \leq n - 1) \). A typical transmission session is depicted in Figure 4.5 (propagation delay is not shown in this figure). After Node \( X \) initiates a multicast transmission by sending out an RTM, it waits for time period \( T_{dn} \) before broadcasting the data. This period allows CTSs from all receivers to be returned. \( T_{dn} \) can be obtained using Equation 4.4. When the RTM reaches a receiver, the receiver waits for a certain amount of time before responding with a CTS. The waiting time \( T_{ci} \) of Node \( N_i \) can be computed by Equation 4.5. After receiving the broadcasted data, the waiting time \( T_{ai} \) for Node \( N_i \) before returning an ACK is given by Equation 4.6. From Equation 4.5 and 4.6, we know that receivers have different time slots to emit CTSs and ACKs, which avoids collisions of these frames at the sender side.

\[
T_{dn} = (n + 1)T_{SIFS} + nT_{CTS} \quad (4.4)
\]

\[
T_{ci} = (i + 1)T_{SIFS} + iT_{CTS} \quad (0 \leq i \leq n - 1) \quad (4.5)
\]

\[
T_{ai} = (i + 1)T_{SIFS} + iT_{ACK} \quad (0 \leq i \leq n - 1) \quad (4.6)
\]

The neighbors of Node \( X \) and all the receivers remain in the NAV period during the multicast session. This is ensured by the duration value in the RTM and CTS frames. The duration \( D_{RTM} \) in an RTM frame should be set to cover at least the

\(^4\)For simplicity, we use the term “order” to refer to “transmission order” throughout this thesis.
CHAPTER 4. SAM PROTOCOL AND APPLICATIONS

Figure 4.5: SAM multicast using virtual carrier sense
last ACK so that neighbors' transmissions do not interfere with the current multicast process. \( D_{RTM} \) can be calculated by the sender using Equation 4.7. Each receiver can adjust the duration value in its CTS frame by subtracting the time elapsed before it sends its CTS. The CTS duration \( D_{CTS_i} \) of Node \( N_i \) can be calculated using Equation 4.8. Similarly, the duration of a data frame and ACK frame can be computed using Equations 4.9 and 4.10, respectively. We should mention here that the ACK duration in an 802.11 unicast is always 0 because a unicast ends with an ACK. In contrast, all ACKs in SAM have a positive value of duration except for the last one. This ensures that the channel reservation continues to be valid during the ACK collecting phase.

\[
D_{RTM} = (2n + 1)T_{SIFS} + T_{DATA} + nT_{CTS} + nT_{ACK} \quad (4.7)
\]
\[
D_{CTS_i} = D_{RTM} - (i + 1)(T_{CTS} + T_{SIFS}) \quad (0 \leq i \leq n - 1) \quad (4.8)
\]
\[
D_{DATA} = n(T_{SIFS} + T_{ACK}) \quad (4.9)
\]
\[
D_{ACK_i} = D_{DATA} - (i + 1)(T_{ACK} + T_{SIFS}) \quad (0 \leq i \leq n - 1) \quad (4.10)
\]

By comparing this process of an IEEE 802.11 unicast (see Figure 2.2), we can see that the unicast is a special case of a SAM multicast with \( n = 1 \) receiver. This characteristic of SAM is a great benefit when implementing a reliable multicast protocol. It greatly reduces the design complexity because both unicast and multicast can be unified in the same transmission framework.

### 4.2 SAM Protocol

In this section, we give a complete description of the SAM protocol. The protocol is described by pseudo code and is split into two parts: multicast sender and multicast receiver.

#### 4.2.1 Multicast Sender

1. A node wishing to multicast must first detect a clear channel.

2. After a clear channel is detected, listen to the channel for a period of DIFS. If the medium remains idle within this DIFS period, go to Step 5.
3. Continue to listen until the medium is idle; then generate a random backoff period $T_{\text{backoff}}$ and start a backoff timer to wait for $T_{\text{backoff}}$.

4. a. If the channel becomes busy before the timer expires, suspend the timer and listen to the channel again; when the channel is detected idle again, resume the backoff timer.

   b. When the backoff timer expires, if the channel is busy, go to Step 3.

5. Initialize RTM counter to 0.

6. Transmit (broadcast) an RTM frame to $n$ intended receivers with corresponding RL bits set to 1.

7. Wait a time period of $T_d$ (see Equation 4.4) for CTS frames to arrive.

8. If no CTS has been successfully received, increase the RTM retry counter by 1:
   a. If the RTM retry counter is less than the maximum allowed value$^5$, go back to Step 1 and begin retransmission.
   b. If the RTM retry counter is equal to the maximum allowed value, the multicast transmission stops.

9. If at least one CTS has been received, send the data frame.

10. Wait for a time of $D_{\text{DATA}}$ (see Equation 4.9) to collect ACK frames.

11. If ACK frames from all intended receivers have been correctly received, the multicast session terminates.

12. If at least one ACK frame has not been received, increase the RTM retry counter by 1:
   a. If the RTM retry counter is less than the maximum allowed value, go back to Step 1 and begin retransmission.

---

$^5$In IEEE 802.11, the recommended value is 7. We will use this value in simulations of SAM and other protocols.
b. If the RTM retry counter is equal to the maximum allowed value, the multicast transmission stops.

4.2.2 Multicast Receiver

1. Listen to the channel.

2. If Node X receives an RTM frame from Node A (sender), it should look in its NIT. If Node A is not in its NIT, then this multicast transmission does not involve Node X. Hence, Node X sets its NAV according to the duration field in the RTM. Go back to Step 1.

3. If Node A is in Node X’s NIT, suppose that A’s neighbor sequence in X is $j$, let $x=\text{Neighbor}(X, j)$, X checks the $x^{th}$ bit in the RL field of the RTM:
   
   a. If the $x^{th}$ RL bit is 0, Node X update its NAV according to the duration field. Go back to Step 1.
   
   b. If the $x^{th}$ RL bit is 1, Node X calculate its transmission order $i$. Node X does not update its NAV on receiving other CTS or ACK frames to the transmitter within the time set in the RTM duration field.
   
   c. Wait for $T_{ci}$ (see Equation 4.5), and if the NAV indicates a busy channel, go back to Step 1.
   
   d. Transmit a CTS frame to Node A with the duration field set to $D_{CTS}$ (see Equation 4.8). Start a Wait-For-Data timer $W$.

4. If Node X receives a data frame before timer $W$ expires, it waits for time period $D_{ACK}$ (Equation 4.10), and then sends an ACK frame back to the sender if the NAV indicates an idle channel.

5. If Node X receive no data frame before timer $W$ expires, it goes back to Step 1.
4.3 Applications of SAM

In a MANET, it is highly desirable that the MAC layer provides reliable multicast service to the upper layers. Like any other network, the operations of each node are regulated by a protocol stack, which defines the functions in different layers and the inter-layer cooperation. Any single layer is not sufficient to deliver all network services without interacting with other layers. Generally, an upper layer calls the underlying services of the layer below it through primitives.

There are two scenarios for the network layer to utilize the reliable multicast service realized by SAM. One is during the discovery and maintenance of transmission routes, including unicast routes and multicast routes; the other is to provide reliable multicast transmissions at the application level.

As mentioned in Chapter 1, most of the unicast and multicast routing protocols depend on reliable broadcast to find optimal routes. By deploying SAM at the MAC layer, all routing protocols can call its reliable service. The overall network performance will be improved because the optimal routes avoid unnecessary transmissions and retransmissions.

Today, many multicast protocols at the network layer have been proposed. All of them, however, have serious problems if the multicast application requires high reliability. To satisfy the application layer requirements, it is necessary for these protocols to have reliable support from the MAC layer. Our SAM protocol is able to cooperate with any one of the network layer multicast protocols. In Section 3.1, we categorize the multicast routing protocols into three strategies:

- Strategy (A): by broadcast;
- Strategy (B): by multicast;
- Strategy (C): by unicast.

A multicast router sends data packets out through all or some selected next-hop nodes in Strategy (A) and (B), while in Strategy (C), only a single node is selected as the next hop. To support the protocols in Strategy (A), SAM coordinates all
neighbors and reliably transmits the data frame to them when the network protocol data unit (PDU) is passed down to the MAC layer at a router. For the protocols in Strategy (B), when the network PDU is passed down to the MAC layer, SAM is able to coordinate the specific next hop neighbors and reliably transmits the data frame to them. SAM only needs to coordinate with the single next hop neighbor to support the Strategy (C) protocols. We note that the standard 802.11 unicast protocol is suitable for the Strategy (C) protocols as well.

In summary, SAM is capable of providing reliable services (including unicast, multicast and broadcast) at the MAC layer. Deploying only SAM in 802.11 based MANETs to provide reliable services may simplify the design logic.
Chapter 5

Simulation Setup and Results

In this chapter we compare the performance of SAM, BMW and BMMM* by simulation. Another reliable protocol, Unicast Realized Multicast (URM), is also included for the purpose of comparison. In URM, a multicast request is converted into $n$ unicast requests and transmitted to each receiver separately. URM can be viewed as a theoretical "upper bound" on the average transmission time of a reliable multicast. It can be easily implemented without changing the current 802.11 specification. In fact, URM is used in the simulator NS-2 [2] implementation of the multicast routing protocol MAODV [32]. As mentioned in Chapter 3, BMMM can guarantee that at most one neighbor successfully receives the data packet. For this reason, we will use the modified version, BMMM*, in our simulations.

5.1 Evaluation Metrics

In a busy network where unicast and multicast traffic contend for the same channel, it is possible that the multicast data packets are not correctly delivered to some destinations. We define a successful multicast as follow:

Definition 5.1.1 Successful Multicast

In a multicast transmission, the sender receives ACKs back from a certain percentage of the intended receivers. If this percentage is greater than a threshold $\theta$, 


this multicast is a successful multicast. The threshold $\theta$ is called the success threshold.

To compare the above mentioned four protocols, we use the Successful Delivery Rate, Average Packet Transmission Time, and Average Time for Last ACK to evaluate different protocols. These metrics are investigated under various conditions, including message generation rate, packet size, number of receivers, success threshold, and node max movement speed. We also study the relationship between the node movement speed and the beacon interval in the SAM protocol.

The definitions of the evaluation metrics are given below:

Definition 5.1.2 Successful Delivery Rate (SDR)

The number of successful multicast transmissions $N_{suc}$ (under a certain success threshold) divided by the total number of transmissions $N_{total}$:

$$SDR = \frac{N_{suc}}{N_{total}}$$

Definition 5.1.3 Average Packet Transmission Time (APTT)

The average time for a transmitter to complete a multicast transmission (either successful or unsuccessful).

Definition 5.1.4 Average Time for Last ACK (ATLA)

The average time from the beginning of a multicast until the last ACK from a receiver is received by the transmitter.

Definition 5.1.5 Message Generation Rate (MGR)

If $N$ messages were sent in a time period $T$, let

$$MGR = \frac{N}{T}$$

A reliable protocol should successfully transmit data packets to receivers. In this sense, SDR is a reliability measurement: the higher the SDR of a protocol, the more reliable it is. APTT is a measurement of the network resource (channel) consumption of a protocol. A protocol with a smaller APTT tends to occupy the network channel.
for less time than a protocol with a larger one. This will improve the overall network throughput. On the other hand, the ATLA metric evaluates the "promptness" of a protocol's response. A small ATLA is needed by a reliable protocol so that the sender is able to retransmit at the earliest possible time. This is critical for routing messages and real time applications. In contrast, a large ATLA value may disable the MAC layer retransmission mechanism because the time-out timers in higher layers, such as the TCP layer and the application layer, will likely expire when ATLA is long. The worst case happens when all nodes except the last one successfully receive the data frame. In SAM, BMMM*, and URM, a multicast transmission ends with the last ACK, so ATLA is always the same as APPT. However, in BMW, ATLA is much longer than APPT. It is because BMW does not have all of the ACKs right after the broadcast. We will discuss this more in Section 5.3.

5.2 Simulation Setup

The experimental tool we choose is the Network Simulator (version 2), NS-2. It is a discrete event-driven simulator designed for networking research. It was developed at UC Berkeley and is written in C++ and OTCL. It comes with some frequently used protocols such as TCP, UDP and FTP. Users can extend NS-2 to study new protocols. We implemented SAM, BMMM*, BMW and URM as modules of NS-2 and integrated them into the simulator. We also added in NS-2 the functionality of exchanging neighborhood information among network nodes.

In our experiments, 100 wireless nodes randomly locate on a 1000m x 1000m square area. All the network nodes operate on a 1Mbps channel. The transmission range of every node is 250m. The Direct Sequence Spread Spectrum (DSSS) technology is chosen for the underlying physical protocol. The parameters we choose are typical values for DSSS [2, 3, 22, 27]. Among the 100 nodes, 4 multicast senders are randomly selected such that no two of them are neighbors. A multicast sender is surrounded by a specific number of multicast receivers. The rest of the nodes are not involved in multicast transmissions. Data traffic, including multicast and unicast transmissions, is issued at the MAC layer. Multi-hop routing is not present in the simulations.
In order to compare the performance of SAM and other multicast protocol, we have conducted two series of experiments under static and dynamic network environments.

In a static network environment, the mobile nodes do not move in the experiments. The network nodes do not communicate neighbourhood information with each other. The multicast neighborhood relationship and multicast groups are fixed throughout the simulation. The performance measured in this setup could be regarded to be a theoretical value without considering network overhead such as management frames.

While all previous research only studied protocol performance in a static network environment, we believe it is necessary to investigate the protocols in a dynamic network environment since mobile ad hoc networks are constructed by roaming stations. For this reason, we conducted another set of simulations under a dynamic network environment. We allow network nodes to randomly move with maximum speed $v_{\text{max}}$ in the simulated dynamic network environment. Every node broadcasts neighborhood information periodically. The multicast receivers at the MAC layer are no longer fixed in this environment; instead, nodes join and leave the neighborhood of the sender depending on whether they are within or outside the sender’s transmission range. By incorporating these characteristics, we have a more realistic MANET environment to study. Since more network traffic occurs in the air, we expect the performance in this environment will be degraded to some degree compared to the “ideal” one in a static network environment. It is interesting to see how protocols behave in a dynamic network environment and what impact the environment will have on the performance of these protocols.

In our simulations, all data packets, including multicast and unicast, are randomly generated at a configurable rate. They are evenly distributed throughout the experiment period. The unicast to multicast message ratio is 0.6 : 0.4. Since we focus our study on multicast messages, unicast messages serve as “noise” or bandwidth contenders. We take the means of 10 simulation runs with different random seeds. Each experiment simulates a network transmission period of 9 minutes (540 seconds).
5.3 Experimental Results

5.3.1 Static Network Environment

In this section, we discuss performance in a static network environment.

Successful Delivery Rate

Figure 5.1 plots the SDR of the four protocols against message generation rate (a), success threshold (b), receiver density (c), and packet size (d). From (a) and (c), we notice that the SDR of all protocols drops when either the message generation rate or the number of receivers (i.e., the size of a multicast group) is increased. The underlying reason is that, under both circumstances, more network traffic is generated in the transmission area around the multicast transmitter and receivers. Intuitively, more traffic forces more receivers into the NAV so that they cannot get involved in the transmissions. These two charts show that: SAM enjoys the best SDR among the four protocols, followed by BMMM*, BMW and URM.

Figure 5.1 (d) shows the SDR comparison when the packet size of a message is increased. Since the length of a message is in a linear relationship to the transmission duration, the longer the message, the more time it occupies the channel. Hence, the possibility of packet collision is increased accordingly. As a result, SDR degrades for all protocols. However, SAM still tops the other protocols.

Figure 5.1 (b) shows that the SDR for all protocols decreases under the circumstances of higher success threshold $\theta$. Obviously, the higher the threshold, the less the multicast transmission is considered to be successful. Once again, we observe that SAM is the best. From the chart, the SDR of SAM is about 0.95 when $\theta = 1.0$. This means that 95% of the multicast transmissions have been successfully delivered to all receivers (100%). In contrast, the runnerup protocol BMMM* only has an SDR of about 0.64.

From Figure 5.1, we know that SAM shows the highest SDR under different conditions, which suggests that SAM is the most reliable protocol in the static network environment. BMMM* is the second best performer in terms of SDR. In all cases,
CHAPTER 5. SIMULATION SETUP AND RESULTS

Figure 5.1: Successful delivery rate in a static network
URM drops more sharply as network traffic becomes intense. It cannot deliver reliable service most of the time. These results imply that URM reaches network saturation much faster than the other protocols. This is due to the fact that URM sends each data frame $n$ times while the others send only once. BMW is more stable than URM, but its performance is far from that of SAM.

**Average Packet Transmission Time**

Figure 5.2 shows the APTT against message generation rate (a), receiver density (b) and packet size (c). From (a), the APTT increases as message generation rate becomes greater in all protocols except for BMW. The underlying reason is that more traffic leads to more collisions and more retransmissions for these protocols. Nevertheless, the APTT of BMW drops gradually with higher message generation rate because BMW takes advantage of round-robin behavior: a cumulative acknowledgement of previous packets can be received when broadcasting a new packet. Thus, when the message generation rate increases, a sender can save substantial acknowledgement polling time. The results in (b) and (c), show that BMW has the lowest APTT while SAM has the second lowest one.

To sum up the observations in Figure 5.2, the BMW protocol produces the shortest APTT while SAM demonstrates the second shortest one. The data of Figure 5.1 and 5.2 were collected in the same experiments, therefore, we should point out that BMW sacrifices the SDR to achieve lower APTT (see Figure 5.1). With a slightly higher APTT, SAM achieves a much better SDR. It seems to be a good tradeoff. From these two sets of charts, we can see that the APTT of URM increases significantly faster than any other protocols. This confirms the finding in the previous subsection that URM is far from optimal for multicasting.

**Average Time for Last ACK**

Even though BMW produces the smallest APTT most of the time, it always suffers from the largest ATLA. This is because the multicast sender cannot tell if a data packet has been correctly received by every receiver until it has polled all the receivers.
Figure 5.2: Average packet transmission time in a static network
CHAPTER 5. SIMULATION SETUP AND RESULTS

This polling process takes up a lot of network transmission time. Consequently, the ATLA of BMW is much higher than the other protocols. Figure 5.3 is a comparison of ALTA for the four protocols. The ATLA of BMW can be as large as one second, which is unacceptable in most real time applications.

BMW is not considered to be a good candidate for reliable multicast for the reason that it suffers a very long ATLA. In contrast, SAM demonstrates a considerably shorter ATLA with a comparable APTT. From the experimental results for SDR, APTT and ATLA, we conclude that SAM is not only the most reliable protocol but also one of the most efficient protocols in the static network environment.

5.3.2 Dynamic Network Environment

In this section, we explore the protocol performance in a dynamic network environment. Not much research has been conducted in this area. Results for some protocols, such as BMW and BMMM, were obtained by simulations based on a static network environment. Since mobility is one of the most crucial features of a MANET, it is necessary to create a more realistic simulation environment to study its dynamic
CHAPTER 5. SIMULATION SETUP AND RESULTS

characteristics.

We introduce a few new parameters for a dynamic network. Firstly, network nodes are moving instead of staying in fixed locations. The maximum speed $v_{\text{max}}$ of nodes is configurable in our experiments. A node moves at a speed $v$ generated randomly within the range of $[0,v_{\text{max}}]$. Every 15 seconds, the moving nodes regenerate their speed $v$ and redetermine their directions of movement.

Secondly, nodes keep exchanging neighbourhood information in the simulation. The purpose of this exchange is two-fold:

1. To obtain neighborhood data so that the sender can determine how many neighbors are in the multicast transmission.

2. In SAM, neighbor sequences are interchanged among nodes. This mechanism forms the basis of the protocol’s transmissions.

Mobile nodes broadcast their neighborhood information cyclically. The interval $T_{\text{int}}$ of such a broadcast should affect the nodes’ NIT and hence the transmission performance. We made it an adjustable parameter in our experiments and studied its impact on the protocols. Other parameters such as $T_{\text{exp}}$ and $T_{\text{lease}}$ had been discussed at the end of Section 4.1.1.

In the dynamic network environment, we redid all the experiments from the static network environment. Besides, we also evaluated the performance of each protocol under various moving speeds. For the SAM protocol, we also investigate the relationship between the speed and different intervals.

Successful Delivery Rate

Figure 5.4 shows the SDR against maximum moving speed $v_{\text{max}}$ (a), message generation rate (b), success threshold (c), receiver density (d) and packet size (e).

In Figure 5.4 (a), SAM retains the highest SDR, followed by BMMM*, URM and BMW. On the other hand, the SDR of all protocols drops with increased node maximum speed $v_{\text{max}}$. We expect this result because a higher speed will cause the following effects in the network:
CHAPTER 5. SIMULATION SETUP AND RESULTS

Figure 5.4: Successful delivery rate in a dynamic network
CHAPTER 5. SIMULATION SETUP AND RESULTS

1. A node has a higher probability of moving out of the transmission range of the sender during a transmission;

2. A node tends to travel across a larger transmission region overlapping with more nodes, which leads to a higher chance of collisions;

3. The NIT of a node becomes obsolete sooner before an update can occur. This results in more unnecessary transmissions to absent nodes.

In Table 5.1, we compare Figure 5.4 (b)-(e) with the corresponding charts in Figure 5.1. The table indicates a similar trend in both the static and dynamic network environments with the changes in the parameters (such as MGR, $\theta$). The explanations of the performance trends in Section 5.3.1 still hold for dynamic network environment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SDR in Static network</th>
<th>SDR in Dynamic network</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGR †</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Success threshold $\theta$ †</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Number of receivers †</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Packet size †</td>
<td>↓</td>
<td>↓</td>
</tr>
</tbody>
</table>

Table 5.1: Trend comparison between static and dynamic network (SDR)

SAM again performs best in terms of SDR under all conditions, and BMMM* shows the second best performance in terms of SDR. The performance of URM becomes more stable than it was in the static network environment.

Another observation is that BMW shows the lowest performance in terms of SDR in a dynamic network environment, while in the static network environment, BMW performs better than URM. This is due to the round-robin mechanism of BMW. If a receiving node moves out of the transmission range of a sender, the sender can no longer query the reception status of previous packets when the time comes to query the receiving node. Hence, the sender considers the node to be missing the unacknowledged packets. This delay of the ACK responses severely damages the performance of BMW in the dynamic network environment. Thus, BMW is not a good choice for a protocol in ad hoc networks.
Average Packet Transmission Time

Figure 5.5 depicts the APTT versus maximum moving speed $v_{\text{max}}$, message generation rate (b), success threshold $\theta$ (c), receiver density (d) and packet size (e).

We notice that in Figure 5.5 (a), the APTT of all protocols increases as nodes move faster. As explained in the preceding subsection, mobile nodes are more likely to fail to return ACKs when they are moving with higher speed. Retransmissions prolong the APTT.

The results in Figure 5.5 (b)-(d) are similar to those in Figure 5.2. BMW has the lowest APPT while SAM is second best, followed by BMMM*. URM still has the worst APPT curve. The underlying reasons that we stated in the static network environment still apply in the dynamic network environment.

As shown in Figures 5.4 and 5.5, SAM demonstrates the second smallest average packet transmission time while achieving the most reliable transmissions. (Note that data are collected in the same experiments for Figures 5.4 and Figure 5.5.)

Average Time for Last ACK

The ATLA comparison in Figure 5.6 confirms that BMW still suffers from the largest ATLA in the dynamic network as it does in the static one. It has an ATLA much higher than the ones of other protocols.

Beacon Interval vs SDR

From the analysis in the preceding sections, we find that SAM achieves the best SDR with a low APTT. In this subsection, we investigate the relationship between the beacon interval $T_{\text{int}}$ and the SDR at different maximum speeds. Figure 5.7 exhibits how the beacon interval affects the SDR in the SAM protocol. Generally, a shorter interval helps to produce a higher SDR with the same $v_{\text{max}}$. This is because the beacon interval determines the frequency for refreshing the mobile nodes’ NITs; and the up-to-date neighborhood data reduces the probability of transmitting data frames to non-existent neighbors. For a MANET with a high $v_{\text{max}}$, we recommend that the
Figure 5.5: Average packet transmission time in a dynamic network
CHAPTER 5. SIMULATION SETUP AND RESULTS

Figure 5.6: Average time for last ACK in a dynamic network

SAM protocol use a low $T_{int}$. In Figure 5.7, when $v_{max} = 15m/sec$, we should set the value of $T_{int}$ to be less than 0.5sec.

Although reducing $T_{int}$ can improve the SDR of SAM, the beacon interval cannot be set to too short. Broadcasting beacon frames occupies the channel, and this adds to the management overhead in ad hoc networks. The beacon messages should not be sent too frequently or they will block the data frames. Although decreasing the beacon interval is helpful, to some degree, to improve the SDR, the gain in SDR may not be substantial. For example, in Figure 5.7, when $v_{max} = 1m/sec$, the change in the SDR is not significant for $T_{int}$ between 0.5s and 3.5s. In this case, it is better to set the interval to 2.5 seconds so that the management overhead is smaller while maintaining relatively good performance.

To further study the relationship among SDR, $T_{int}$, and $v_{max}$, we plot SDR vs $d_{avg}$ in Figure 5.8. The variable $d_{avg}$ denotes the average distance that a node travels within each beacon interval. We computed $d_{avg}$ as following:

$$d_{avg} = \frac{v_{max} \times T_{int}}{2}$$  \hspace{1cm} (5.1)

As the chart indicates, SDR is almost linearly related to $d_{avg}$. This emerges out as an interesting finding and provides a simple relationship between SDR and $d_{avg}$.
Therefore, we have obtained a single criterion $d_{avg}$ to replace the parameters $T_{int}$ and $v_{max}$. For example, if we want an SDR higher than 0.8, $d_{avg}$ should be less than 10m. Therefore, we should choose the values of $T_{int}$ and $v_{max}$ according to Equation 5.1.

5.3.3 Experiment Summary

Using two series of experiments, we have investigated four MANET protocols: SAM, BMMM*, BMW, and URM. We empirically show that SAM generates the best SDR while maintaining a short APTT. SAM outperforms the other protocols in both the static and dynamic network environments. Our results for URM indicate that we cannot accomplish a reliable multicast by using the unicast function in 802.11. In this sense, SAM can serve as a good complement to the current IEEE 802.11 protocol.
Figure 5.8: Average moving distance vs SDR
Chapter 6

Conclusions and Future Work

6.1 Research Review

In this thesis, we have discussed the importance of having a reliable multicast protocol at the MAC layer. Such a protocol can service upper layers for a reliable multicast or broadcast transmission. It allows a sender to efficiently send packets to multiple receivers.

Although the demand for a reliable multicast protocol has greatly increased, researchers are still looking for appropriate protocols to address the need. The existing IEEE 802.11 standard supports reliable unicast transmission. However, it only includes a non-reliable broadcasting scheme. This broadcasting scheme does not properly handle the hidden terminal problem, hence it cannot provide reliable broadcast/multicast service.

Previous research had been conducted in several areas. As discussed in Chapter 3, some protocols have been proposed. BMW is one approach to broadcasting data which uses a round-robin mechanism to achieve the streamlined transmissions. BMMM modifies the current unicast scheme by incorporating n pairs of RTS/CTS. Both of the protocols can improve the performance of 802.11.

To the best knowledge of the author, BMW and BMMM are two of the best methods so far. However, both of them suffer a few problems. BMW usually performs well in a static network where nodes seldom move or move slowly. If the network is rather
dynamic and nodes keep moving around, BMW's performance degrades drastically. It is claimed that BMMM performs better than BMW in a mobile network, but it suffers from implementation issues (see Chapter 3). We have introduced BMMM*, a modified version of BMMM, and shown that it is better than BMW. Still, it does not follow the carrier sensing mechanism in 802.11 standard.

6.2 Contributions

We proposed a reliable multicast protocol SAM in this thesis. We devised a method to assign transmission orders to neighbors by broadcasting neighborhood information. Based on that, receiving nodes communicate with the sender sequentially during a multicast session. SAM not only avoids collisions at the sender side, but also tackles the hidden terminal problem properly. As a result, SAM improves a MANET's performance in terms of SDR and APTT. Our major contributions include:

1. We designed a new protocol SAM to reliably deliver a multicast transmission. Two series of simulations were done in static and dynamic network environments. We empirically showed that, by applying a sequential transmission technique, SAM achieves better performance than existing schemes such as BMW and BMMM. Furthermore, we discovered a simple relationship between SDR and $d_{avg}$, which could help determine the parameters of a MANET. We also studied a unicast-based pseudo-protocol: URM. We came to the conclusion that URM does not satisfy the need of a reliable multicast transmission.

2. We studied the behaviors of different protocols in static and dynamic environments. The dynamic network experiments have never been carried out in previous research. We designed a method to exchange neighborhood information among moving nodes. By this means, we simulated a more realistic dynamic network environment. Our work lays the ground for subsequent research in this domain.

3. A feature of SAM is that it is fully compatible with the widely-used 802.11 protocol. First of all, the RTM frame uses a format similar to that of RTS.
A mobile node can easily identify an RTM from an RTS. Secondly, SAM can operate on a current 802.11 network without changing the unicast transmission scheme. Moreover, the unicast process of 802.11 can be viewed as a special case of SAM's multicast (replacing RTS with RTM). This could be beneficial for the implementation of mobile nodes. We can use the same framework to transmit both unicast and multicast data.

6.3 Future Work

Further study of MAC layer multicast protocols for MANETs may follow several avenues:

1. Since our research focuses on MAC layer protocols, we need to examine the effects when higher layers (such as the network layer) invoke the functionalities of SAM.

2. Our study is currently based on single-hop multicasting. In practice, a multicast transmission may involve a multi-hop path. How to extend the SAM protocol to work in a multi-hop network is an interesting topic.

3. Our work is based on the assumption that network nodes share the same transmission range. Real MANETs, however, should support various types of wireless devices working with different transmission ranges. For example, Node A can hear Node B, but not vice versa. How to modify SAM to work in this kind of network (modeled by a directed graph) is worth pursuing.

4. Our performance study of SAM uses success thresholds. The success threshold can affect the number of retransmissions directly. Generally, the higher the success threshold is, the more retransmissions are required in a specific network environment. Obviously, retransmissions introduce more traffic in MANETs, which will worsen the performance in congested networks. To address these problems, a success threshold adjusted dynamically according to the network situation may help.
5. One of the goals of providing reliability to the MAC layer multicast is to improve the efficiency of unicast and multicast routing protocols. However, it is an open question whether reliable multicasts lead to positive effects on the performance of MANETs. Further study is required to evaluate the improvement in efficiency for routings versus the overhead of providing reliability in multicasts.

We believe that research in the MANET domain will lead to more effective protocols and achieve more exciting results.
Bibliography


