This is to certify that the
dissertation entitled

Interactive Video Multicast in Wireless LANs

presented by

Peng Ge

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science

Major Professor's Signature

12/10/04

Date

Interactive Video Multicast in Wireless LANs

By

Peng Ge

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

2004

Abstract

Interactive Video Multicast in Wireless LANs

By

Peng Ge

A *wireless local area network* (WLAN) is a flexible data communication system implemented as an extension to, or as an alternative for, a wired LAN within a building or campus. By using electromagnetic waves, WLANs transmit and receive data over the air, minimizing the need for wired connections. Thus, WLANs combine data connectivity with user mobility in a convenient and simple way.

With rapid advances in wireless technology, mobile users can now be provided with not only voice and data connections, but also video communication services. Many emerging mobile applications involve the delivery of the video streams to mobile hosts (MHs). Some, such as video-on-demand and video-clip browsing, deliver recorded video to users. Others, such as video conferencing, require real-time delivery of live, interactive video streams.

However, WLANs are less reliable than wired LANs due to the higher error rate in wireless channels. Many error control methods may be used to enhance the reliability of wireless communications, with the penalty of more bandwidth consumption and longer delay in delivery. A buffering mechanism is usually adopted if a large delay

(e.g., in seconds) is acceptable. But for those applications that demand *interactive* video streaming over a WLAN, buffering cannot meet the real-time requirement.

This research investigated how to deliver interactive video streams in 802.11 WLANs. *Interactive* in this study means that each video frame in the stream has a real-time deadline in which to be played back. The main focus of the research was on multicast video streaming, where multiple MHs receive the same video stream. Although unicast can be considered as a special case of multicast with only one receiver, wireless multicast is fundamentally different from unicast. In particular, error control for multicast is more difficult than for unicast in current WLANs, due to the minimal support of multicast in IEEE 802.11 MAC layer protocol.

The contributions of this research can be summarized as follows: First, a packet corruption model was proposed to describe the error behavior in WLANs more accurately in simulations. Secondly, *forward error correction* (FEC) was studied and evaluated for its positive impact on the quality of the video streaming service in WLANs. Thirdly, to further improve the video quality, two new error control methods were proposed: *extra packet request* (EPR) in the application layer and *leader-driven multicast* (LDM) in the MAC layer. Finally, the combinations of multiple error control methods were investigated, each combination's performance was evaluated, and the run-time adaptation of error control strategies was studied in order to match the changing conditions. Both experiments and simulations were conducted. The results indicated the best ways to stream video in WLANs under different situations.

To my parents, who raised me into what I am today.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to professor Philip K. McKinley for his guidance over the past four and a half years, and for his assistance in the preparation of this dissertation. In addition, special thanks are due to my colleague, Chiping Tang, whose contribution to MX simulator was really helpful to my research. I wish to acknowledge my gratitude to S. Masoud Sadjadi, Zhinan Zhou, and other CRG students for their valuable input. I also thank Zhinan Zhou for taking time to proofread this dissertation.

The investigation was supported in part by US office of Naval Research under Grant No. N00014-01-1-0744, and in part by NSF grants CDA-9617310, NCR-9706285, CCR-9912407, EIA-0000433, and EIA-0130724.

Table of Contents

LIST OF FIGURES x

LIST OF TABLES xii

1 Introduction 1
1.1 Motivations . . . 1
1.2 Thesis Statement . . . 3
1.3 Contributions Produced . . . 3
1.3.1 Time-based model to describe WLAN error behavior . . . 3
1.3.2 Experimental evaluation of FEC for wireless video . . . 4
1.3.3 Proxy-based error control . . . 4
1.3.4 MAC layer enhancement . . . 5
1.3.5 Adaptive error control . . . 6
1.4 Outline . . . 6

2 Background Knowledge 8
2.1 IEEE 802.11 WLANs . . . 8
2.1.1 802.11 MAC layer . . . 9
2.1.2 802.11 PHY layer . . . 10
2.1.3 802.11e QoS Support . . . 12
2.2 Unicast vs. Multicast in 802.11 . . . 13
2.3 Video Formats . . . 14
2.3.1 H.261 and H.263 . . . 15
2.3.2 MPEG-1 and MPEG-2 . . . 16
2.3.3 MPEG-4 . . . 17
2.4 Outline of the Remainder . . . 18

3 Test Environment and Tools 21
3.1 Experimental Testbed . . . 21
3.1.1 Hardware configuration . . . 21
3.1.2 Software architecture of network streaming . . . 22
3.2 Interactive Video Streaming Player . . . 24
3.2.1 Microsoft DirectShow . . . 25
3.2.2 DivX codec . . . 25
3.2.3 Implementation of real-time video streaming . . . 26
3.3 MX Simulator . . . 28
3.3.1 Overview of MX . . . 29

4 Modeling Errors in WLANs **31**
4.1 Existing Models for Packet Corruption 32
4.1.1 Packet-based models 32
4.1.2 Bit-based model - CBER 34
4.1.3 Trace-based model 35
4.2 Proposed Time-based Model 35
4.3 Corruption Model Evaluation 40
4.4 Correlation among Multicast Receivers 42
4.5 Summary 45

5 FEC-Based Video Error Control **47**
5.1 FEC Background 47
5.1.1 Approaches to FEC 48
5.1.2 FEC-based reliable multicast 49
5.2 Related Works for Video Streaming 51
5.3 Forward Error Correction for Video 53
5.3.1 Dummy packets 54
5.3.2 QoS-Differentiated Error Control (QDEC) 55
5.3.3 Drawbacks of FEC 55
5.4 Extra Parity Request (EPR) 56
5.5 Experimental Evaluation for MPEG-1 58
5.6 Simulations for MPEG-1 62
5.6.1 FEC+EPR, varying the number of the responders 63
5.6.2 Effect of the packet size 64
5.7 Experimental Evaluation for MPEG-4 65
5.7.1 Single video stream 67
5.7.2 Video stream with interfering traffic 68
5.7.3 Two concurrent video streams 70
5.8 Simulations for MPEG-4 71
5.8.1 Performance of FEC+EPR 72
5.8.2 FEC+EPR, varying the number of the responders 72
5.9 Summary 74

6 Leader-Driven Multicast **75**
6.1 Related Works in MAC Layer Enhancement 76
6.2 LDM Operation and Issues 79
6.2.1 Overview of LDM 79
6.2.2 Advantages of LDM 80
6.2.3 Drawbacks of LDM 81
6.2.4 Effect of correlated packet corruptions 84
6.3 Comparing LDM with Pure Multicast 84
6.3.1 Effect of different corruption models 85
6.3.2 Effect of different packet sizes 87
6.3.3 Effect of correlated packet corruptions. 89
6.3.4 Effect of the leader's relative location 90

6.4 Combining LDM with FEC 92
6.4.1 Effect of different corruption models 92
6.4.2 Effect of different packet sizes 93
6.5 Combining LDM with FEC and EPR 95
6.5.1 Effect of different corruption models 95
6.5.2 Effect of different packet sizes 97
6.5.3 Analysis of the backward traffic 98
6.6 Summary 100

7 Adaptive Strategies **101**
7.1 Methods of Adaptation 101
7.1.1 Adaptive FEC 102
7.1.2 Adaptive FEC + EPR 102
7.1.3 Multiple EPR 103
7.1.4 Dynamic Responder in EPR 103
7.1.5 Dynamic Leader in LDM 104
7.2 Information to Coordinate Adaptations 105
7.2.1 Average signal strength 105
7.2.2 Packet error rate (PER) 106
7.2.3 Frame reception log 107
7.3 Effect of Adaptive FEC+EPR 107
7.3.1 Using single EPR 108
7.3.2 Using multiple EPR 111
7.3.3 Effect of different numbers of responders 112
7.3.4 Effect of dynamic responder 113
7.4 Dynamic Leader in LDM 115
7.4.1 Original leader moving away from AP 116
7.4.2 Original leader moving close to AP 117
7.5 Summary 118

8 Conclusions **120**
8.1 Summary of Dissertation 120
8.1.1 Time-based model in simulation 121
8.1.2 FEC-based video error control 121
8.1.3 Leader-driven multicast 122
8.1.4 Adaptive strategy 124
8.2 Topics in Future Work 125
8.2.1 Refined time-based model 125
8.2.2 Experimental evaluation of LDM 126
8.2.3 Further study of adaptive strategies 126

A MX Description Language **129**
A.1 Comments 129
A.2 For loop 129
A.3 Domain 130

A.4 Host 131
A.5 Noise 132
A.6 Topology of the Graph 133
A.7 Network Interface 133
A.8 Loss Model 134
A.9 Link 135
A.10 Attach 136
A.11 Route 136
A.12 IO Control 136
A.13 Property 137

B Sample MXDL script **138**

BIBLIOGRAPHY **142**

List of Figures

2.1 Traces of packet reception using unicast and multicast in a 2Mbps WaveLAN network. . . . 14
2.2 Proxy configuration for a wireless LAN. . . . 19

3.1 Experimental testbed. . . . 22
3.2 Software architecture of the network streaming. . . . 23
3.3 Sample video processing pipelines in DirectShow. . . . 25
3.4 Interactive video streaming system. . . . 27
3.5 Architecture of the MX kernel and the virtual node. . . . 30

4.1 Topology of the simulated noise sources 37
4.2 Bit error rate vs. signal-noise ratio [1] 38
4.3 (1000-byte) Packet error rate vs. signal-noise ratio 39
4.4 Error pattern of a normal WLAN, 2msec interval 42
4.5 Error pattern of a normal WLAN, 5msec interval 42
4.6 Error pattern of irregular environment, 2msec interval 43
4.7 Error pattern of irregular environment, 5msec interval 43
4.8 Correlated packet losses in a WLAN. . . . 44

5.1 Block erasure code for FEC. . . . 49
5.2 Frame reception rate at three different locations. . . . 61
5.3 Comparison of visual quality for four video frames. . . . 62
5.4 Reception vs. the number of the responders, 5% PER. . . . 63
5.5 Reception vs. the number of the responders, 20% PER. . . . 63
5.6 Reception vs. the number of the responders, 40% PER. . . . 64
5.7 Effect of the packet size to FEC+EPR. . . . 65
5.8 Single video stream, FEC-only, 1448-byte packets. . . . 67
5.9 Single video stream, FEC+EPR, 700-byte packets. . . . 68
5.10 With interfering traffic, FEC alone, 700-byte packets. . . . 69
5.11 With interfering traffic, FEC+EPR, 700-byte packets. . . . 69
5.12 Two concurrent video streams, Stream1, 700-byte packets. . . . 70
5.13 Two concurrent video streams, Stream2, 700-byte packets. . . . 71
5.14 MPEG-4 video streaming to 3 moving MHs, 1000-byte packets. . . . 72
5.15 MPEG-4 video reception vs. the number of the responders. . . . 73

6.1 LDM operation. . . . 80
6.2 LDM combined with FEC+EPR. . . . 82
6.3 LDM vs. PM, Random Model. . . . 86
6.4 LDM vs. PM, Markov Model. . . . 86

6.5 LDM vs. PM, ParEx Model. 87
6.6 LDM vs. PM, effect of the packet size. 88
6.7 LDM vs. PM, effect of correlated packet corruptions. 90
6.8 Topology of the test for the leader's relative location. 91
6.9 Effect of the leader's relative location. 91
6.10 Reception rate of I frames as the effect of LDM+FEC. 93
6.11 LDM+FEC, effect of the packet size. 94
6.12 LDM+FEC+EPR, 5% PER. 96
6.13 LDM+FEC+EPR, 20% PER. 96
6.14 LDM+FEC+EPR, 40% PER. 97
6.15 LDM+FEC+EPR, effect of the packet size. 98
6.16 LDM+FEC+EPR, number of the requests. 99

7.1 Frame reception rate of one moving MH, adaptive FEC+EPR. 109
7.2 FEC rate adaptation of one moving MH, adaptive FEC+EPR. 110
7.3 Multiple EPR vs. single EPR, when combined with adaptive FEC. . . . 111
7.4 Reception vs. the number of the responders. 112
7.5 LDM with 3 MHs. The leader (receiver1) moves away from the AP. . . . 116
7.6 LDM with 3 MHs. The leader (receiver1) moving toward the AP. 118

List of Tables

2.1 Typical frame sizes for MPEG-1 format. 16

5.1 The sample MPEG-1 video clip. 58
5.2 Average frame reception rate at different locations (%) 61
5.3 The sample MPEG-4 video clip. 66

6.1 The PER according to the BER and the packet size. 87

7.1 Bandwidth consumption of using adaptive FEC+EPR. 110
7.2 Bandwidth consumption of using adaptive FEC+multiple EPR. 111
7.3 Effect of the dynamic responder in EPR. 114

Chapter 1

Introduction

1.1 Motivations

Wireless computing is a rapidly emerging technology that provides users with network connectivity without having to remain at fixed locations. Many standards have been established to meet different needs in wireless networks. Examples include the *wireless application protocol* (WAP) [2] to provide mobile data connection for cellular phones, the IEEE 802.11 protocol [3–5] to construct *wireless local area networks* (WLANs) as a replacement for wired Ethernet, and Bluetooth technology [6] to enable wireless connectivity among personal digital devices. These standards have stimulated and accelerated the development of both new wireless technologies and new wireless applications.

With rapid advances in technology, the available bandwidth and the extent of wireless connectivity continue to grow. Today, wireless networks can support many applications that were previously limited by scarce wireless resources. One example

is video streaming. Although many packet video streaming programs have been developed for wired networks, such as the Internet and LANs, these programs may not perform well if directly ported to wireless networks, since a wireless environment is different from its wired counterpart in many ways. Even today, relatively few packet video streaming products are specifically designed for wireless environments. This is partially because until recently, the usable bandwidth provided by many wireless networks was not large enough to accommodate video streaming with acceptable quality. But the main reason, we believe is that more effective methods are needed to accommodate the highly variable channel conditions found in wireless networks.

In this dissertation, the author investigated the issues related to the interactive video streaming service in wireless environments, with a primary emphasis on WLANs. *Interactive* in this study means that each video frame in the stream has a real-time deadline in which to be played back. Video frames that are successfully delivered, but have missed their deadlines, are considered the same as undelivered frames. New mechanisms were proposed and evaluated to improve either the quality or the efficiency of the video streaming. The main focus was on video multicast, which is needed in applications such as video broadcasting and video conferencing. In these applications, video content must reach multiple *mobile hosts* (MHs) at the same time. Unicast is a special case of multicast with only one MH receiver. Many approaches to support mobile users involve the use of *proxies* [7, 8], which represent wireless nodes to the rest of the wired network. This concept was adopted and proxies were used to support video multicast in WLANs.

1.2 Thesis Statement

The problem of multicasting digital video across wireless networks involves several issues: channel conditions, encoding schemes, data characteristics, error recovery methods, and interactions among destinations and sources. The multidimensional nature of the problem implies that no single parameter setting is appropriate to all the situations. Rather, adaptive strategies are required.

This study showed that adaptive approaches can either significantly improve the video quality compared to the non-adaptive approaches, at the cost of the same or more bandwidth consumption, or maintain the same video quality, at reduced cost in terms of bandwidth.

1.3 Contributions Produced

This research produced the following contributions:

1.3.1 Time-based model to describe WLAN error behavior

How to accurately model the error behavior in a wireless channel is yet an unanswered question. In their simulation studies, many research groups have used either packet-based or trace-based models, which assume that corruptions occur on a per-packet basis. Since most packet corruptions in wireless networks are due to noise, however, this study took on the argument that whether or not a packet is corrupted depends primarily on the quality of the channel at a particular time, but not on the fate of preceding or subsequent packets. Moreover, large packets are more likely to be

corrupted than small packets, as they need more time to be transmitted in the wireless channel. Any model not addressing these issues cannot be considered accurate.

To support this argument, a time-based model was proposed to describe the error behavior in WLANs and was incorporated into the simulation tests. After calibration was performed on the results of both simulations and the experiments, it was concluded that the time-based model generates realistic loss conditions and is more accurate than other error models.

1.3.2 Experimental evaluation of FEC for wireless video

Forward error correction (FEC) is commonly used as an error control method in real-time communication. The basic idea of FEC is to introduce redundancy in the data stream, so that receivers may recover some or all of the lost packets without contacting the sender. A block erasure code (one of the FEC techniques that this study used) enables the same set of the parity packets to be used to recover from any combination of the packet losses, as long as the number of the lost packets is less than or equal to the number of the parity packets. The experiments indicated that FEC is very effective in correcting small burst errors in packet streams, but is less effective in dealing with large burst errors, which can occur frequently in WLANs.

1.3.3 Proxy-based error control

To further improve the quality of the video streaming service, a protocol was proposed for receivers to obtain additional error control information from the sender when

FEC alone could not recover from packet losses. Until recently, the use of such negative feedback would introduce unacceptable latency for interactive wireless video streaming (adding at least the round-trip time between the sender and the receiver). However, by assigning the retransmission duties to a proxy near the receivers, and by using a WLAN with a data rate of 11Mbps or higher, the delay could be reduced to an acceptable level (i.e., 100 milliseconds).

In order to maximize the benefit of FEC, the additional packets sent from the proxy should always be the parity packets generated by the FEC encoder, instead of the original data packets. In this way, a single extra parity packet can correct any single-packet loss at any MH. This mechanism was referred to as *extra parity request* (EPR).

1.3.4 MAC layer enhancement

Unicast is more reliable than multicast in IEEE 802.11 WLANs, since unicast can benefit from the RTS/CTS/ACK signaling in the MAC layer. Although a new MAC layer protocol with better multicast-support should be able to improve the reliability of wireless multicast, it was not necessary to design an entirely new protocol from scratch. Instead, a *leader-driven multicast* (LDM) was proposed, which would be an enhancement to the existing IEEE 802.11 protocol. Rather than sending the multicast stream to a group address, the proxy sends a unicast stream to a *leader*, which is a designated MH. Each non-leader MH monitors the unicast stream toward the leader, collects the packets, and reconstructs the data stream.

Simulations have shown that the leader MH's packet reception improves significantly due to the ACKs and the retransmissions in the MAC layer. The non-leader MHs, on the other hand, can benefit from the residual effect of the retransmission traffic from the AP to the leader. As a MAC-layer enhancement, LDM can improve the reliability of not only video multicast, but also any other multicast session in the WLAN. Moreover, it is possible to combine LDM with application layer error control methods to further improve the reliability of wireless multicast.

1.3.5 Adaptive error control

The last part of this study addressed the run-time adaptation. All the available mechanisms for error control were studied individually, as well as in combination. The study was conducted by first using experiments on the testbed, then it was extended to simulations involving more MHs. Dynamic reconfiguration at run-time was adopted to investigate possible adaptive error control strategies. Results showed that adaptive strategies can achieve the same video quality with less bandwidth consumption in some cases, and can achieve better video quality with the same or more bandwidth consumption in other cases.

1.4 Outline

The remainder of this dissertation is organized as follows. Chapter 2 reviews the relevant background knowledge that was used in this study. Chapter 3 introduces the testbed in the laboratory and the tools that were developed for the experiments

and simulations. Chapter 4 describes the modeling of the error behavior in WLANs through simulation. Chapter 5 discusses the two application layer error control methods, FEC and EPR, both of which are based on FEC technology. Chapter 6 describes LDM and presents the results of the simulations. And Chapter 7 investigates how different techniques can be combined to form adaptive strategies. Finally, Chapter 8 presents a summary and then draws conclusions from the research results.

Chapter 2

Background Knowledge

In this chapter, several topics related to our investigation will be reviewed.

2.1 IEEE 802.11 WLANs

A *wireless LAN* (WLAN) is a flexible data communication system implemented as an extension to, or as an alternative for, a wired LAN within a building or campus. By using electromagnetic waves, WLANs transmit and receive data over the air, minimizing the need for wired connections. Thus, WLANs combine data connectivity with user mobility in a convenient and simple way.

An international standard for WLANs, IEEE 802.11 [3–5], was recommended in 1997. This standard includes detailed specifications for both the *media access control* (MAC) layer and the physical (PHY) layer.

2.1.1 802.11 MAC layer

802.11 includes two services in the MAC layer protocol: a basic medium access protocol called *distributed coordination function* (DCF), and an optional protocol called *point coordination function* (PCF). DCF works in a contention mode, in which stations have to contend for use of the channel with each data packet transmission. PCF, on the other hand, works in a polling mode, which provides contention-free frame transfer. In PCF, access to the channel is controlled by a point coordinator (PC), which is always located in an access point (AP). The PC maintains a polling list and regularly polls the registered stations for data. In this manner, the PCF can provide real-time service to all the stations on the polling list.

Although PCF may provide better service for real-time applications, it does not support multicast. Therefore, this study focused only on WLANs working in DCF mode and tried to determine how to stream interactive video in such environments.

The DCF operations are based on *carrier sense multiple access* with *collision avoidance* (CSMA/CA). *Collision detection* (CSMA/CD) is not used in 802.11 because a wireless station cannot listen to the channel for collision while transmitting, due to the difference between transmitted and received power levels. Moreover, *carrier sense* (CS) in 802.11 is performed at both the physical layer, which is referred to as physical carrier sensing, and at the MAC layer, which is also known as virtual carrier sensing.

There are different types of the packets that DCF uses, such as the beacons an AP transmits periodically and the data packets that encapsulate the upper layer protocol

packets. In DCF, data packets can be transmitted using one of two methods:

1. In the basic access method, a positive MAC-layer acknowledgement (ACK) is transmitted by the destination station to confirm that the *data packet* (DATA) has been delivered successfully. If an ACK is not received within a specified period of time, the sender retransmits the DATA, restarts a timer, and waits for the corresponding ACK. This procedure repeats until either the ACK is received or the number of the retransmissions for the same packet reaches an upper limit.

2. The second (optional) access method is a four-way handshaking mechanism, which uses request-to-send/clear-to-send (RTS/CTS) in addition to the DATA/ACK signaling, to reserve the channel before each packet transmission. RTS/CTS hand-shaking can effectively reduce the performance degradation caused by hidden terminals [9].

2.1.2 802.11 PHY layer

The original IEEE 802.11 draft [3] defines three different physical-layer implementations: *frequency hopping spread spectrum* (FHSS), *direct sequence spread spectrum* (DSSS), and infrared (IR). The data rates of both 1Mbps and 2Mbps are supported by all the three implementations.

FHSS utilizes the 2.4GHz *industrial, scientific, and medical* (ISM) band, specifically, 2.4000-2.4835GHz of frequency. In the United States, a maximum of 79 channels are specified in the hopping set and three different hopping sequence sets are estab-

lished with 26 hopping sequences per set. Different hopping sequences enable multiple sessions to coexist in the same physical area.

DSSS also uses the 2.4GHz ISM band. Only 11 channels are available in the US, with 5MHz of frequency spacing between adjacent channels. However, coexisting sessions require their center frequencies to be at least 25MHz apart from each other. This implies that at most three overlapping or adjacent sessions can operate without interfering with each other, if and only if they use channels 1, 6, and 11, respectively.

IR PHY differs from FHSS and DSSS because it uses near-visible light (wavelength 850-950nm). Thus, IR communication relies on line-of-sight or reflected transmissions, and cannot pass through walls, as can FHSS and DSSS radio signals.

In addition to the three PHY layers defined in IEEE 802.11, three more PHY layers were approved as 802.11 extensions to support higher data rates. The first extension, IEEE 802.11a [10], defines *orthogonal frequency division multiplexing* (OFDM) PHY with data rates ranging from 6Mbps to 54Mbps. It operates in the 5.0GHz *unlicensed national information infrastructure* (U-NII) frequency spectrum. The second extension, IEEE 802.11b [1], is also known as high rate DSSS (HR/DSSS) PHY. It shares with DSSS PHY the same frequency band (ISM 2.4GHz) and the same channel allocation (11 channels in the US). The highest data rate supported by 802.11b is 11Mbps. The third one, IEEE 802.11g [11] extends the maximum data rate of 802.11b to 54Mbps by using OFDM modulation. 802.11g products have backward compatibility with the existing 802.11b products, as they both work at the same frequency band with the same channel allocation.

2.1.3 802.11e QoS Support

In July 1999, the 802.11 working group initiated a study group (SG11e) with the charter to enhance the 802.11 MAC layer protocol. The resulting 802.11e standard [12] expands the support of applications with QoS requirements, while maintaining backward compatibility. Thus far, two new modes have been added to the original specifications: *enhanced distributed coordination function* (EDCF), an enhanced version of DCF, and *hybrid coordination function* (HCF), a new form of PCF.

EDCF provides differentiated control of access to the medium. The 802.11e draft (D3.0) defines eight traffic categories for priority-based traffic. A *QoS-capable station* (QSTA) marks each of its packets to indicate the service requirement. QSTAs still contend for the medium, but the channel access parameters differ from one *traffic class* (TC) to another. The *QoS-capable access point* (QAP) can dynamically adjust the parameters for each TC to use and advertise such information in the beacon frame. EDCF provides relative QoS differentiation among TCs, but it does not provide any QoS guarantee. A traffic contract for a connection is only an objective that the wireless network tries to honor as often as possible. Therefore, EDCF is less predictable than a reservation-based mechanism.

HCF is controlled by a *hybrid coordinator* (HC), which is co-located within the QAP. HC's role is to allocate *transmit opportunities* (TxOPs) to QSTAs at any time, either in the *contention period* (CP) or in the *contention free period* (CFP), by using the highest priority to gain access to the medium. HCF needs a centralized scheduling algorithm, which takes into account the traffic contract and QoS requirement of each

active connection. By knowing the amount of pending traffic belonging to different traffic flows, the HC can adjust its scheduling accordingly and provides QoS guarantees. Since an HC is always located within an AP, it should have the best knowledge to make scheduling decisions.

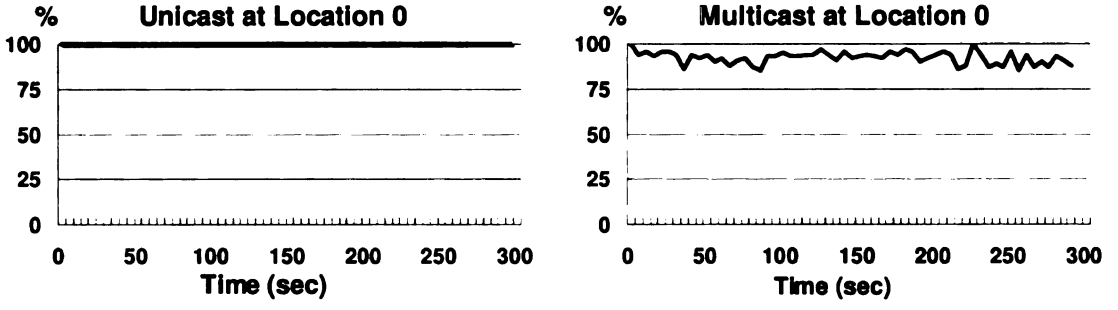
Although 802.11e supports applications with QoS requirements, it is not used in this study because it does not include any special support for multicast.

2.2 Unicast vs. Multicast in 802.11

The IEEE 802.11 MAC layer was designed to improve the reliability of the packet delivery in DCF mode. Collisions can be avoided by using RTS/CTS hand-shaking, and most corrupted packets can be recovered by the DATA retransmissions. However, multicast communications cannot benefit from this frame exchange protocol, since there may be more than one MH involved in a multicast session. Allowing multiple MHs to send CTS'/ACKs in response to single RTS/DATA would most likely produce many collisions.

According to the 802.11 protocol, the sender in a multicast session starts to send DATA whenever the wireless channel is available. No RTS/CTS hand-shaking procedure is invoked. Moreover, the sender does not delay between packets to wait for acknowledgements, since ACKs are not supported. As the result, multicast is less reliable than unicast in WLANs. Multicast traffic is more likely to encounter collisions than unicast traffic, and any corrupted packet in a multicast session has no chance to be recovered in the MAC layer.

In order to illustrate the difference of performance between unicast and multicast in WLANs, the following experiment was conducted in a 2Mbps WaveLAN. A workstation was set up to send a simple packet stream to a *mobile host* (MH) at a fixed location for five minutes. Unicast was used for the first run and multicast was used for the second. When using unicast, the maximum number of retransmissions for each packet was set to four. Figure 2.1 shows the packet reception rate according to time for each run. As shown in Figure 2.1, unicast produced 100% reception rate for the entire test, but multicast experienced up to 15% of corrupted packets. Therefore, error correction for multicast is more challenging than for unicast.



(a) unicast (b) multicast

Figure 2.1: Traces of packet reception using unicast and multicast in a 2Mbps WaveLAN network.

2.3 Video Formats

The bandwidth of network connections is usually considered a critical resource, especially in wireless environments. To determine how to support video streaming in WLANs, the video format is an important factor, since the data-compression technology is the key component of the video formats. Moreover, many video encoding

algorithms also include error-resilience features, such that a partially delivered video stream can still reveal most of the video content.

While a wide variety of video encoding algorithms have been developed over the past decade [13–18], video coding technology is likely to be dominated in the foreseeable future by the standards from the *International Organization for Standardization* (ISO) (MPEG-1, MPEG-2, MPEG-4, etc.) and *International Telecom Union* (ITU) (H.261, H.263, etc.). These standards share a basic algorithm that combines block-based motion compensation, which considers one or more nearby frames, with *discrete cosine transform* (DCT) coding of the motion prediction error. In order to maximize the coding efficiency, variable-length Huffman codes are used to represent both the motion compensation information and the transformed prediction error. We briefly review each of these standards.

2.3.1 H.261 and H.263

H.261 [13] is a video coding standard published by ITU in 1990. It was designed for *integrated services digital network* (ISDN) lines. Hence data rates that are multiples of 64Kbps are supported. It is sometimes called $p \times 64Kbps$ (p is in the range of 1-30). The coding algorithm of H.261 is a hybrid of inter-picture prediction, transform coding, and motion compensation. Motion vectors are used to help the codec compensate for motion.

H.263 [14] was designed in 1995 for low bit-rate communications. Early drafts specified data rates less than 64Kbps. However, this limitation has now been removed.

It was expected that H.263 would replace H.261 in many applications and would be used for a wide range of data rates, in addition to supporting low bit-rate applications. The coding algorithm of H.263 is similar to that used by H.261, with some changes to improve the performance and error recovery. For example, half-pixel precision is used by H.263 for motion compensation, whereas H.261 uses full-pixel precision.

2.3.2 MPEG-1 and MPEG-2

MPEG-1 [15, 16] was approved in 1991 as a standard of compressing video files into a convenient format for downloading—or even streaming—across the Internet. The MPEG-1 standard streams video and audio data at 150 kilobytes per second (or 1.2Mbps), the same data rate as a single-speed CD-ROM drive. The standard defines three types of frames: I(ntra) frames encoded as still images, P(redicted) frames derived from the most recent I or P frame, and B(idirection) frames derived from the closest two surrounding I or P frames. According to the standard, the typical encoding parameters include 320 × 240 pixels in size, at 30 *frames per second* (fps), with typical frame sizes as shown in Table 2.1 [19].

	I frames	P frames	B frames	Average
320×240, 30fps	150kbits	50kbits	20kbits	38kbits

Table 2.1: Typical frame sizes for MPEG-1 format.

Approved in 1994, MPEG-2 [17] was designed to be compatible with MPEG-1, yet remain suitable for digital television and *high definition television* (HDTV), which requires interlaced video scanning not supported in MPEG-1. MPEG-2 is widely used

in *digital video disc* (DVD) technology, which typically encodes video at 640 × 480 pixels in size, with a typical output bandwidth of 6Mbps.

If some video frames are corrupted during transmission, both MPEG-1 and MPEG-2 streams may be interrupted and have no visual content to playback, until the next re-synchronization marker (I frame) arrives. Therefore, it may take up to half of a second to recover from a broken video stream if the typical encoding parameters are applied.

2.3.3 MPEG-4

MPEG-4 [18] was finalized in October 1998, and became an international standard three months later. Several extensions have been added since then, and the work on certain features is still in progress. Analogous to the effects of MPEG-1 and MPEG-2 on the CD-ROM and DVD industries, respectively, MPEG-4 was designed to facilitate inter-operability among video streams delivered over the Internet and other distribution channels, such as wireless channels.

One of the new features introduced in MPEG-4 is fine granular scalability (FGS) [20–22]. FGS supports layered video encoding and is resilient to unrecoverable packet losses, which are very common in the Internet and wireless networks. Schaar and associates [21] showed that FGS provides a clear advantage over non-scalable coding in all the situations. In [22], Schaar and Meeham discussed how FGS can be used in wireless networks.

Another new feature of MPEG-4 for error resilience is the reversible variable-length

code, which can be uniquely decoded even when read backwards. This means that the decoder can make use of all the uncorrupted information from the point of corruption to the next re-synchronization marker (I frame). In MPEG-1 and MPEG-2, such information would be useless.

In addition, MPEG-4 introduces data partitioning, which divides the video bit-stream into finer logic units. This further improves the decoder's ability of localizing errors. Each logic unit contains one type of information (DCT information, shape data, etc.) for all the macroblocks in the entire frame. This approach is in contrast to the method used in MPEG-2, in which each macroblock contains its own header, motion, and texture data. Secondary markers are placed between logic units for the decoder to locate each logic unit. When an error is detected in a logic unit, only the rest of that logic unit is discarded instead of the rest of the packet. Therefore, more data can be salvaged and utilized than in other approaches.

2.4 Outline of the Remainder

This study investigated video multicast streaming in 802.11 WLANs. The investigation focused on how to manage error correction in interactive video streaming to multiple MHs. A major challenge was that interactive video streaming required real-time delivery, which excluded an extensive buffering mechanism as an option. While FEC is commonly used for error correction in real-time communication, it is insufficient to address all the issues in wireless video multicast. Therefore, other error control methods were explored in addition to FEC.

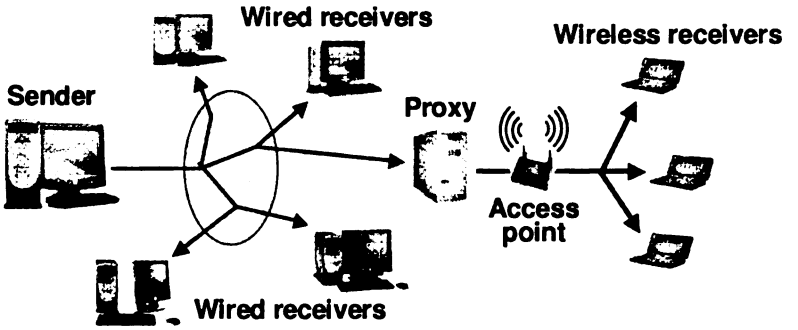


Figure 2.2: Proxy configuration for a wireless LAN.

Figure 2.2 depicts a typical scenario in the investigation. A WLAN is attached to wired networks. A sender, located in a wired network, provides the video streaming service to multiple receivers. For those receivers residing in wired networks, many existing mechanisms can be used, such as IP-multicast [23–26] and layered video multicast [27–30]. As for the wireless receivers in WLANs, a proxy is placed at the edge of the WLAN. It receives the video stream from the sender and relays it to the WLAN. There are two advantages of using a proxy in this context: (1) the proxy may provide extra processing power for error correction; and, (2) the proxy may localize the error control process. The proxy, instead of the sender, can handle an error correction request if any packet is lost in the WLAN.

Both experiments and simulations were conducted. In many cases, simple experiments were first conducted to collect preliminary results and to obtain a basic understanding of the issues. Then the study was extended by simulation to explore settings not supported in experiments, such as those involving many MHs. The testbed in the laboratory, the testing tools, and the simulation program used in the study will be described in Chapter 3.

A key issue in obtaining accurate simulation results is the accuracy of the error

model applied in the simulation. The first contribution of this study was to propose a new model to describe the error behavior in WLANs. Chapter 4 describes the new model in detail after a review of existing models used in simulation. Information on how the configuration of the new error model was calibrated is also included.

Equipped with the tools mentioned earlier, this dissertation can start addressing the issues in video streaming. Chapter 5 discusses two application layer error control methods, FEC and EPR, both of which are based on the technology of FEC. Both experiments and simulations were conducted to evaluate the performance of each individual approach and in combination.

Next, the study was extended by considering enhancements to IEEE 802.11 MAC layer protocol to support multicast. Chapter 6 describes the idea of LDM and presents the simulation results. The experiments are not available for this part since LDM requires modification on wireless network interface card.

Chapter 7 is an investigation of how to combine different mechanisms to form adaptive strategies. The adaptive strategies include dynamically adjusting the redundancy ratio of the forward traffic and dynamic role switching in some error control methods. The simulation results showed that the adaptive strategies are more efficient or more effective than non-adaptive error control methods in many occasions.

Chapter 8 is a summary of the contributions of this study.

Chapter 3

Test Environment and Tools

The investigation of wireless video streaming included both experiments and simulations. For the experiments, a small network testbed was configured to include several wired desktop PCs and wireless laptop computers. A video streaming program was implemented and run on this testbed to test interactive video multicast in a WLAN. For the simulations, a simulation tool, MX, is implemented based on the CSIM simulation package [31].

This chapter describes the experimental testbed, the video streaming program, and the architecture of the MX simulator.

3.1 Experimental Testbed

3.1.1 Hardware configuration

The mobile computing testbed includes both desktop PCs and laptop computers. All the desktop PCs have dual 2.2GHz CPU, 1.5GB memory, and are connected

by 100Mbps Ethernet. All the laptop computers have single 1GHz CPU, 256MB memory, and are connected by an 11Mbps Cisco Aironet 340 WLAN. The Aironet access point (AP) is attached to a 100Mbps Ethernet network.

Figure 3.1 depicts a typical testbed configuration used in the experiments. The sender transmits a video stream to a proxy via the wired network. After necessary processing (transcoding and inserting FEC redundancy), the proxy multicasts the video stream on the WLAN via the AP. Each laptop computer in the WLAN receives the video stream via the wireless connection and plays it back in real-time. The video stream and the statistics of the packet/frame losses are stored at each receiver for analysis at a later time.

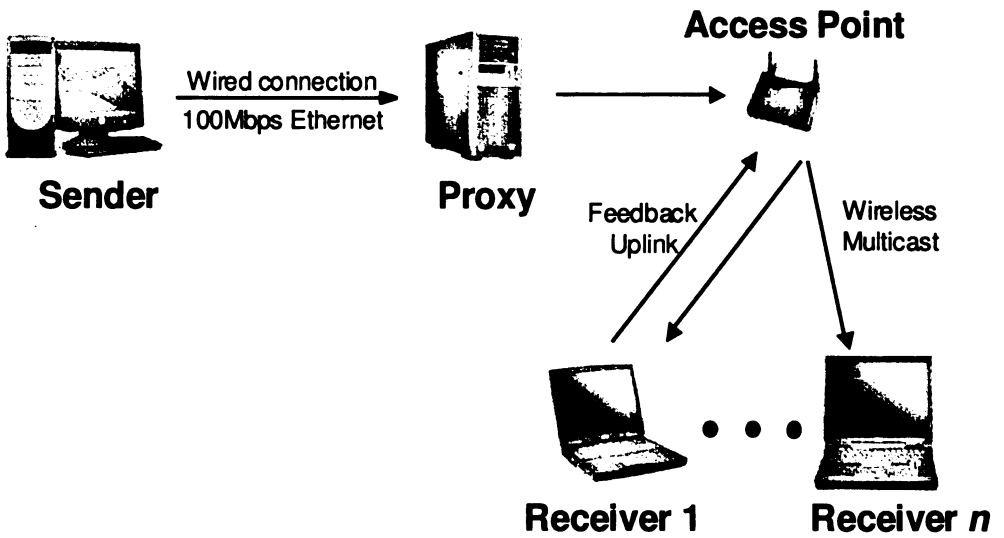


Figure 3.1: Experimental testbed.

3.1.2 Software architecture of network streaming

Figure 3.2 shows the software architecture of the streaming program running on the testbed.

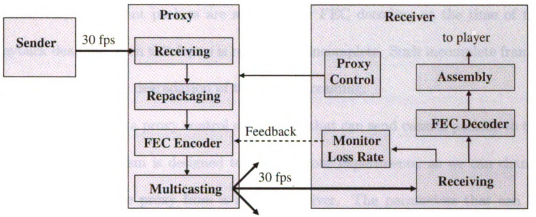


Figure 3.2: Software architecture of the network streaming.

The sender acquires each video frame from the video source and sends it to the proxy. In some experiments, the source is a video camera, although to ensure we can reproduce the video stream characteristics for different tests, we often use a recorded video clip as the video source.

The proxy divides each frame into packets, applies FEC encoding if required, and multicasts all the original data packets and the parity packets on the WLAN. Typically, all the packets from the same frame constitute an FEC group and are buffered at proxy. If more parity packets for this FEC group are requested later by a receiver, the FEC encoder may be re-invoked to generate additional parity packets for the group. An optional feedback channel is available for the proxy to collect information from each receiver. The information to be collected includes the statistics of the recent packet/frame losses, the playback deadline of the video frame, and so on.

Each receiver collects the packets it has received and applies FEC decoding if necessary to reconstruct the video frame, which is then delivered to the video player.

If, however, insufficient packets are available for FEC decoding at the time of the playback deadline, then this frame is reported as incomplete. Such incomplete frames can cause the quality degradation of the video streaming.

Each receiver has a proxy control component that can send control packets to the proxy. This mechanism is designed to facilitate our experiments, as we can change any parameter of the proxy from a remote receiver. The parameters that can be changed include packet size, FEC rate, and streaming delay allowance, among others.

3.2 Interactive Video Streaming Player

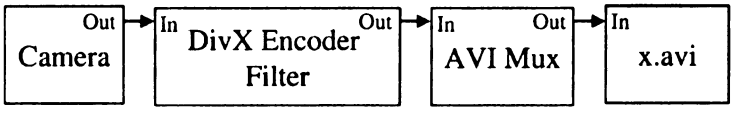
Based on the software architecture described in Subsection 3.1.2, a set of programs was implemented to support real-time video streaming and playback. A video camera was used by the sender to generate a live video feed. The sender then streamed the video to multiple wireless receivers, which played the video in real time. In this way, the video quality could be observed and subjectively evaluated at each receiver. In order to focus on the study of error correction, however, a stored video clip was often used as the video source to generate comparable results. In addition, each receiver could disable the actual video playback to save processing power as well as battery consumption.

The implementation of the video streaming player was based on Microsoft DirectShow [32] and DivX [33, 34], a codec for MEPG-4 video format. Each of these components is described in the remainder of this section.

3.2.1 Microsoft DirectShow

DirectShow is a component of the Microsoft DirectX *application programming interfaces* (APIs) [32]. It encapsulates the functionalities of video capture and video playback, so that the programmers may ignore their implementations and focus on other details. The user of DirectShow typically constructs a *graph* comprising multiple *filters*, connects the filters in proper order, and then starts execution of the graph. Once all the filters are connected correctly, the video capture or playback can proceed autonomously.

Figure 3.3 shows the graph of (a) a video camera connected to the DivX encoder, with encoded data written to an AVI file, and (b) the AVI file being played back.



(a) the graph for video capture

(b) the graph for AVI file playback

Figure 3.3: Sample video processing pipelines in DirectShow.

3.2.2 DivX codec

DivX [33, 34] is a MPEG-4 codec created by DivXNetworks, Inc. The bit-rate of the encoded video stream can be set as a constant value, which may be as low as 300 Kbps while still maintaining acceptable video quality with 30 fps and 320 × 240

resolution. Considering the possible corruptions during the streaming, the maximum spacing between adjacent key-frames (i.e., I frames) was set at 15, that is, at least one I frame in every half of second interval if the video frame rate is 30 fps.

The DivX encoder was configured to generate only I and P frames, but no B frames in the experiments. The reasoning behind this decision was two-fold:

1. The primary reason to introduce B frames into MPEG-1 and MPEG-2 video streams is to reduce bandwidth consumption by recording only the difference between frames. However, DivX can encode video streams of high visual quality at low bit-rates even without using B frames.

2. Using B frames requires the video frames to be delivered out of order (the I or P frame following a given B frame must be delivered before that B frame) to ensure proper decoding. This out-of-order delivery can complicate the video streaming and introduce extra delay.

3.2.3 Implementation of real-time video streaming

Three major components are necessary for an interactive video streaming system to work:

1. a sender that connects with a video source for streaming,

2. a receiver that can play back the streamed video, and

3. a streaming protocol that delivers video data from the sender to the receiver in real-time.

The architecture of the video streaming system is shown in Figure 3.4.

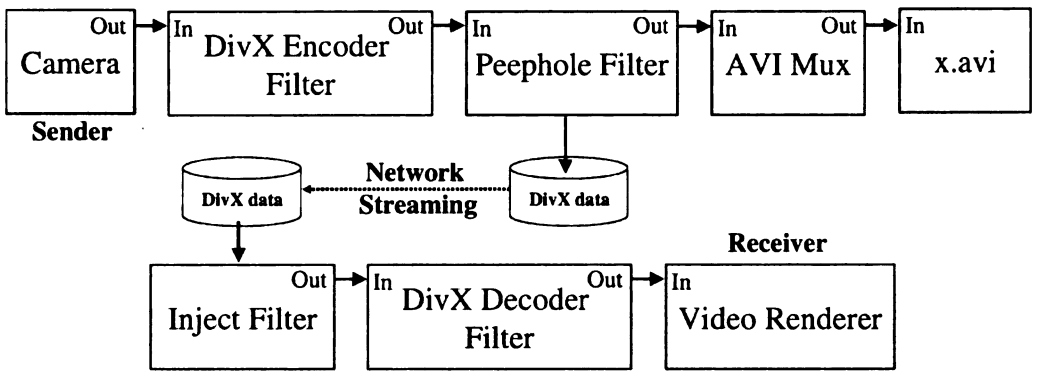


Figure 3.4: Interactive video streaming system.

The graph of the sender is similar to the graph in Figure 3.3 (a), except that it incorporates an extra filter, called a *peephole filter*. The peephole filter takes the data generated by the DivX encoder as input and delivers these to the next filter. Meanwhile, this filter duplicates and redirects the DivX-encoded video data to the network streaming component, which delivers all the data to the receiver using the streaming protocol.

The graph of the receiver is similar to that shown in Figure 3.3 (b), except that the video source file is replaced by an *inject* filter. The main purpose of this filter is to provide the DivX-encoded video data to the DivX decoder, which connects itself with the video renderer. Such video data should be identical to the data collected by the sender's peephole filter, except that some of the data may have been missing due to unrecoverable packet losses during streaming.

In most cases, keeping a one-frame buffer at the receiver helps to smooth the video playback. When the buffer is empty, the video player has to stop and wait for the next frame. If there is more than one video frame in the buffer, the receiver will *fast*

forward the extra frames, that is, displaying these frames for a shorter time than it normally should (e.g., 20 milliseconds instead of 33 milliseconds for a 30 fps video).

Although the delay may vary depending on the computing platform and the type of the underlying network, typical delays introduced by this streaming player are as follows:

- 0-10 milliseconds at the sender (from a frame being captured to the first packet of this frame being sent);
- 0-5 milliseconds for a whole frame streaming across WLAN; and
- 0-40 milliseconds at the receiver (from the time the packet arrives until the entire frame is played back).

In general, the total delay of this streaming player is estimated to be less than 60 milliseconds, which is acceptable for real-time human-to-human interaction.

It is worth pointing out that, although this streaming program was designed with DivX in mind, the same technique can be used with any codec available to the Microsoft Windows operating system. If a better codec for MPEG-4 or even a different video format becomes available, the same streaming program can work properly with the new codec.

3.3 MX Simulator

The simulations in this study were conducted using MX [35], a simulation tool developed to allow unmodified applications to be executed atop a simulated network. Al-

though existing simulators could have been used, like ns-2 [36,37] and GloMoSim [38], neither provided this plug-and-play emulation functionality. Moreover, with MX any property or functionality of any network component can be added or changed as the study demands.

3.3.1 Overview of MX

In MX, a description language (MXDL) is used to define the virtual network topology. All the network components, such as domains, nodes, routers, network interfaces and channels, are also configured using MXDL. A script file written in MXDL is read by MX upon initialization, so that MX understands the configuration of each component as well as the topology of the entire network. Detailed information on MXDL can be found in Appendix A.

Protocol modules in MX are implemented as objects with uniform interfaces, such that multiple modules can be linked together to form a protocol stack. For example, a particular Ethernet connection may be described as using *transmission control protocol* (TCP), over *Internet protocol* (IP), over IEEE 802.3, while a particular WLAN stream may be described as using *user data protocol* (UDP), over IP, over IEEE 802.11. MX currently supports the following protocols: TCP, UDP, IP, CSMA/CD, *multiple access/collision avoidance* (MACA), and IEEE 802.11. Other protocols can be added at a later time.

Figure 3.5 depicts the kernel design of MX and the architecture of the virtual node. Each application program (sender, proxy and receiver) is mapped to a virtual

node, so that the application executes on that node in the simulated environment. MX provides socket-level APIs to link the application code with the core simulator. Therefore, all the networking communications are handled by the simulator.

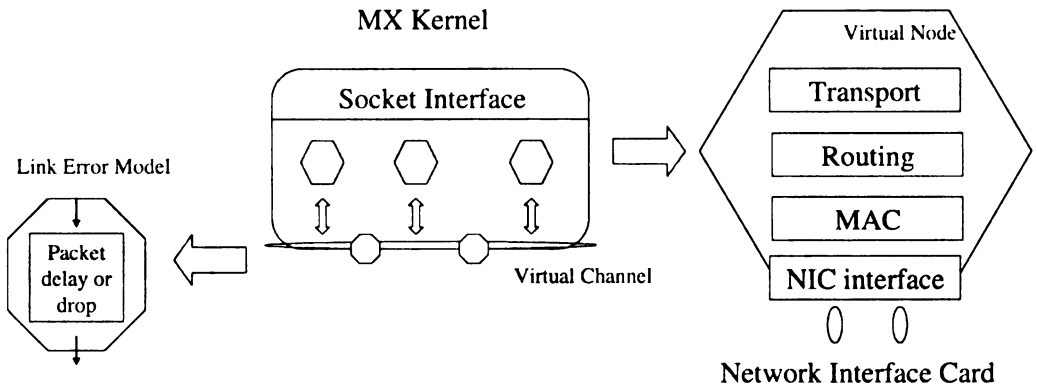


Figure 3.5: Architecture of the MX kernel and the virtual node.

Inserting error models into MX virtual channels enables simulation of lost or corrupted packets. Each link segment in MX can be configured with any of the error models currently supported. Additional details of MX can be found in [35].

Chapter 4

Modeling Errors in WLANs

A key issue in simulating WLANs is the model of error behavior. Some research groups [39, 40] proposed packet-based corruption models based on independent random processes, such that the packet corruptions each receiver experienced over a period of time follow a specific distribution. Others [41, 42] adopted trace-based models, which involved collecting packet reception traces from experiments first, then applying the traces to simulations.

Both packet-based and trace-based models assume that corruptions occur on a per-packet basis, namely, whether or not the next packet will be corrupted is determined before its transmission time, regardless of when the next packet is actually transmitted. In these models, a packet's corruption status will be the same whether it is transmitted at time x or ten minutes later, assuming no other traffic.

This study proposed that whether or not a packet is corrupted depends primarily on the quality of the wireless channel at a particular time, and not on the fate of preceding or subsequent packets. To be more specific, the wireless channel could have

become noisy over a short period of time and become noiseless during the next period of time. If a packet was sent during the noisy period, it was more likely to be corrupted than if it was sent during the noiseless period. Therefore, whether or not a packet would be corrupted due to noise (besides collisions and other protocol/application phenomena) depended not on its relationship with other packets, but on when it was transmitted. Moreover, large packets were more likely to be corrupted than small packets, because they needed more time to be transmitted in the wireless channel. This study stated that any model not addressing these issues could not be considered accurate.

This chapter will initially be a review of existing corruption models, then a new time-based model will be proposed that considers the time factors. To evaluate the accuracy of the model, the generated traces will be compared with actual traces to show that the two are quite close. Finally, there will be a discussion about the correlation of corrupted packets among multiple MHs in the same WLAN.

4.1 Existing Models for Packet Corruption

Several models for packet corruptions were proposed. This section describes those existing models used in this simulation study.

4.1.1 Packet-based models

Three models were adopted to describe the packet-level corruptions. For each test configuration described in Sections 5.6, 6.3, 6.4, and 6.5, the same simulation was

conducted three times, each time with a different packet corruption model applied. The results were analyzed to see how different models can affect the performance. If all the tests with different models exhibit the same pattern in their results, it is likely that such a pattern is model-independent and, therefore, only related to the testing configuration. To make the results comparable in each group of the tests, the parameters of each model were configured such that the overall *packet error rate* (PER) of each model was the same as each other. The three models were:

- *Random model.* Each packet has the same probability, P, of being corrupted.

- *Two-state Markov model [43].* Each packet is related to its preceding packet. Two parameters, $P_{1\rightarrow 0}$ and $P_{0\rightarrow 1}$, define this relationship: $P_{1\rightarrow 0}$ is the probability of the next packet being corrupted, given that the previous packet was successfully delivered; $P_{0\rightarrow 1}$ is the probability of the next packet being delivered, provided that the previous packet was corrupted.

- *ParEx model [44].* In the third model (the most complicated) packet corruptions are represented by the error bursts (i.e., groups of consecutive corrupted packets), and each burst occurs as an independent event. The length of each burst follows the *Pareto* distribution with parameters P_a and P_k, and the starting time of each burst follows the *Exponential* distribution with parameter λ, hence the name ParEx.

4.1.2 Bit-based model – CBER

All the packet-based models determine the corruption status of each packet regardless of its size. For example, a 2346-byte packet (the maximum packet size in the 802.11 MAC layer) would have the same chance of being corrupted as a 20-byte RTS packet. This is an acceptable simplification if there have been only a comparison of the results among different models, but it may be inappropriate for other purposes, such as to determine the optimal packet size for video streaming in WLANs.

A more accurate model may consider the *bit error rate* (BER) of a specific wireless channel to be constant, at least over a short period of time, so that the large packets are more likely to be corrupted than the small packets. To distinguish the results of using different packet sizes, a *constant-BER* (CBER) model was adopted, in which each bit had the same probability of being corrupted. Assuming the BER is P_{ber}, then a k-byte packet is corrupted with the probability of $1 - (1 - P_{ber})^{8 \times k}$. In situations where all the packets have the same size, the CBER model is equivalent to the packet-based random model.

To evaluate the effect of different packet sizes, the same simulation was conducted several times in Subsections 5.6.2, 6.3.2, 6.4.2, and 6.5.2, each time with a different packet size. The CBER model was applied in all the tests. In general, a smaller packet size produces a lower packet corruption rate, but requires more packets for the same data stream. This introduces more processing overhead and more bandwidth consumption. Besides, each of the tests usually favors either larger or smaller packet size, depending on how it operates, which makes it difficult to determine the optimal

packet size for each test configuration. This study expected to provide results that would help in making decisions about what packet size to use in different situations.

4.1.3 Trace-based model

Used by many research groups, a trace-based simulation involves collecting traces from experiments and applying them in simulation to reproduce the same error conditions. This method is straightforward and can generate reasonable results in many cases, especially when an accurate model is unavailable.

A trace-based model was adopted to use in some of the simulations as well. MX can take trace files as input and apply each trace to a specified physical link. Each trace file maintains a record of which packets were lost during a previous session.

4.2 Proposed Time-based Model

A time-based model was proposed and incorporated into MX that described the error behavior in WLANs. The model was based on the following assumptions: (1) any electromagnetic wave propagated in the area using the same frequency as the wireless channel can be considered as either signal or noise; (2) both signal and noise in the air follow the same rule of propagation; and (3) whether or not a packet can be received by an MH without any corruption depends on the *signal-to-noise ratio* (SNR) at the MH's location.

The AP and each MH send out signals that convey data information in the form of packets. When a collision occurs, the corrupted signals are considered the same

as the noises. Other types of noise include background noise, multipath interference, and noises emitted by electronic devices such as microwave ovens, cordless phones, and air conditioning systems.

For the rule of propagation, the formula used by [45] was adopted: assuming the power of the energy source is P_{source}, the power level at an indoor location with distance d from the source should be:

$$P_{destination} = \begin{cases} C_0 \times P_{source} \times d^2 & \text{if } d < 8m \\ C_1 \times P_{source} \times d^{3.3} & \text{if } d \geq 8m \end{cases}$$

where C_0 and C_1 are constant coefficients related to the channel frequency.

Figure 4.1 shows the topology of the noise sources used in one set of simulations, and their relative locations with respect to the AP. There are twelve noise sources (NS1-NS12) present in the WLAN. Each noise source independently follows the ParEx error burst model, which switches between noisy and noiseless periods. To simplify the modeling process, each noise source maintains the same power level for an entire noisy period. The power level can be different in different noise periods, but should fluctuate around the same value. In fact, the power level from a noise source can fluctuate up to 40% above or below the designated value, according to our implementation of the time-based model. Modeling noise sources for a particular WLAN depends on the physical setting. The topology shown in Figure 4.1 was considered to be typical of WLANs.

With the information of the location and the power level for each signal and noise

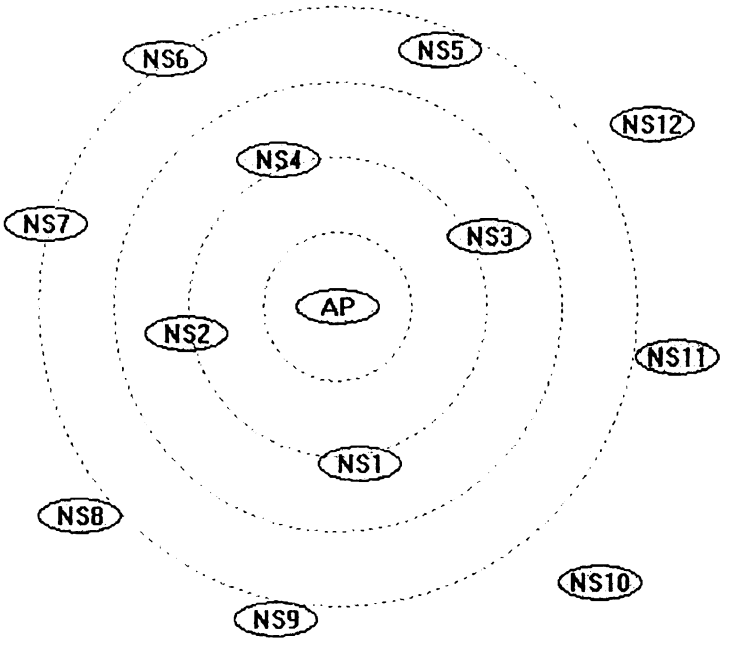


Figure 4.1: Topology of the simulated noise sources

source, the SNR at each MH's location could be calculated at any moment of time. Moreover, if an MH's BER was assumed to be directly related to the SNR and the modulation code it used, the corresponding BER could be calculated for any MH at any moment. Figure 4.2 shows a typical relationship between the BER and the SNR in 802.11 PHY layer, with different modulation codes and different data rates [1]. Figure 4.3 shows the corresponding plot for the PER instead of the BER if all the packets are 1000 bytes in size. Using the data in Figures 4.2 and 4.3 as guidance, it could be calculated and determined that whether or not a packet would be received by an MH without any corruption. The BER needs to be calculated bit by bit, since each packet may experience different SNR values and different BER values during its transmission time.

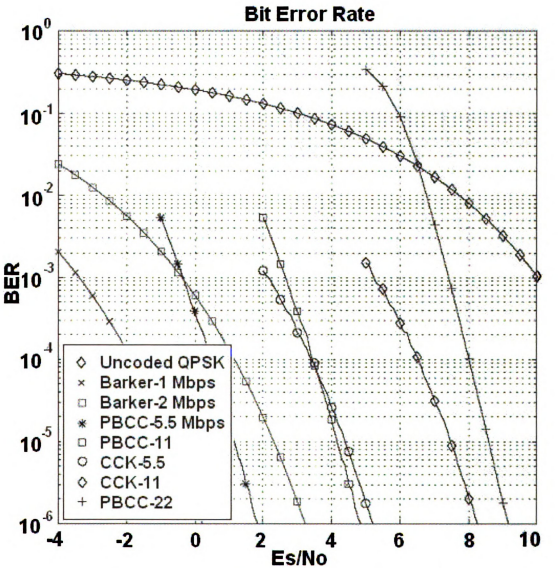


Figure 4.2: Bit error rate vs. signal-noise ratio [1]

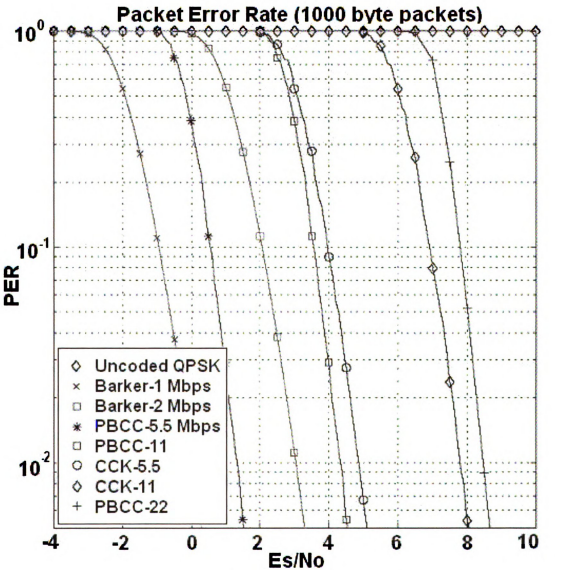


Figure 4.3: (1000-byte) Packet error rate vs. signal-noise ratio

The time-based model was defined by the entire calculation procedure described previously. Although this procedure was complicated and time consuming, a prototype of the time-based model was implemented and incorporated into MX. The performance was acceptable as long as the total number of noise/signal sources was no more than 50.

An advantage of the time-based model is that the occurrence of each corrupted packet is less random and more closely related to the changing environment and the occurrence of other events. For example, when a noise burst from a neighboring noise source interferes with the packet transmission, or another MH happens to transmit a packet at the same time, the MH in question is very likely to receive a corrupted packet. Moreover, if the MH is moving toward the periphery of the wireless cell, the signal strength associated with each incoming packet will decrease, thus making the SNR decrease and BER increase. If a packet-based or bit-based model is applied, on the other hand, the packet corruptions occur based only on the random process, but not on the other events in the WLAN.

4.3 Corruption Model Evaluation

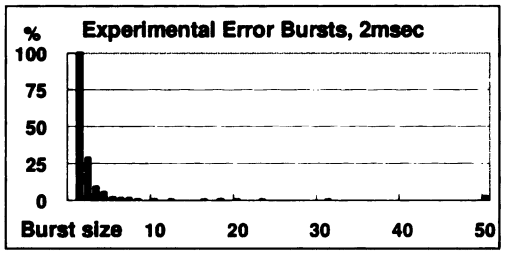
Before applying MX to the simulation study, the following calibration procedure was conducted to make sure that MX, combined with the time-based model, can generate accurate results comparing to the results collected from experiments: A sender program was configured to periodically multicast 1000-byte packets, with a fixed time interval between consecutive packets (e.g., 2 milliseconds). The corresponding receiver

program was run on multiple MHs. Each receiver listened to the packet stream for several minutes. Each receiver program maintained a record of which packets were received and which packets were lost. The same pair of sender/receiver programs could be executed either in the real world testbed or in simulation.

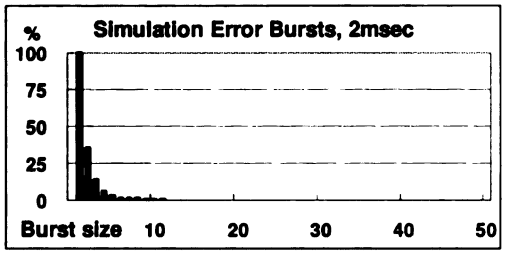
The regular model for WLANs, in which each packet had approximately the same probability of being lost, was evaluated first. Figure 4.4 shows the typical pattern of the packet loss bursts collected from experiments and simulations, with the inter-packet interval set to 2 milliseconds. Figure 4.5 shows the results of the same tests with the inter-packet interval set to 5 milliseconds. Each plot refers to the total number of 1-packet losses as 100%. As shown, most packet losses are one-packet losses. Two-packet losses are less common than one-packet losses. The longer the burst is, the lower the occurrence probability.

Next, a special calibration was conducted by trying to reconstruct an irregular environment. There is a part of the laboratory where abnormal error behavior was observed during the experiments. For every 600 milliseconds, many packet losses were observed over a period of 200 milliseconds in that area. Moreover, during each of the 200-millisecond period with high packet loss rate, the distribution of the lost packets showed some special patterns depending on the inter-packet interval.

Since the testbed was severely interfered with many noise sources, it was believed that there was a periodic noise source in the area that was responsible for the abnormal error behavior. A special noise source was set up in simulation that produced strong interference in a period of 200 milliseconds for every 600 milliseconds. Figure 4.6 shows the typical pattern of packet loss bursts collected from experiments and

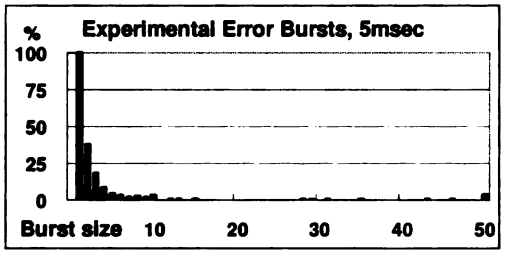


(a) Experiment

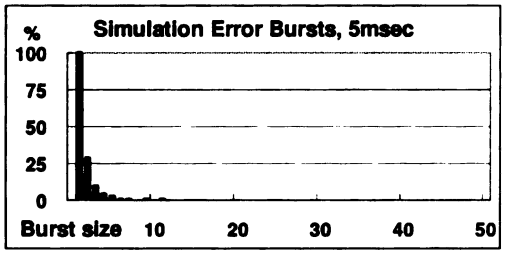


(b) Simulation

Figure 4.4: Error pattern of a normal WLAN, 2msec interval



(a) Experiment



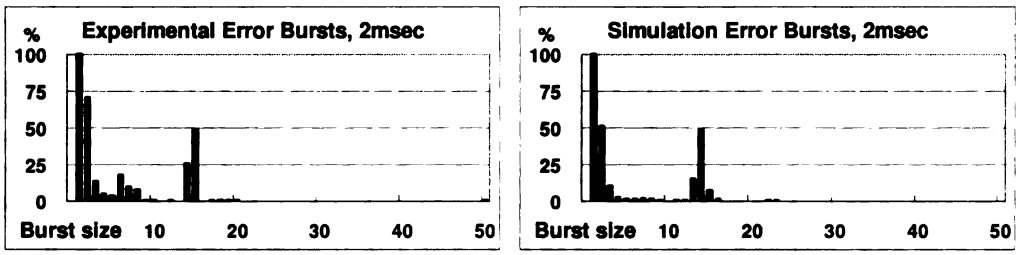
(b) Simulation

Figure 4.5: Error pattern of a normal WLAN, 5msec interval

simulations, with the inter-packet interval of 2 milliseconds, while Figure 4.7 shows the results of the same tests with the inter-packet interval of 5 milliseconds. Tests were also run with inter-packet interval of 10, 15, and 25 milliseconds, and each time the simulation result showed the same pattern of error bursts as the result of the experiment.

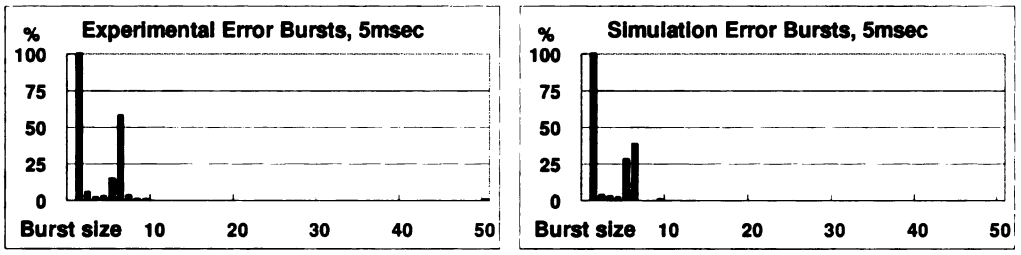
4.4 Correlation among Multicast Receivers

The correlation of packet losses among MHs in WLANs was not thoroughly studied, although some results have been published recently [46]. Most of the earlier studies assumed that the packet losses experienced by multiple MHs in a WLAN were inde-



(a) Experiment (b) Simulation

Figure 4.6: Error pattern of irregular environment, 2msec interval



(a) Experiment (b) Simulation

Figure 4.7: Error pattern of irregular environment, 5msec interval

pendent from each other. However, our experiments showed that the real scenario was more complicated.

There are many factors that may prevent an MH from receiving a packet successfully. Some of the factors, such as signal fading and multi-path interference, can be considered as independent events for each MH. Other factors, however, may affect multiple MHs simultaneously. For example, a packet collision can produce garbled signals, which render the collided packets unavailable to any MH that resides between the two colliding sources. Figure 4.8 illustrates the correlation in packet losses among five MHs in a WLAN. In the experiments, the correlation was observed with the value of as low as 10% and as high as 30%.

One way to simulate correlated packet losses in MX is to use the trace-based

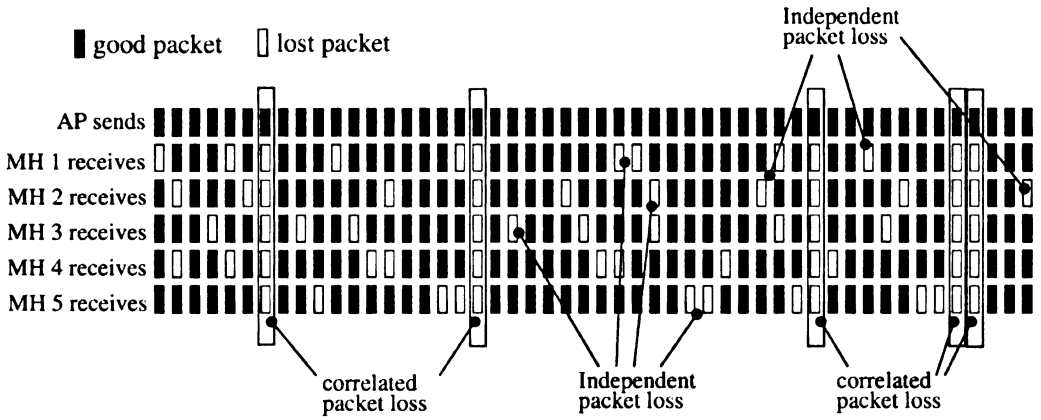


Figure 4.8: Correlated packet losses in a WLAN.

model. Following is the procedure to generate trace files with correlated packet losses using one of the packet-based corruption models (assume the correlation coefficient is $d\%$):

1. Generate a *root* trace according to the specified packet-based corruption model.
2. Take $d\%$ of all the packets from the root trace and copy the corruption status of each packet to each of the non-root traces.
3. Generate all the other packets of each non-root trace independently according to the same packet-based corruption model.

All the traces generated from this procedure should have the same PER with their packet error bursts following the same distribution defined by the packet-based corruption model. Moreover, the packet losses of the root trace are $d\%$ correlated with the packet losses of each non-root trace.

If, however, the time-based model is applied, all the MHs close to each other should have high correlation in their packet losses, since they are likely to be affected

by the same subset of the noise sources. Therefore, different correlations in packet losses among all the MHs are expected in a multicast session, depending on their relative locations.

4.5 Summary

In this chapter, modeling of wireless channel errors was discussed. Obtaining an accurate model was important to the simulation studies of video streaming in WLANs.

Three types of the corruption models were adopted in the simulation study. They are: (1) packet-based models, which include the random model, the two-state Markov model, the ParEx model, and the trace-based model; (2) the bit-based model (CBER), which is useful in distinguishing large packets from short ones; and (3) the time-based model, which is one of the contributions of this dissertation.

In the time-based model, each energy source is considered either a signal source or a noise source. Both signals and noises follow the same rule of propagation and energy loss. Also, it is assumed that the BER of an MH depends on the SNR at the MH's location. Although the time-based model is complicated and time-consuming to calculate the strength of all the signals and noises at each MH location at any moment, it is the most accurate model to author's knowledge to describe the error behavior in WLANs.

A calibration procedure was conducted to verify the accuracy of the MX simulator and the time-based model. The same tests were conducted both in the real world testbed and in simulation. Although the simulation results were not identical to those

of the experiments, the distribution of the packet loss bursts were produced in the simulations that could match those of the experiments, which indicates that both the MX simulator and the time-based model are reasonably accurate in modeling real-world environments.

Finally, the concept of correlated packet losses among MHs in a WLAN was discussed. This concept will be revisited in Chapter 6, when a new MAC layer enhancement to the IEEE 802.11 protocol will be discussed for improving the reliability of the multicast communications.

Chapter 5

FEC-Based Video Error Control

So far, this dissertation has discussed the testbed and the software tools that were used in both the experimental and simulation components of this study. The following chapters will focus on protocols that were used to enhance interactive video streaming to multiple MHs in a WLAN.

This chapter investigates error control methods based on *forward error correction* (FEC). Results from both experiments and simulations are presented. A main contribution of this chapter is to propose and evaluate *extra parity request* (EPR), which is a feedback-based error control method at the application layer.

5.1 FEC Background

A wireless channel is often characterized by clustered, location-dependent errors. Therefore, each MH in a WLAN is likely to lose some packets, and different MHs are likely to lose different packets. A simple *automatic repeat request* (ARQ) approach

to error control is insufficient for multicast communications, since the sender may experience acknowledgement/negative-acknowledgement (ACK/NAK) implosion [47]. Moreover, since the same packet may be requested multiple times by different MHs, the sender may have to retransmit every packet repeatedly in the worst case. Instead of ARQ, this study adopted FEC as the fundamental error control method in the investigation.

5.1.1 Approaches to FEC

FEC [48] is a technique commonly used to handle packet losses in real-time communications. Most FEC protocols make use of the *cyclic redundancy check* (CRC)-based error detection in the link layer. A CRC mechanism usually removes the corrupted packets, thus generates some "erasures" of the data stream. FEC introduces redundancy into the data stream, so that the receiver may correct such erasures without contacting the sender.

One way to implement FEC is to use a *block erasure code* [49]. As shown in Figure 5.1, a (n, k) block erasure code converts k original data packets into a group of n encoded packets, such that any k out of n encoded packets can be used to reconstruct the k data packets. Usually, the first k packets in each group are identical to the k original data packets; the remaining $n - k$ packets are referred to as *parity packets*. The advantage of using a block erasure code for multicast is that the same set of parity packets can be used to correct different packet losses among different receivers.

The number of the parity packets transmitted pro-actively with the original data

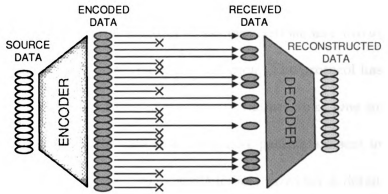


Figure 5.1: Block erasure code for FEC.

packets can be variable [50]. The sender may send fewer than $n - k$ parity packets initially to save the bandwidth. In cases where more parity packets are needed, the sender may transmit additional parity packets either based on prediction or upon request.

Karande and Radha [51] recently proposed a rate-constrained FEC coding scheme to improve the data recovery under severe channel conditions, so that partial data can be recovered even if less than k packets are received for an FEC group. In the same situation, a normal (n, k) block erasure code cannot recover any partial data unless some of the received packets are identical to the original data packets. Integration of this technique with those proposed in this dissertation is a topic for future research.

5.1.2 FEC-based reliable multicast

FEC, and especially the use of block erasure codes, has been applied to the problem of reliable multicasting in wireless networks.

Rizzo and Vicisano [52, 53] proposed a *reliable multicast distribution protocol*

(RMDP) based on Rizzo's work in efficient implementations of block erasure codes [49]. RMDP is a hybrid FEC+ARQ protocol to use over MBone and wireless mobile networks with asymmetric channels for communication. The protocol has several operating parameters to set according to the type of the network. One such parameter, the expansion factor D, is the rate at which parity packets are sent unconditionally with the original data packets. Rizzo and Vicisano provided a detailed analysis of the parameter D, pointing out that the appropriate value depends on the loss rate. Moreover, the value of D between 1.5 and 2.0 makes the probability of NAKs very low. A software implementation of FEC is somewhat expensive, but their results show that the performance is adequate for a wide range of applications, in either wired or wireless networks.

McKinley and Mani [54] extended the work of Rizzo and studied proxy-based adaptive FEC on reliable multicast in WLANs, using (n, k) block erasure codes. Since the quality of the wireless channel may vary dramatically within the wireless cell, using static FEC parameters may be insufficient when the noise level is high and may waste bandwidth when the channel is noiseless. In [54], the number of the parity packets transmitted with k data packets varies, depending on the current packet loss rate. If any MH does not receive sufficient packets for an FEC group to reconstruct the original data packets, the additional parity packets can be requested from the proxy. The goal of this research was to reduce the amount of NAK feedbacks while minimizing the bandwidth consumed by the parity packets. A parameter for the proactive rate, α, was adjusted by taking into account the feedback from multiple MHs.

5.2 Related Works for Video Streaming

Over the past few years, several methods have been proposed to improve the quality of digitized video transmitted over wireless channels.

Liu and El Zarki [55] argued that retransmissions in hybrid ARQ schemes cause delay and that long delays are intolerable for interactive real-time applications. Therefore, they proposed *adaptive source rate control* (ASRC) in addition to hybrid ARQ. The ASRC scheme dynamically sets the target source rate based on the channel condition, transport buffer occupancy, and delay constraints so that the available channel bandwidth is efficiently utilized. The channel condition can be described in terms of the ACKs received by the sender. In this way, the video data encoded at the target source rate can be correctly transmitted within the delay bound imposed by the applications. This study was based on the streaming of low bit-rate video encoded in H.263 (QCIF, 15 fps) over a constant bandwidth wireless channel (32kbps), which is typical for current *personal communications service* (PCS) networks. A similar approach might be applicable to WLANs, although this issue has not yet been studied to the best of the author's knowledge.

Xu et al. [56] proposed *QoS-directed error control* (QDEC) for video multicast in wireless networks. They argued that error control should be performed in a differentiated manner, since most video formats generate encoded video frames with various importance. In the case of MPEG-1 video format, for example, QDEC applies FEC to I and P frames, but introduces no redundancy for B frames. Since I and P frames are more important than B frames in a MPEG-1 video stream, this approach pro-

vides a good tradeoff between *quality-of-service* (QoS) and bandwidth consumption in WLANs. The study presented in [56] evaluated this tradeoff through simulation, and we [57] later confirmed and extended those results through experiments.

Feamster and Balakrishnan [58] discussed an idea similar to QDEC, but in the context of MPEG-4. In addition, their experiments on an emulated network showed that selective retransmission of I-frame data can result in significant performance gains. All the links of their emulated network had an effective bandwidth of 1.5Mbps. A buffering time of at least 200 milliseconds was introduced at the receiver end. Although this study targeted on real-time video streaming over Internet, the same results might be effective in WLANs.

Qiao and Shin [39] discussed a two-step adaptive error recovery scheme for video transmission over wireless networks using H.263 video format. Several FEC codes with different encoding parameters are available to choose from. The best code is determined based on the current channel status, the available time slots before the playback deadline, and the target QoS requirement. The methods investigated in [39], however, were intended for unicast service only.

Ramchandran and associates [59] proposed a hybrid FEC+ARQ method for wireless video streaming service using either unicast or multicast. For each group of k data packets, $n - k$ parity packets are generated using the FEC coding scheme described in Section 5.1. Only the first k data packets are sent initially to an MH or several MHs. The sender will then start to send parity packets one by one until one of the two events occurs: either an ACK (or ACKs in multicast) for this group arrives, or the deadline for sending this group is reached. Each MH will send an ACK to the

sender once it receives k packets of a given group. However, this method does not address the issue of ACK implosion in the case of multicast communications since it requires each MH to send an ACK for every group of data packets.

The study presented in this dissertation complemented the contributions described previously by investigating how different protocols could be used in an adaptive manner to support interactive video streaming in WLANs.

5.3 Forward Error Correction for Video

As mentioned in Section 5.1, FEC introduces redundancy into forward traffic to correct partial packet losses of an FEC group. The special advantage of FEC for multicast is that the same set of parity packets can be used to correct independent packet losses among different MHs.

This study focused on the block erasure code implemented by Rizzo [49]. The FEC encoder and decoder from Rizzo's implementation need to allocate the data structures and the buffering space according to the parameters n and k before any invocation of encoding or decoding. Therefore, the value of n and k cannot be changed at run-time.

As a general mechanism beneficial to many applications, FEC could be applied to our study of wireless video multicast. In the tests, the sender always transmitted the video data in the unit of frames. Each frame was then divided into packets by a proxy according to the designated packet size. As soon as k data packets were available at the proxy, the FEC encoder was invoked to generate $n-k$ parity packets. After that, all of the k data packets and some or all of the $n-k$ parity packets were forwarded

to multiple MHs in the same WLAN by multicast.

5.3.1 Dummy packets

Many video frames were too small to compose k packets, even when the packet size was very small. Allowing an FEC group to span two or more video frames might introduce unnecessary delay in streaming, in decoding, and in playing the first frame. Therefore, it would be better if each FEC group comprised no more than one video frame for interactive video streaming.

Dummy packets [54] were used to address the issue of insufficient packets for some of the FEC groups. When fewer than k (assuming j) packets were to be encoded, the encoder was invoked with the j data packets and $k - j$ packets initialized to all 0's. These "dummy" packets were used only for computing the parity packets, and were not actually sent on the network. The application-level header of each packet contained the number of the dummy packets used in the current FEC group, so that each MH could invoke the FEC decoder correctly after receiving sufficient packets from the same FEC group.

For example, assume a 5-packet frame is to be transmitted using a (12, 8) block erasure code. The encoder is invoked with 5 data packets and 3 dummy packets, producing 4 parity packets, which can be transmitted together with the 5 data packets. As long as an MH receives 5 out of the 9 packets, it can reconstruct the frame by invoking the (12, 8) decoder with 5 received packets (data or parity) and 3 dummy packets. In other words, using dummy packets with a (12,8) block erasure code is

equivalent to using a (9,5) block erasure code alone for all the 5-packet video frames.

5.3.2 QoS-Differentiated Error Control (QDEC)

In addition to dummy packets, QDEC, a technique proposed by Xu and associates [56], was also applied. In the case of MPEG-1 video format, a higher FEC rate was applied for I frames, a lower FEC rate for P frames, and an even lower or zero FEC rate for B frames. A similar arrangement of differentiated FEC rates was applied to the case of MPEG-4 video format.

FEC rate is defined as the rate at which parity packets are sent initially with the original data packets. For example, a 30% rate means that if the number of the original data packets from an FEC group is d (which can be less than or equal to k), then the number of the parity packets sent initially for this FEC group is $\lceil 0.3d \rceil$.

5.3.3 Drawbacks of FEC

There are some drawbacks in using FEC for wireless video multicast.

First, a (n, k) block erasure code requires at least k packets in each FEC group to be successfully delivered. This happens only if (1) the $n : k$ ratio is high, (2) the packet loss rate is low, and (3) the packet losses are uniformly distributed. However, multicasting in a wireless environment usually experiences clustered error bursts with high bit error rate (BER), and thus prerequisites (2) and (3) may not be satisfied. A high $n : k$ ratio implies that bandwidth is wasted on unnecessary redundancy, while a low $n : k$ ratio means that consecutive packet losses due to large error bursts are

unlikely to be recovered.

Secondly, many video frames, especially non-key frames, are only a few hundred bytes in size, and can be easily accommodated into one packet for wireless multicast. For such frames, applying FEC encoding with dummy packets is unnecessary since any parity packet can be replaced by an identical copy of the only data packet. For this reason, FEC is considered beneficial only to large video frames.

5.4 Extra Parity Request (EPR)

In order to overcome the drawbacks of the FEC approach, this study proposed the following modifications: For each FEC group, the proxy computes $n-k$ parity packets. However, the proxy may choose to send only a subset of the parity packets, along with the k data packets, to reduce bandwidth consumption. The remaining parity packets may be sent in response to the requests from the MHs that have received insufficient packets to reconstruct the FEC group. This technique has been used previously in a reliable multicast protocol in WLANs [54]. The contribution of this study is to explore the possible benefits when it is applied to interactive video streaming to multiple MHs.

The number of the parity packets initially sent can be as small as one for each FEC group. In this way, all the 1-packet loss in each FEC group is automatically covered. Two or more lost packets in an FEC group may be recovered by requesting the proxy to send extra parity packets.

In the case of interactive video streaming, an MH should request extra parity

packets from the proxy depending on how likely the extra parity packets can arrive prior to the playback deadline of the video frame in question. In a first-generation WLAN, meeting the real-time deadline is difficult, at least for large video frames. Considering Table 2.1, the average size of I frames in a typical MPEG-1 video stream is 150Kbits. At the data rate of 1.6Mbps, the maximum effective data rate of a 2Mbps WLAN, a typical I frame requires at least 94 milliseconds of transmission time. This implies that the delivery of this frame is delayed by three frames if the frame rate is 30 fps. The additional delay incurred by requesting extra parity packets and waiting for their arrival is unacceptable in this situation.

Consider an 11Mbps WLAN with the maximum effective data rate of 6Mbps: a 150Kbit frame requires only 25 milliseconds to be transmitted. By inserting a relatively small delay (e.g., 100 milliseconds), the overhead of this "backward" error control method can be easily absorbed without noticeable impact on the video quality. This method was referred to as *extra parity request* (EPR), and those MHs that were allowed to send requests were referred to as the *responders*. Non-leader, non-responder MHs were referred to as the *listeners*. Considering the real-time requirement of the interactive video streaming, each responder was originally designed to send at most one request for each video frame. This restriction was removed in a later stage of this study. More discussion of multiple EPR vs. single EPR can be found in Chapter 7.

If EPR is enabled, then the issue of NAK implosion may arise [47]. As the number of the responders increases, the proxy risks being flooded by the requests for extra parity packets. To address this problem, global NAK suppression [60] was used. Instead of sending each request to the proxy by unicast, a responder sends the request

by multicast and staggers the transmission time randomly. Meanwhile, each responder monitors the requests sent by other responders. If any responder overhears another responder's request that subsumes one of its own pending requests, then this pending request will be suppressed. If the proxy receives more than one request for the same FEC group, it will respond only to the request that asks for the most packets.

5.5 Experimental Evaluation for MPEG-1

This section discusses the experiments from a Cisco Aironet (11Mbps) using the MPEG-1 video format. The video clip used as the video source is 1.33MByte in size and plays for a period of 13.7 seconds, which yields an average bandwidth consumption of approximately 800Kbps. Table 5.1 shows the statistical information of the 412 frames in this video clip.

frame type	I	P	B
number of frames	69	69	274
average size(bytes)	6195	5937	2057

Table 5.1: The sample MPEG-1 video clip.

As shown in Figure 3.1, the sender read the video clip and sent out the video frames according to its frame rate, which is 30 fps. The sender was configured to read from the video clip repeatedly since 13.7 seconds is rather short for experiments. The proxy received the video frames, divided each frame into 1440-byte packets, applied FEC encoding, and forwarded them into the WLAN. Each MH in the WLAN collected all the packets from the video streaming and saved all the statistical information to the

hard drive. There was no simple way to play a MPEG-1 video stream while receiving it, so saving the video stream to the hard drive allowed watching and evaluation of visual quality afterwards.

The locations of the MHs were varied with respect to the access point (AP) for different tests. The walls in the building where the experiments were conducted are constructed of concrete blocks, which may have dramatically affected each MH's *packet error rate* (PER). Location 1 was inside the laboratory. Location 2 was outside the laboratory and approximately 30 feet down the hallway. Location 3 was inside an instructional laboratory approximately 60 feet down the same hallway. For each location, each MH continuously received the video streaming for five minutes and recorded the number of the frames that were successfully delivered (a) without using any error control information, (b) by using FEC, and (c) by using the FEC+EPR combination. If a video frame could not recover from packet losses before its play-back deadline expired, it was considered corrupted, even if some extra parity packets arrived later to make the video frame recoverable.

During each test, the proxy applied a fixed FEC rate of 30% to I frames and 20% to both P and B frames. Considering that B frames usually span no more than two packets in this video stream, the 20% FEC rate for B frames implies that for each B frame, one and only one parity packet is sent initially.

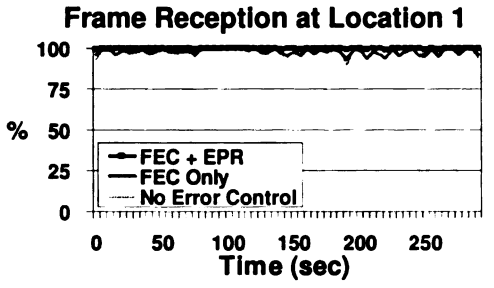
All the MHs in this group of the tests were configured to act as responders, so each MH could request extra parity packets from the proxy based on its experience of the packet losses. When a video frame arrived at each MH, a delay of 200 milliseconds was introduced to absorb the overhead required by EPR. Such a delay was considered

acceptable for interactive video streaming. Moreover, since IEEE 802.11a and 802.11g wireless products with the maximum data rate of 54Mbps are becoming increasingly popular, it is possible to reduce this delay considerably in a 54Mbps WLAN, hence rendering EPR a more suitable error control method than it was in a 11Mbps WLAN.

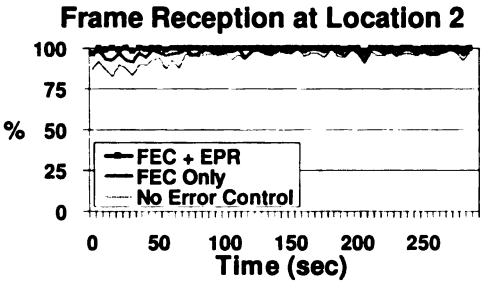
Figure 5.2 (a) shows a trace from one of the MHs at location 1. The most interesting point concerning this trace is the frame reception rate. When the same test was conducted on a WaveLAN network with the maximum data rate of 2Mbps, it was observed that the MHs within the laboratory rarely experienced packet losses. The higher-speed Cisco Aironet was apparently more sensitive to noises than the first-generation WLAN since the multicast traffic experienced up to 10% corrupted frames in the high-speed WLAN. According to Figure 5.2 (a), FEC alone improved the frame reception only slightly, but the FEC+EPR combination produced a frame reception rate close to 100%.

Figures 5.2 (b) and (c) show the traces of a typical MH at location 2 and location 3, respectively. Despite the increased PER at location 2 comparing to location 1, the final frame reception rate is close to 100% when the FEC+EPR combination was used. But at location 3, even the FEC+EPR combination could not overcome the poor channel condition. However, considering that the frame corruption rate was over 50% for several seconds, FEC and EPR improved the frame reception rate considerably. In the worst 5-second period, over 70% of the video frames were corrupted without error control, but about half of those were recovered by the FEC+EPR combination.

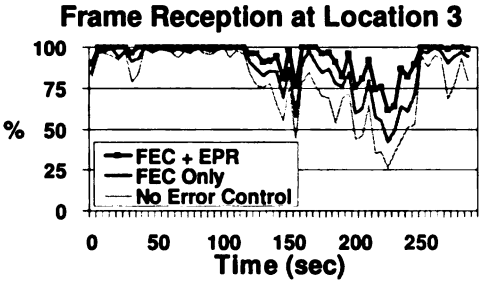
Table 5.2 summarizes the average frame reception rate over multiple traces at each location. The FEC+EPR combination produced the highest frame reception rate at



(a) location 1



(b) location 2



(c) location 3

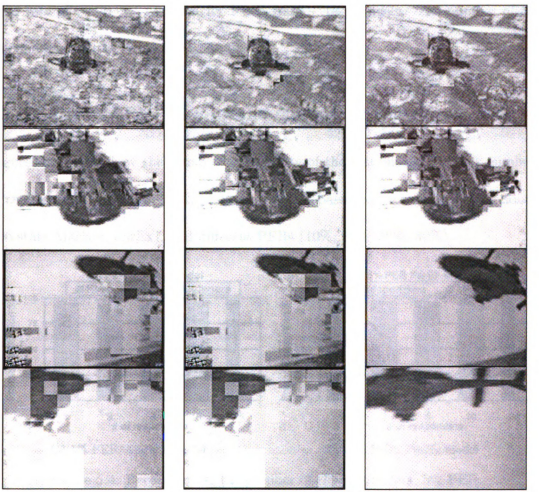
Figure 5.2: Frame reception rate at three different locations.

each of the three locations.

Location	No Control	FEC Only	FEC+EPR
Location 1	97.25	97.77	99.87
Location 2	93.86	97.25	99.73
Location 3	77.99	86.89	94.07

Table 5.2: Average frame reception rate at different locations (%)

Figure 5.3 compares the visual quality of four video frames received by an MH at location 3 when different error control methods were used. For the first two frames shown in Figure 5.3, different mechanisms exhibited relatively small differences in video quality. For the last two frames, the improvement due to EPR was more apparent.



(a) No error control (b) FEC (c) FEC+EPR

Figure 5.3: Comparison of visual quality for four video frames.

5.6 Simulations for MPEG-1

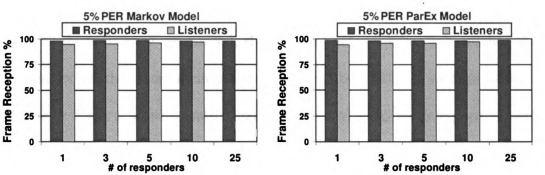
This section extends the study of MPEG-1 video streaming into simulation. The topology of the simulated network is similar to the physical testbed shown in Figure 3.1, except that the simulated network consists of more MHs than the physical testbed. In the following tests, up to 25 MHs were configured as the video receivers.

For the video source of each test, the same MPEG-1 video clip as the one in previous section was used. The packet size is 1100 bytes for most simulations. An

FEC rate of 60-60-0% was used for I-P-B frames, respectively. The number of the responders in EPR varied from 1 to 25.

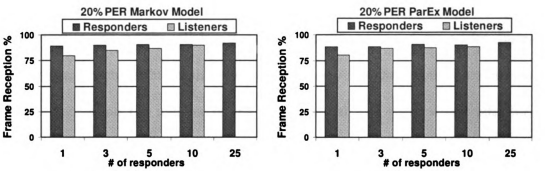
5.6.1 FEC+EPR, varying the number of the responders

Figures 5.4 – 5.6 show the frame reception rate of the tests using different numbers of the responders (1, 3, 5, 10, 25), different packet-based corruption models (random, two-state Markov, ParEx), and different PERs (10%, 20%, 30%, 40%).



(a) 5% PER, Markov model (b) 5% PER, ParEx model

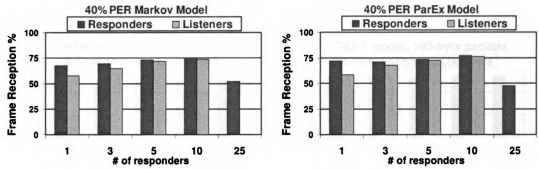
Figure 5.4: Reception vs. the number of the responders, 5% PER.



(a) 20% PER, Markov model (b) 20% PER, ParEx model

Figure 5.5: Reception vs. the number of the responders, 20% PER.

As shown, the frame reception rate of either the listeners or the responders improved steadily when the number of the responders increased from 1 to 10 for all



(a) 40% PER, Markov model

(b) 40% PER, ParEx model

Figure 5.6: Reception vs. the number of the responders, 40% PER.

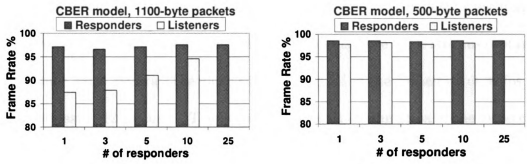
the error models and error conditions. When the number of the responders became 25 (i.e., all the MHs were responders), the general frame reception rate reached the maximum for all the error conditions except when the PER was as high as 40%. The network finally reached its saturation point when as many as 25 responders were present and the mean PER was 40%.

In addition to the extra bandwidth required by the mechanism of EPR, more responders and higher PER consumed even more bandwidth. When the network was approaching its saturation threshold, collisions were more likely to happen. Those responders that sent out the colliding signals were unable to receive any packet during the collisions, which might result in lower reception rates than that of the normal listeners. Increasing the number of the responders at this point would introduce negative impact on the video quality.

5.6.2 Effect of the packet size

In order to evaluate the effect of different packet sizes, the CBER model was applied in the following tests. The BER was set to be 0.00375%. Figure 5.7 shows the I-frame

reception rate of the tests with different packet sizes.



(a) 1100-byte packets (b) 500-byte packets

Figure 5.7: Effect of the packet size to FEC+EPR.

As shown in Figure 5.7, smaller packet size promotes better reception, especially when only a few responders are present. If the number of the responders increases, the reception rate may further improve, until the network reaches its saturation point.

5.7 Experimental Evaluation for MPEG-4

In a later set of experiments, the MPEG-4 video format was used instead of MPEG-1, since MPEG-4 is a newer video coding technology. The interactive video streaming player from Section 3.2 was used in all of the experiments with MPEG-4. As shown in Figure 3.4, a video camera captured a live video stream and relayed it into the WLAN. Each MH received the video stream and played it in real time, which allowed users to evaluate the visual quality during experiments. Meanwhile, each MH stored the video data and the statistical information into its hard drive, making it possible for the quality of the video streaming to be evaluated quantitatively at a later time.

To make the different runs of the tests more comparable with each other, a stored

video clip was used instead of the live video stream captured from a video camera in each of the following tests. Since this video clip was generated by capturing from a live video camera. it should have the same features as the live video streams captured by the same video camera. The video clip was encoded at the frame rate of 30 fps by DivX and lasted 120 seconds. The average bandwidth consumption of the video clip was about 780Kbps. Due to the reason explained in Subsection 3.2.2, this video clip has only I and P frames, but no B frames. Table 5.3 shows the statistical information of the 3602 frames in the video clip.

frame type	I	P
number of frames	241	3361
average size(bytes)	9906	2782

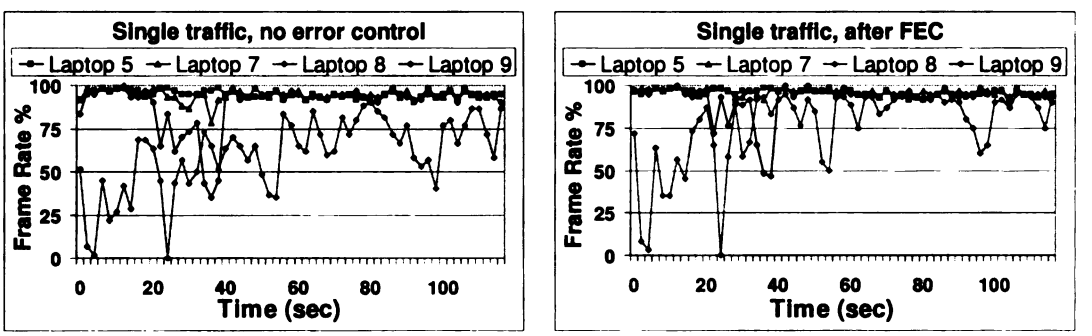
Table 5.3: The sample MPEG-4 video clip.

In all of the following tests, the sender transmitted the video stream to multiple MHs in a WLAN. Each test was repeated several times, each time with a different combination of error control methods. The final reception status of each frame was recorded by each MH. Similar to the MPEG-1 tests, each MH recorded the information of each delivered frame whether it was reconstructed by using (a) original data packets only (no packet loss occurred), (b) original data and FEC parity packets initially sent by the proxy, or (3) all the available packets (including the extra parity packets used in EPR). In this way, each test generated up to three different plots, yielding the frame reception rate of each MH for 1) no error control, (2) FEC only, and (3) the FEC+EPR combination.

5.7.1 Single video stream

In the first set of the experiments, the video stream was the only traffic in the WLAN, therefore, the interference from other traffic was minimal.

Four MHs participated in the following tests. All the MHs used Cisco 350 Aironet *network interface cards* (NICs) except *Laptop 8*, which used a Cisco 340 Aironet NIC card. Laptop 8 always has lower signal strength than the other MHs, probably because the 350 card is an enhanced version of the 340 card. Figure 5.8 shows the frame reception rate of each MH when FEC was enabled with the FEC rate of 50-30% for I-P frames, respectively. The EPR was turned off. The packet size was 1448 bytes for this test.



(a) No error control.

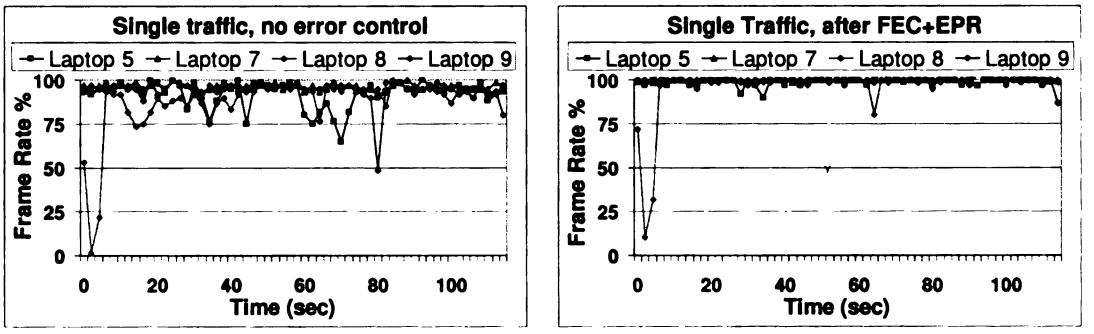
(b) After FEC.

Figure 5.8: Single video stream, FEC-only, 1448-byte packets.

As shown in Figure 5.8, FEC improved the reception of Laptop 8 considerably, but was of marginal benefit to the other MHs. No MH experienced perfect delivery by using FEC.

Figure 5.9 shows the frame reception rate at each MH when EPR was activated in addition to FEC. All the MHs were configured to be responders, and the packet size was 700 bytes. The video playback was delayed by 80 milliseconds to absorb

the overhead of EPR. As shown in Figure 5.9 (b), the reception was significantly improved at all MHs by using the FEC+EPR combination.



(a) No error control (b) After FEC+EPR

Figure 5.9: Single video stream, FEC+EPR, 700-byte packets.

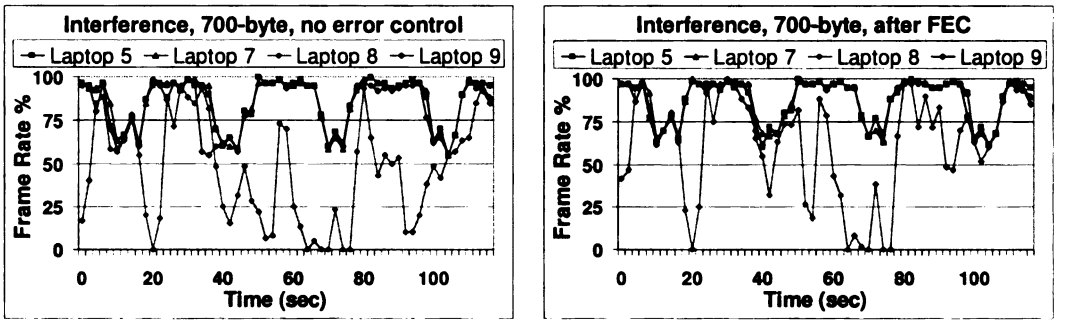
5.7.2 Video stream with interfering traffic

Since DivX-encoded video streams can maintain high visual quality while consuming relatively low bandwidth, a single video stream can be easily accommodated in an 802.11b 11Mbps WLAN, especially when no other traffic exists in the WLAN at the same time. The following group of the tests focused on situations where the interference from other wireless traffic was present in the WLAN.

To establish the testing scenario, one of the MHs was configured to produce the *interfering traffic*. The interfering MH kept sending two 1000-byte packets by multicast in every 33 milliseconds for 10 seconds, then stayed quiet during the next 20 seconds. This pattern of behavior was repeated every 30 seconds. During each 10-second period when the interfering traffic was on, each packet was sent from the MH to the AP by unicast, then the AP sent the same packets to all the MHs by multicast. The bandwidth consumption for each direction (either MH→AP or AP→MH)

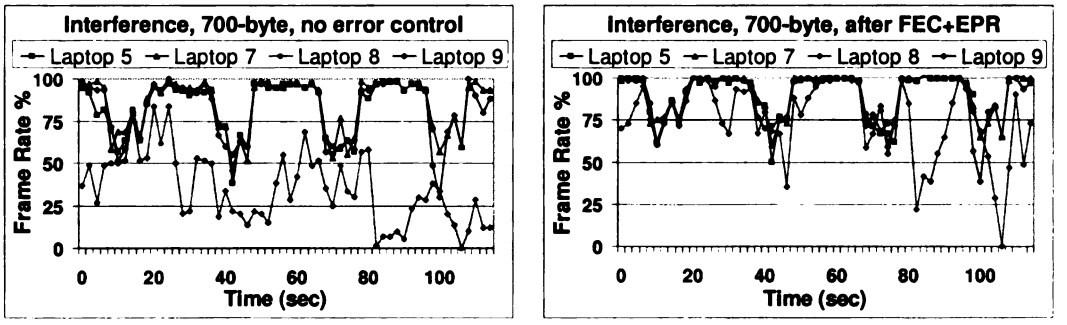
was 484Kbps, plus the RTS/CTS/ACK signaling and DATA retransmissions for the unicast traffic from the MH to the AP.

Figures 5.10 and 5.11 show the frame reception rate of each MH when the proxy used either FEC alone or the FEC+EPR combination. All the MHs were configured to be responders when EPR was activated, and the packet size was 700 bytes for this test.



(a) No error control. (b) After FEC

Figure 5.10: With interfering traffic, FEC alone, 700-byte packets.



(a) No error control. (b) After FEC+EPR

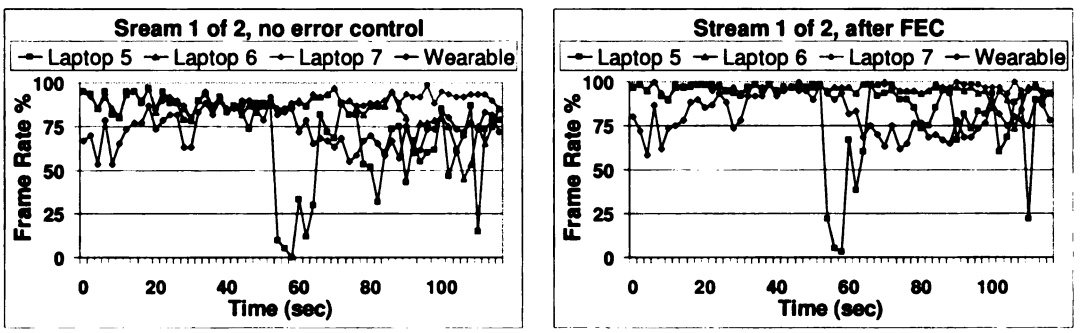
Figure 5.11: With interfering traffic, FEC+EPR, 700-byte packets.

As shown in Figure 5.10 and 5.11, the interfering traffic severely affected the quality of the video streaming, such that a tooth-shaped pattern appeared in all of the plots. FEC could improve the frame reception to a limited degree, but the FEC+EPR

combination achieved the best video quality whether or not the interfering traffic was present.

5.7.3 Two concurrent video streams

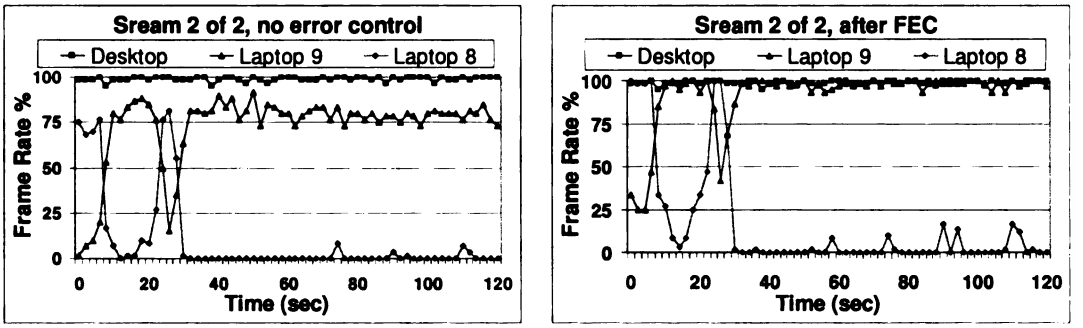
In this group of the tests, two concurrent multicast sessions of the video streaming service were used to see if they could co-exist in a WLAN. The first stream was sent from a desktop PC to four MHs. The second stream was sent from another MH to two other MHs and the desktop PC that was sending the first stream. Each video stream consumed about 780kbps of bandwidth and the packet size for both streaming was 700 bytes.



(a) Stream1, no error control (b) Stream1 after FEC

Figure 5.12: Two concurrent video streams, Stream1, 700-byte packets.

Figure 5.12– 5.13 show the frame reception rate of the two video streams, without any error control or when FEC was used. No usable results for this test could be collected if EPR was activated, since both video streams would suffer from severe quality degradation. It was probably due to the massive packet collisions caused by the two concurrent sessions of the video streaming service and the requests for extra parity packets when EPR was activated for both sessions.



(a) Stream2, no error control

(b) Stream2 after FEC

Figure 5.13: Two concurrent video streams, Stream2, 700-byte packets.

As shown in Figure 5.12 and 5.13, all the MHs had better reception when FEC was used, except Laptop 8, which had zero reception rate most of the time. This might be caused by the weak signal strength resulting from its configuration with a Cisco 340 wireless NIC instead of a 350 NIC.

5.8 Simulations for MPEG-4

This section extends the study of MPEG-4 video streaming to simulation. The topology of the simulated network is similar to the physical testbed shown in Figure 3.1, except that the simulated network consisted of more MHs than the physical testbed. In the following tests, up to 20 MHs were configured as the video receivers.

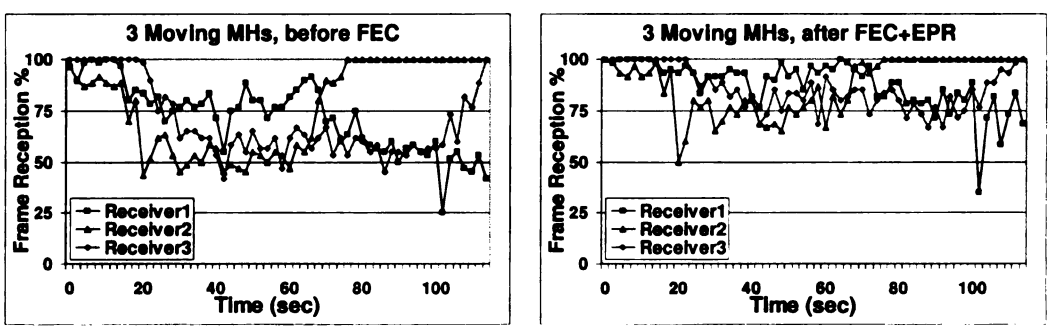
All of the following simulations used the same MPEG-4 video clip previously used in the experiments (120 seconds, 30 fps, DivX-encoded). The time-based corruption model was applied in all of the simulations. Different error conditions were configured to see how the error condition affected the performance. Since no simple parameter could describe the time-based model's error condition, the mean PER of a typical MH was used as a reference when a plain video stream (with no error control) was

sent to this MH by using multicast. The mean PER could be either 15% or 50%.

5.8.1 Performance of FEC+EPR

In the first test, the FEC+EPR combination was used to stream the MPEG-4 video clip to three MHs. Each MH was configured to move randomly in the area. The moving style of each MH was as following: each node moved in one direction until it arrived at the destination and stayed there until the next time it moved again.

Figure 5.14 shows the frame reception rate of each MH according to time. All the MHs were configured to be responders, and the packet size was 1000 bytes. As shown, the FEC+EPR combination improved the frame reception rate significantly at all MHs.



(a) No error control (b) After FEC+EPR

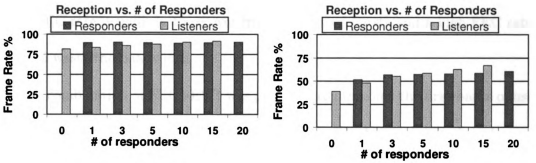
Figure 5.14: MPEG-4 video streaming to 3 moving MHs, 1000-byte packets.

5.8.2 FEC+EPR, varying the number of the responders

In the next group of the tests, the video clip was streamed to 20 MHs in the WLAN, using FEC+EPR with different numbers of the responders each time. To avoid drawing a conclusion from only one trial run that was atypical, five runs were conducted for

each testing configuration, each run with different seed value to initialize the random number generator.

Figure 5.15 shows the frame reception rate of the responders and the listeners with the number of the responders varying from 0 (no responder) to 20 (all the MHs are responders).



(a) Mean PER 15%

(b) Mean PER 50%

Figure 5.15: MPEG-4 video reception vs. the number of the responders.

As shown in Figure 5.15, the listeners always had better reception when the number of the responders increased. As for the reception of the responders, although it stayed at the same level or became better when more MHs acted as responders, it was not as good as the reception of the listeners when the number of the responders was greater than or equal to 10. This suggests that only a small subset of the MHs should be selected to act as responders if the FEC+EPR combination is used, since having more responders does not necessarily improve the reception of the responders due to the collisions between the forward traffic and the backward requests. The listeners, on the other hand, always benefited from the extra parity packets requested by the responders.

5.9 Summary

As a commonly used error control method in the application layer, FEC can improve the quality of the video streaming service in general. However, the effectiveness of FEC in a WLAN may be limited due to the nature of the error behavior in a wireless environment. Since wireless channels are often characterized by clustered error bursts, a low FEC rate is unlikely to recover from large error bursts and a high FEC rate is likely to waste the bandwidth.

The combination of FEC+EPR was proposed to overcome the drawbacks of using FEC alone. EPR allows designated MHs to request extra parity packets from the proxy, if the parity packets sent initially were insufficient to reconstruct the FEC group, due to packet losses. Experiments and simulations showed that the FEC+EPR combination could improve the reception of multicast video streaming in WLANs considerably, with a final frame reception rate close to 100% in some cases. Even when severe interference was present in the WLAN, the FEC+EPR combination could still provide noticeable improvement.

Chapter 6

Leader-Driven Multicast

In Chapter 5, the application layer error control methods for the video streaming in WLANs was discussed. The focus of this chapter, however, will be on the solutions in the MAC layer. The combination of MAC-layer solutions and application layer error control methods will also be discussed.

It was established in Section 2.2 that unicast was more reliable than the regular *pure multicast* (PM) in 802.11 WLANs, due to the *request-to-send/clear-to-send/ acknowledgement* (RTS/CTS/ACK) signaling in the MAC layer. Also due to all the wireless signals that were transmitted in the air, a logical thought would be to leverage the advantages of unicast for the purpose of multicast.

Leader-driven multicast (LDM), a new MAC layer protocol, was proposed to better define this concept. Instead of sending a packet to a multicast group address, the sender (or the proxy) could send the packet to one of the MHs (the leader) by unicast. All the other MHs needed to monitor the unicast traffic toward the leader. Some modifications in the *network interface card* (NIC) of each MH was necessary to

support the functionality of LDM. The details of LDM will be discussed in Section 6.2.

The rest of this chapter is organized as follows: First, there will be a summary of the enhancements to the IEEE 802.11 MAC layer protocol proposed by other research groups. The proposed LDM protocol will then be introduced with the analysis of its advantages and drawbacks. After that, the tests to evaluate the performance of LDM comparing to PM will be described. The tests were more complicated when LDM was combined with the application layer error control methods, like FEC or the FEC+EPR combination. The simulation results are shown together with the description of each testing configuration. Since LDM required some modifications to the NIC of each MH, all the LDM-related tests were conducted only in simulation.

6.1 Related Works in MAC Layer Enhancement

The drawbacks of the IEEE 802.11 MAC layer protocol for multicast communications are well known. Besides applying FEC in the higher layers, some researchers have focused on the MAC layer itself. New MAC layer protocols [61–65] have been proposed to enhance the reliability and the efficiency of the multicast communications in WLANs.

Kasera [61] suggested an 802.11 MAC layer enhancement to support reliable multicast. In a multicast session, a leader was selected to handle the RTS/CTS/ACK signaling as in normal unicast. If a non-leader MH detected the corruption of a *data packet* (DATA), it would transmit a NAK immediately to deliberately collide with the ACK sent by the leader. This resulted in the retransmission of the DATA. How-

ever, two issues needed to be addressed before this mechanism could become practical. First, if one of the non-leader MHs had poor reception, then most of the DATA transmissions would be NAK-ed by this MH, which might cause excessive retransmissions in the MAC layer. Second, a non-leader MH might not know when it should transmit the colliding NAK if the boundary of the DATA transmission is garbled due to a large noise burst.

Tang and Gerla [62] proposed a multicast extension to the IEEE 802.11 protocol as well, based on the RTS/CTS signaling. The sender transmitted an RTS frame and waited for the corresponding CTS frame. Any receiving MH in the multicast session replied with a CTS frame when it received the RTS frame. The sender then transmitted the DATA as long as it received a CTS. The reasoning behind this approach was that delivering the DATA to at least one MH was better than waiting for all the MHs to get ready. If multiple CTS frames produced a collision, the sender should have been able to sense the busy channel and therefore transmit the DATA when the channel became idle. This approach allowed the sender to compete for the wireless channel and make a reservation before transmitting the DATA, which improved the reliability of wireless multicast. However, it did not involve ACKs or DATA retransmissions. To achieve reliable multicast, other error control mechanisms in the higher layers were necessary.

Tang and Gerla [63] also extended the previous approach to support MAC layer broadcast in ad hoc networks. The sender transmitted an RTS frame before transmitting the DATA. Any MH that received the RTS frame and wanted to receive the following DATA would respond with a CTS frame. The sender transmitted the DATA

upon receiving the CTS frame. Any MH not receiving the DATA right after the CTS transmission would transmit a NAK to the sender. If the sender did not receive any NAK within a short period of time following the DATA transmission, it could assume that the DATA had been successfully received by all the MHs. Here, the authors made the following assumption: by using *direct sequence spread spectrum* (DSSS), an MH could lock onto the strongest signal even in the presence of other interfering signals. In this way, the protocol could accommodate multiple CTS' or multiple NAKs sent at the same time. The sender would just need to lock onto the strongest CTS or ACK, analyze the situation, and proceed accordingly. However, this scheme is not completely fail-safe. Consider an MH broadcast in an ad hoc network: no other MH within a three-hop radius can broadcast at the same time, or the CTS' from the two broadcast sessions may interfere with each other. The interference can prevent some of the MHs located between the two broadcasting MHs from receiving the broadcast content correctly.

In addition, both Tang and Gerla [64] and Sun et al. [65] discussed reliable broadcast protocol in ad hoc networks. Tang and Gerla [64] proposed to treat each broadcast as multiple unicast transmissions, which were processed the same way as normal unicast transmissions with some minor modification. For each DATA to broadcast, the sender picked out one of its neighboring MHs each time and transmitted the DATA to it by unicast, until each of its neighboring MHs had received the same DATA. If an MH overheard the DATA from a previous unicast transmission toward another MH, it would indicate so in its CTS, so that the sender could save the DATA transmission for this MH and move on to the next neighboring MH.

Sun et al. [65] extended the protocol described in [64] to make it more efficient. The design issue in [65] was how to coordinate the transmissions of the control frames, including RTS, CTS, and ACK. The sender used its RTS frames to sequentially instruct each neighboring MH to transmit a CTS, and used a new frame type called *request for ACK* (RAK) to sequentially poll each receiver to transmit an ACK. To deliver a DATA, the broadcast protocol in [65] required only one contention phase instead of the "one contention phase per neighboring node" as in [64]. However, [65] still needed one RTS/CTS pair and one RAK/ACK pair for each neighboring node.

6.2 LDM Operation and Issues

6.2.1 Overview of LDM

The LDM protocol operates as follows: instead of sending the data stream to a multicast group address, the proxy sends the data stream by unicast to the *leader*, which is a designated MH. Each of the non-leader MHs monitors the unicast traffic from the proxy to the leader, collects all the DATAs in the MAC layer, and reconstructs the data stream.

Since all the DATAs in a WLAN are transmitted in the air, each MH within the wireless cell can receive any DATA transmitted by the access point (AP). Figure 6.1 depicts the operation of LDM.

A minor modification to the NIC of each non-leader MH is necessary, so that when a DATA targeted to a different MAC address is received, the modified NIC can deliver

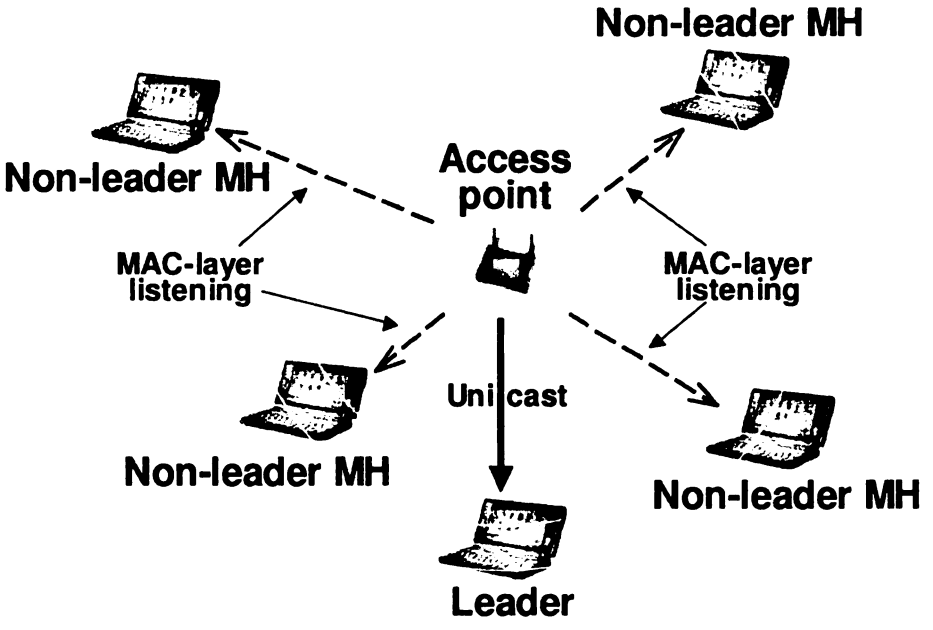


Figure 6.1: LDM operation.

the DATA to the higher layer instead of discarding it. Moreover, if a non-leader MH receives more than one copy of the same DATA due to the DATA retransmission [1], the modified NIC driver must detect the duplication and discard the extra copy.

6.2.2 Advantages of LDM

LDM improves the reception of the leader compared to PM, due to the MAC-layer signaling: the RTS/CTS hand-shaking ensures that the wireless channel is reserved for the following DATA transmission; the retransmission of the DATA in the absence of the ACK helps the leader to recover from the corruption of the previous DATA.

In addition, the reception of the non-leader MHs may also improve. Suppose both the leader and a non-leader MH have received the same corrupted DATA, the AP

[1]This happens when the ACK from the leader is not received by the AP in time

waits for the ACK from the leader to confirm the delivery, but its timer will expire eventually. Then the AP will retransmit the DATA, which gives both the leader and the non-leader MH another chance to recover from the previous corruption. Since PM provides no second chance for any DATA corruption, the reception of the non-leader MH in LDM may be better than that in PM. Simulations are needed to determine how many corrupted DATAs can be recovered due to LDM's residual effect. There may not be many DATA corruptions shared between the leader and other MHs. However, the reception of each non-leader MH in LDM should be no worse than its reception in PM.

Since LDM is a MAC-layer enhancement to the IEEE 802.11 protocol, it can be combined with application layer error control methods (e.g., FEC alone or the FEC+EPR combination) to achieve better quality of the interactive video streaming service. To use the FEC+EPR combination together with LDM, a subset of the non-leader MHs should be appointed as the responders, while the other non-leader, non-responder MHs are the listeners. As described in Section 5.4, each responder can request extra parity packets from the proxy according to its individual experience of packet losses. The relationship among the leader, the responders, and the listeners is illustrated in Figure 6.2.

6.2.3 Drawbacks of LDM

The first drawback of LDM is that it consumes more bandwidth than does PM. In fact, LDM consumes the same bandwidth as unicast does. The extra bandwidth

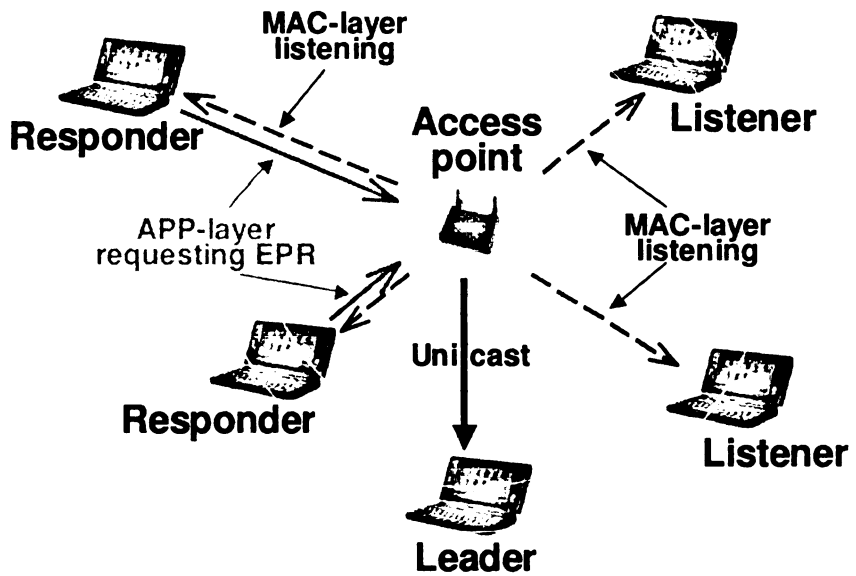


Figure 6.2: LDM combined with FEC+EPR.

is used by the RTS/CTS/ACK signaling and possible DATA retransmissions in the MAC layer. If the DATA retransmission rarely happens, the extra bandwidth consumption required by LDM is negligible comparing to the bandwidth consumption of the video streaming. If, however, the leader's error rate is high, it may incur DATA retransmissions repeatedly and consume much more bandwidth than PM. This may reduce the bandwidth available to the other applications, and in the worst case, cause traffic congestion in the wireless channel, especially when other applications demand high bandwidth consumption at the same time.

Besides consuming more bandwidth, LDM may also introduce long delay to prevent each video frame from making its real-time deadline. Every time an ACK is missing, the AP needs to back-off and wait for a random time before it can retransmit the DATA. If the AP misses the ACKs consecutively, the accumulated delay can

be long enough for the DATA to miss its deadline. On the other hand, PM involves no ACK, thus delay is never an issue when PM is used.

The delay issue introduced by LDM can be even worse if the "automatic data rate dropping" is implemented in the WLAN. Most 802.11 wireless NICs are configured to automatically drop the data rate when multiple responses (ACKs or CTS') are missing consecutively. For example, a regular 802.11b MH may drop the data rate from 11Mbps to 5.5Mbps, then to 2Mbps, and eventually to 1Mbps if no ACK is received in responding to either the DATA or the following retransmissions. Although this mechanism is good at serving the MHs that suffer from low signal-noise ratios (SNRs) [2] by sacrificing the data rate, it makes the interactive video streaming service even more difficult to meet its real-time requirement. A lower data rate implies longer transmission time for each DATA, thus longer delay. In the worst case, the AP cannot get any acknowledgement for a DATA from the leader, so it has to send the DATA repeatedly in the MAC layer. By the time the AP reaches the retry limit and gives up, not only the current frame this DATA belongs to, but also several video frames following the current frame are delayed and have already missed their deadlines.

In addition, allowing MHs to monitor unicast traffic not targeted to themselves may be considered a security risk. However, the same problem exists in a wired Ethernet, where any node attached to the cable can acquire all the DATAs regardless of the destination address. Encryption and application-level authentication may be used to address the security issues. The details of these mechanisms, however, are not discussed in this dissertation.

[2] For example, MHs in the periphery of the wireless cell

6.2.4 Effect of correlated packet corruptions

Theoretically, the benefit of LDM is related to the correlation of the DATA corruptions among MHs in a WLAN. Since each non-leader MH benefits from the leader's residual effect, the more DATA corruptions shared by the leader and a non-leader MH, the more benefit the non-leader MH can get from LDM. Higher correlation in DATA corruptions implies that more DATA corruptions are shared among MHs. On the other hand, if the leader experiences the DATA corruptions independently from a non-leader MH, the DATA retransmissions for the leader are unlikely to cover the non-leader MH's corrupted DATAs. But as long as they share some common DATA corruptions by coincidence, the non-leader MH can still benefit from LDM.

When the time-based model is applied, the correlation of the DATA corruptions among MHs depends on the relative location of each MH. According to the descriptions in Section 4.2, the MHs that are close to each other should have high correlation in their DATA corruptions since they are likely to be affected by the same subset of noise sources. Therefore, the MHs in the leader's close vicinity can benefit more from LDM than the MHs that are far from the leader. On the other hand, if a leader needs to be selected among a group of MHs, a reasonable strategy would be to pick the MH that has the most MHs close to its location.

6.3 Comparing LDM with Pure Multicast

In order to evaluate the performance of LDM, the functionality of LDM was incorporated into MX and the following simulations were conducted. The topology of the

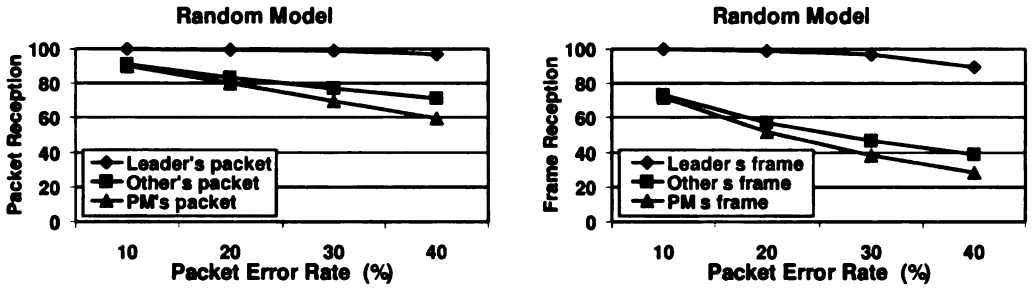
simulated network was similar to the physical testbed shown in Figure 3.1, except that the simulated network consisted of more MHs than the testbed. In the following tests, up to 25 MHs were configured as the video receivers. For the video source of each test, the same MPEG-1 video clip was used as the one in Sections 5.5 and 5.6 (13.7 seconds, 30fps, about 800Kbps). The statistical information of the 412 frames in this video clip is shown in Table 5.1. The packet size was 1100 bytes for the following tests unless otherwise noted.

In this group of the simulations, the proxy sent to multiple MHs the plain video stream, that is, no FEC parity packet was included. In this way, a more precise evaluation was obtained comparing the benefit of LDM to PM by excluding the influence of other error control methods.

6.3.1 Effect of different corruption models

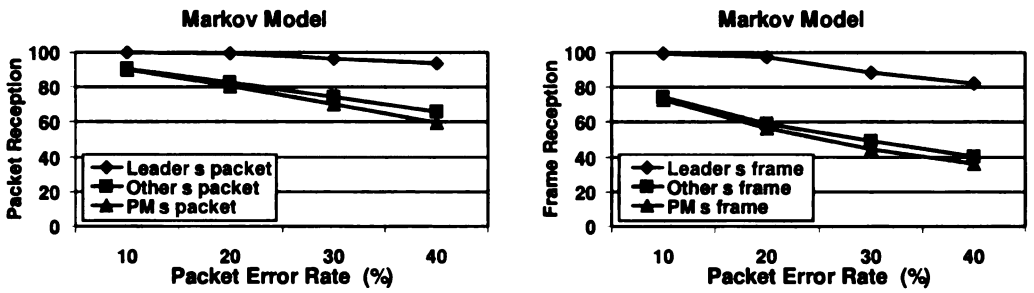
The packet-based corruption model of the wireless channel was varied as well as the parameters of each model to generate different traces of corrupted packets, each with different *packet error rates* (PERs). The packet and frame reception rates of the tests using three different corruption models are shown in Figures 6.3–6.5, respectively. In these tests, at most four retransmissions were allowed for each DATA in the MAC layer. Hence, even the leader in LDM could not receive all the packets correctly under heavy corruptions.

As shown, the reception was different in some degree when different corruption models were applied. However, the same patterns could be recognized regardless of the



(a) Packet Reception Rate (b) Frame Reception Rate

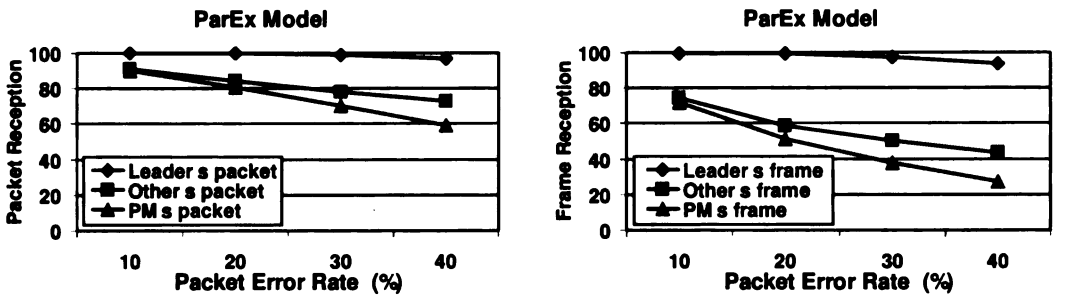
Figure 6.3: LDM vs. PM, Random Model.



(a) Packet Reception Rate (b) Frame Reception Rate

Figure 6.4: LDM vs. PM, Markov Model.

corruption model. First, the reception rate of both the packets and the frames were high for the leader in LDM. The reception rate was usually close to 100%, especially when the PER is less than or equal to 20%. Second, the non-leader MHs in LDM had lower reception rates than the leader, but they always had higher reception rates than the MHs in PM. Third, when the quality of the wireless channel degraded (i.e., the PER increases), the advantage of LDM compared to PM became more apparent, since the non-leader MHs in LDM were more likely to benefit from the leader's residual effect.



(a) Packet Reception Rate

(b) Frame Reception Rate

Figure 6.5: LDM vs. PM, ParEx Model.

6.3.2 Effect of different packet sizes

In order to evaluate the effect of different packet sizes, the CBER model was applied in the following tests. Two channels were defined in different error rates. The *bit error rate* (BER), P_{ber}, is 0.00125% for channel 1, but 0.00375% for channel 2. The packet size was either 1100 bytes or 500 bytes. Table 6.1 shows the corresponding PER (P_{per}) with each combination of P_{ber} and packet size, according to the following formula :

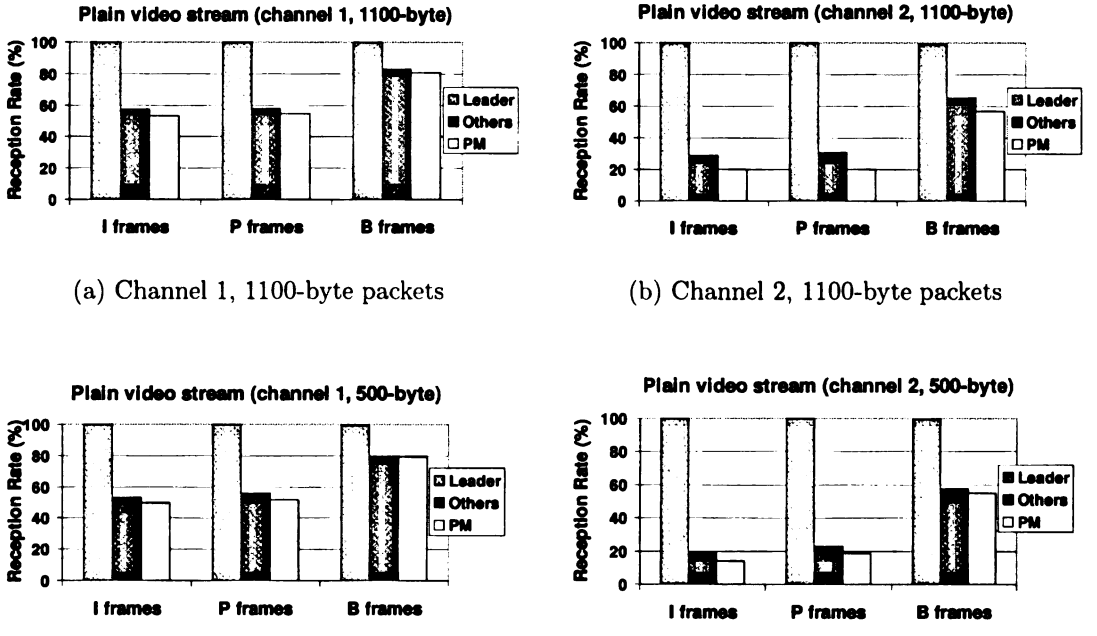
$$P_{per} = 1 - (1 - P_{ber})^{8 \times size}$$

P_{per} =	Channel 1 (P_{ber} = 0.00125%)	Channel 2 (P_{ber} = 0.00375%)
1100-byte packets	10.4%	28.1%
500-byte packets	5.9%	14.9%

Table 6.1: The PER according to the BER and the packet size.

The frame reception rates of I, P, and B frames using different configurations are shown in Figure 6.6. As shown, using 500-byte packets produced lower reception rate than using 1100-byte packets in both channel 1 and channel 2. By excluding any error control method in the application layer, each video frame had exactly the

same bytes of data to transmit, regardless of the packet size. A small packet size had no advantage over large ones, but might introduce more packets for the same video frame, thus more overhead of the transmissions. It could be concluded that plain video streaming should choose the largest packet size possible. If Ethernet was involved in the video streaming, the optimal packet size in the MAC layer should be no more than 1536 bytes, the maximum *protocol data unit* (PDU) defined in IEEE 802.3 [66].



(a) Channel 1, 1100-byte packets

(b) Channel 2, 1100-byte packets

(c) Channel 1, 500-byte packets

(d) Channel 2, 500-byte packets

Figure 6.6: LDM vs. PM, effect of the packet size.

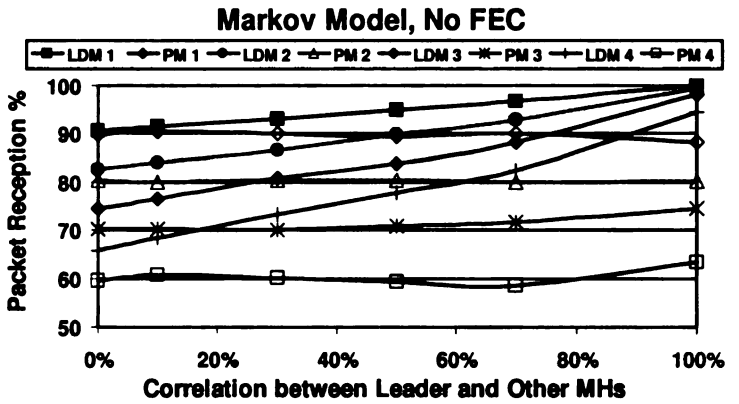
Figure 6.6 also shows that the reception rate of B frames is higher than I or P frames. Since B frames are usually smaller than I or P frames, it gives B frames a better chance to be received without any corruption.

6.3.3 Effect of correlated packet corruptions.

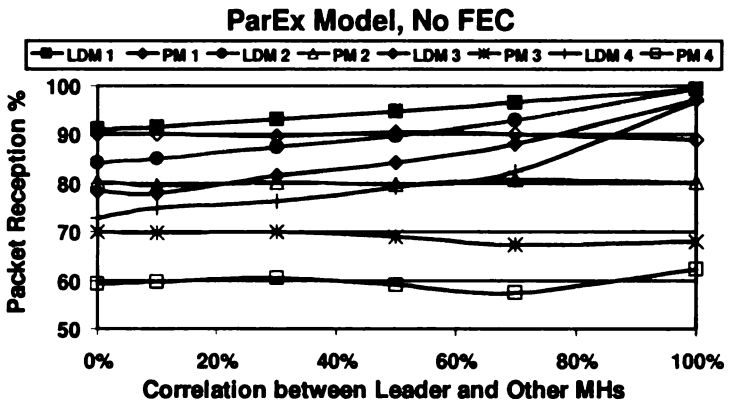
In the previous tests, independent packet corruptions were assumed among the different MHs. If the probability of each MH having a packet corruption was p ($0 \leq p \leq 1$), then the probability of the leader sharing a corrupted packet with a non-leader MH would be p^2. These corrupted packets shared between the leader and the non-leader MHs contributed to the small, though observable advantage of LDM over PM reported in the previous tests.

When a positive correlation existed in the packet corruptions among MHs, more corrupted packets were shared between the leader and the non-leader MHs, which implied a greater advantage of LDM over PM. Figure 6.7 shows the potential improvement of the non-leader MHs' packet reception rate when using LDM instead of PM, with different correlation coefficients in the packet corruptions among MHs. Either the two-state Markov model or the ParEx model was applied in the simulations. Four pairs of LDM and PM curves are shown in each plot, with the PER of 10% (labeled as LDM1 and PM1), 20% (LDM2 and PM2), 30% (LDM3 and PM3), and 40% (LDM4 and PM4), respectively.

As shown in Figure 6.7, when the correlation coefficient approaches 100%, the packet reception rate of the non-leader MHs in LDM also approaches 100%, while the packet reception rate of the MHs in PM remains at the same level as the mean PER of the corruption model. A conclusion drawn from Figure 6.7 is that the advantage of LDM over PM is more apparent when the PER is higher.



(a) Markov Model



(b) ParEx Model

Figure 6.7: LDM vs. PM, effect of correlated packet corruptions.

6.3.4 Effect of the leader's relative location

To evaluate how the reception of non-leader MHs can be affected by the leader's location, the following test was designed to provide simulation of MPEG-4 video streaming service. As shown in Figure 6.8, four MHs, from receiver1 to receiver4 (labeled as R1-R4), are present in the WLAN. Receiver2, receiver3, and receiver4 are close to each other, but are all far from receiver1. The time-based model was applied in this test. Since no simple parameter could describe the error condition of the time-based model, the mean PER of a typical MH was used as a reference when

a plain video stream (with no error control) was sent to this MH by using multicast. The mean PER could be either 15% or 50% in this test.

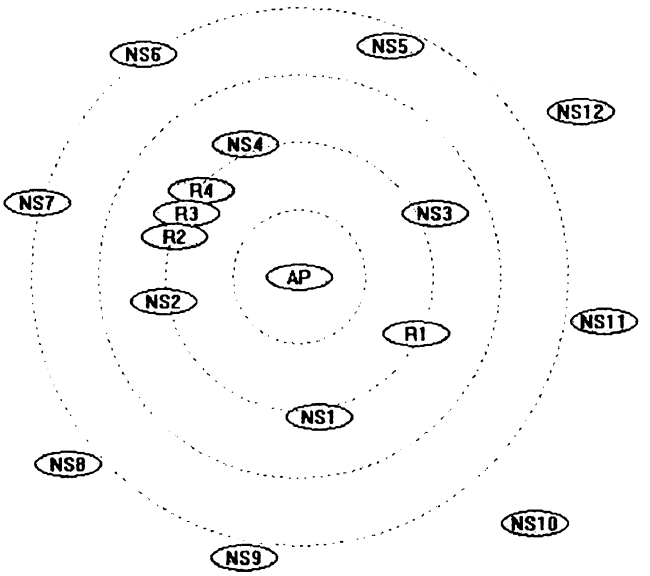
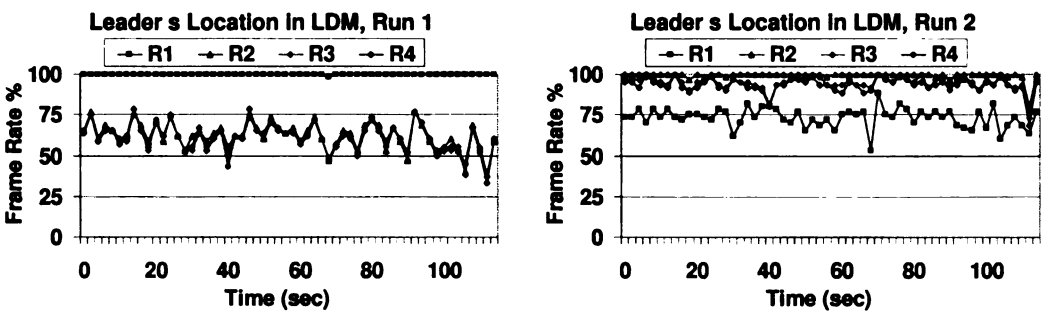


Figure 6.8: Topology of the test for the leader's relative location.

In this test, the MPEG-4 video clip used in previous tests (120 seconds, 30 fps, DivX-encoded) was sent to all the MHs twice, using LDM combined with FEC. Receiver1 is the leader for the first time and receiver2 is the leader for the second time. The FEC rate is 50-30% for I-P frames, respectively. Figure 6.9 shows the frame reception rate of all the MHs in each run.



(a) Run 1, receiver1 is the leader

(b) Run 2, receiver2 is the leader

Figure 6.9: Effect of the leader's relative location.

As shown in Figure 6.9, the leader always has the frame reception rate close to 100%, no matter which MH is the leader. Receiver3 and receiver4 both have higher reception rates when the leader is receiver2 (in run 2) instead of receiver1 (in run 1). This indicates that non-leader MHs benefit more from LDM's residual effect when they are in the closer vicinity of the leader.

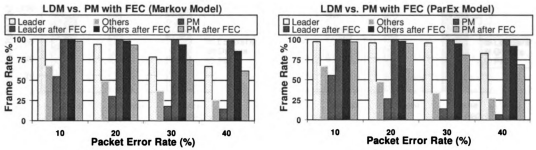
6.4 Combining LDM with FEC

This group of the tests evaluated the performance of streaming the MPEG-1 video clip when both LDM and FEC were used at the same time. The results of not using LDM and/or FEC were also included as reference. If FEC was used, the FEC rate was 60-40-20% for I-P-B frames, respectively.

6.4.1 Effect of different corruption models

Figure 6.10 shows the reception rate of all the I frames during the video streaming, without FEC or after the effect of FEC. Only the I frames were plotted because the reception of I frames was critical to the quality of MPEG-1 video streaming. A corrupted I frame was likely to prevent all the following (P or B) frames from being decoded properly, until the next I frame was successfully received. Both the two-state Markov model and the ParEx model were applied in the following simulations. The correlation coefficient of the packet corruptions between the leader and any non-leader MH was 30%. The packet size was 1100 bytes.

As shown in Figure 6.10, the I-frame reception rate of the leader in LDM is higher



(a) Markov Model (b) ParEx Model

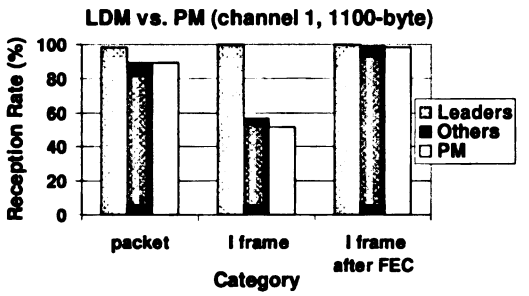
Figure 6.10: Reception rate of I frames as the effect of LDM+FEC.

than that of non-leader MHs in LDM for all the tests. Non-leader MHs in LDM always have higher I-frame reception rates than that of MHs in PM. The advantage of LDM over PM is more apparent when the channel quality degrades. Of particular note is the very high I frame reception rate after FEC for non-leader MHs: the reception rate is over 90% in all the testing configurations except when the two-state Markov model is applied and the PER is 40% [3]. Clearly, the combination of LDM and FEC is an effective mechanism for error control.

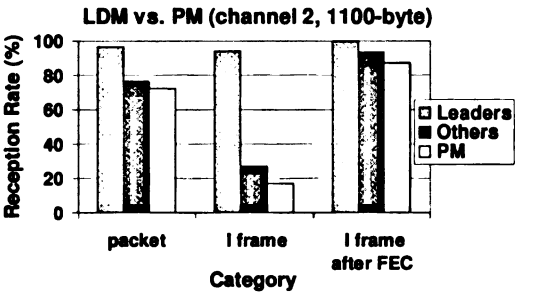
6.4.2 Effect of different packet sizes

In the next test, packet size was examined to determine if it may affect the performance of the LDM+FEC combination. The CBER model was applied the same way as in Subsection 6.3.2. The packet size was either 1100 bytes or 500 bytes. Figure 6.11 shows the general packet reception rate, the I-frame reception rate without FEC, and the I-frame reception rate after FEC of the tests with different channel error conditions and different packet sizes.

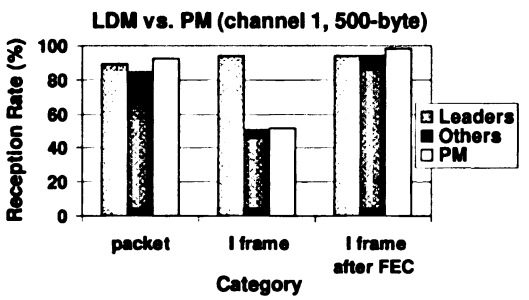
[3]The I-frame reception rate is 86% in this case.



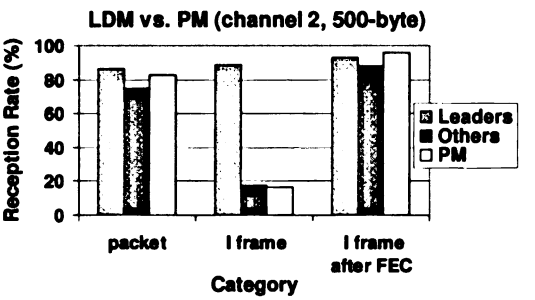
(a) Channel 1, 1100-byte packets



(b) Channel 2, 1100-byte packets



(c) Channel 1, 500-byte packets



(d) Channel 2, 500-byte packets

Figure 6.11: LDM+FEC, effect of the packet size.

As shown in Figure 6.11, smaller packet size increased the frame reception rate in PM if FEC was used. The I-frame reception rate after FEC was higher when a smaller packet size was used. By using smaller packets, the number of the data packets for each frame was bigger and so was the number of the parity packets, which was in direct proportion to the number of the data packets. When the CBER model was applied, each bit error was usually translated into a corrupted packet regardless of the packet size. This implies that the number of the corrupted packets was constant. Therefore the packet corruptions were more likely to be recovered with more parity packets available in each FEC group.

In LDM, however, a smaller packet size might not improve the frame reception

rate when combined with FEC. When the packet size was 500 bytes, as shown in Figure 6.11(c,d), non-leader MHs in LDM actually had a lower I-frame reception rate than the MHs in PM. This was probably due to the excessive bandwidth consumption of LDM, such that the wireless channel was too crowded sometimes. In some cases, such as when using a small packet size with a high-BER channel condition, the available bandwidth of the WLAN might be sufficient for only a PM session, but not for a LDM session with the same video streaming. In such cases, LDM suffered from channel congestion and the advantage of LDM over PM was compromised.

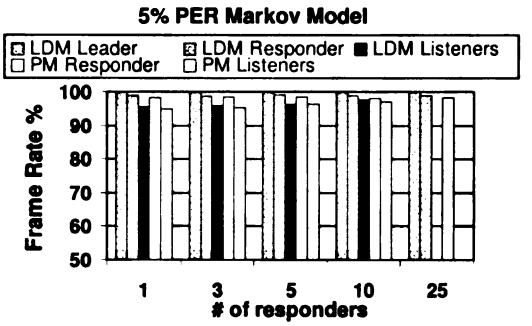
6.5 Combining LDM with FEC and EPR

The next group of simulations were conducted to determine whether or not EPR could improve the reception of the video streaming service when combined with LDM. An FEC rate of 60-60-0% was used for I-P-B frames of the MPEG-1 video clip, and the number of the responders in EPR varied from 1 to 25. The packet size was 1100 bytes for the following tests unless otherwise noted.

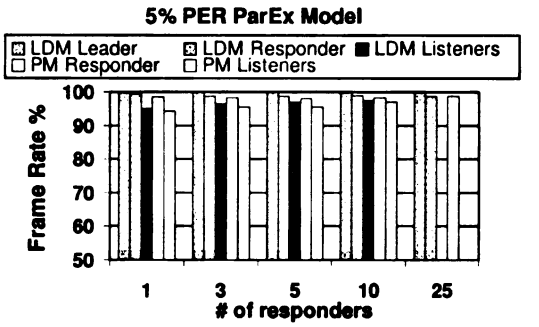
6.5.1 Effect of different corruption models

Figures 6.12 – 6.14 show the frame reception rate corresponding to different numbers of responders, different corruption models, and different PERs. As shown, the frame reception rate of the listeners in PM improved steadily when the number of the responders increased. On the other hand, the frame reception rate of the leader in LDM improved when the number of responders increased, until it reached a specific

number (10 for 5% and 20% PER, 5 for 40% PER). After the number of the responders reached this specific number, increasing the number of responders no longer improved the reception of the listeners.

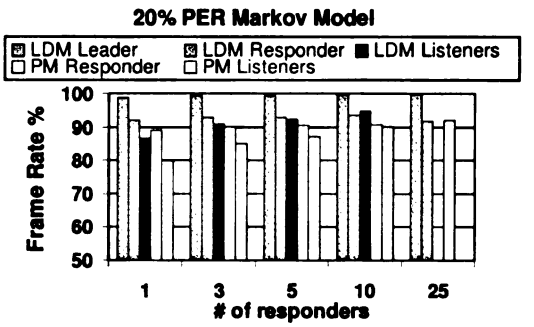


(a) 5% PER, Markov model

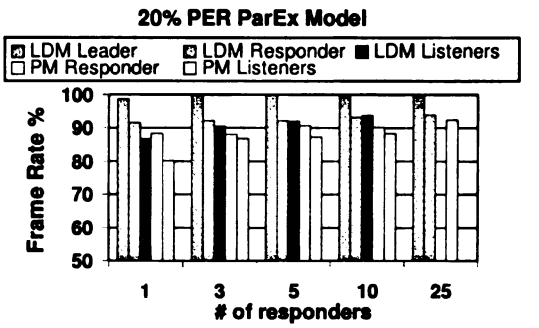


(b) 5% PER, ParEx model

Figure 6.12: LDM+FEC+EPR, 5% PER.



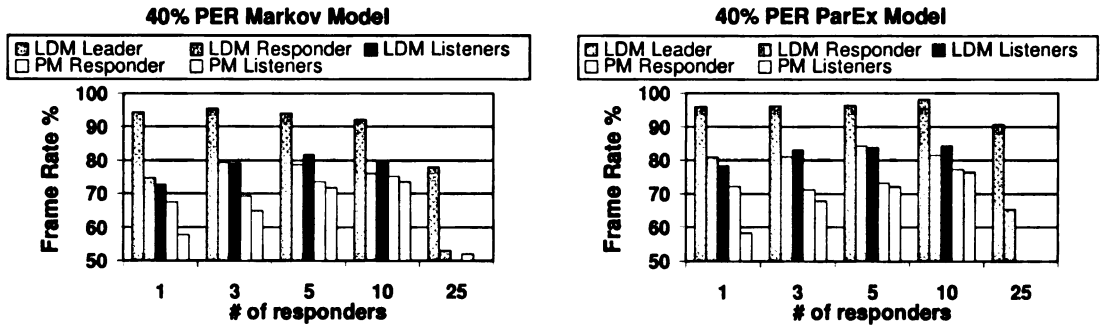
(a) 20% PER, Markov model



(b) 20% PER, ParEx model

Figure 6.13: LDM+FEC+EPR, 20% PER.

Moreover, although the responders were supposed to have better reception than the listeners, Figure 6.13 (a) and Figure 6.14 (a,b) show that the responders in LDM might have worse reception than the listeners if the number of the responders was large enough (10 in Figure 6.13 (a) and 6.14 (b), 5 in Figure 6.14 (a)). This is similar to the results in PM as described in Subsection 5.8.2. Whether or not LDM is used, the responders might suffer from the conflicts between receiving the forward data



(a) 40% PER, Markov model

(b) 40% PER, ParEx model

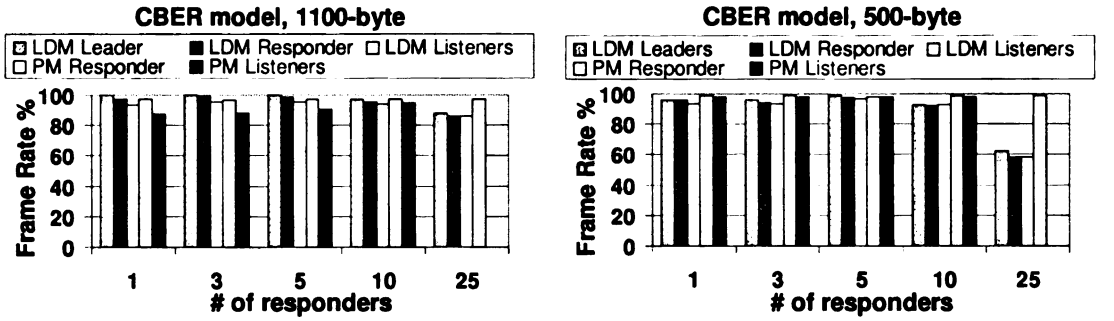
Figure 6.14: LDM+FEC+EPR, 40% PER.

stream and sending the backward requests. When the network was approaching its saturation threshold, collisions were more likely to occur. If the request of a responder collided with a data packet from the video streaming, the responder not only missed the data packet, but also risked not getting any extra parity packet since its request was corrupted due to the collision. Therefore, the reception of the responders might be worse than that of the listeners.

6.5.2 Effect of different packet sizes

Once again, the CBER model was applied to evaluate the effect of different packet sizes. The BER is 0.00375% in the following test. Figure 6.15 shows the reception rate of I frames in LDM and in PM. The packet size can be either 1100 bytes or 500 bytes.

As shown in Figure 6.15, smaller packets may have negative impact on the I-frame reception rate, especially when the number of the responders is as large as 25. This indicates that if LDM is to be combined with EPR, a large packet size is more favorable than the small one.



(a) 1100-byte packets

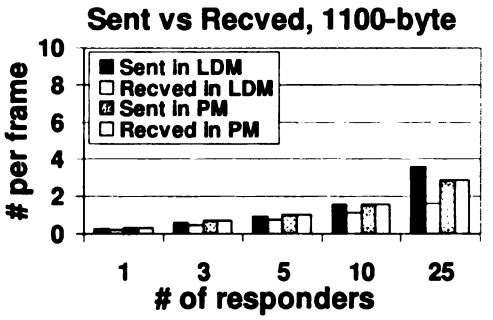
(b) 500-byte packets

Figure 6.15: LDM+FEC+EPR, effect of the packet size.

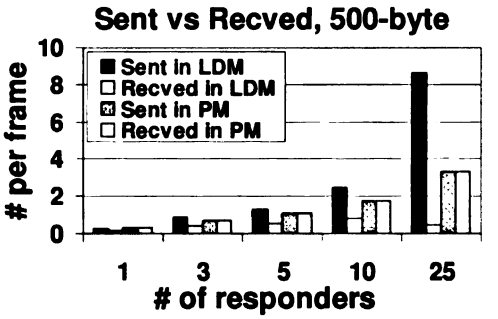
6.5.3 Analysis of the backward traffic

It is also shown in Figure 6.15 that LDM cannot outperform PM when EPR is used with too many MHs being the responders. To understand the reason, the previous tests were repeated and the total number of the extra parity requests were recorded that are (1) sent by any responder, (2) received by the proxy, and (3) suppressed by any responder after overhearing the same or stronger request from another responder. Figure 6.16 shows the number of such requests in LDM and in PM, normalized to the number of the video frames.

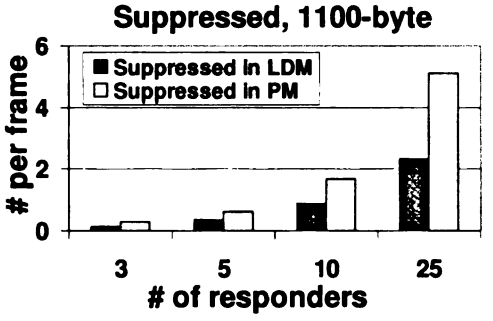
As shown in Figure 6.16(a,b), when the number of the responders increased, the number of the corrupted requests in LDM (i.e., the difference between 'sent' and 'received') also increased, probably due to the increasing collisions between the requests and the forward video streaming traffic. The number of the corrupted requests was even larger when a smaller packet size was used. On the other hand, all the requests in PM were successfully received by the proxy, which was quite a surprising result. Moreover, LDM had more requests to send than PM when the number of the responders was greater than or equal to 10. This suggests that more data packets were



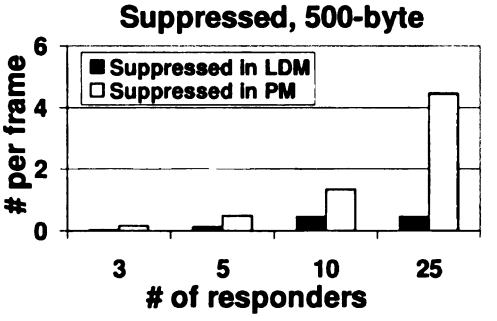
(a) Sent vs. Received, 1100-byte packets



(b) Sent vs. Received, 500-byte packets



(c) Suppressed, 1100-byte



(d) Suppressed, 500-byte

Figure 6.16: LDM+FEC+EPR, number of the requests.

corrupted in the forward traffic, again due to the increasing collisions.

Figure 6.16 (c) and (d) confirm the performance degradation in LDM. Although more requests were sent by the responders when LDM was used, only a small portion could be recognized and used by other responders to suppress their own requests. Therefore, it could be conjectured that when the traffic congestion occurred in the WLAN, most of the requests were either corrupted or severely delayed, such that they could not contribute to the video streaming service in any way. Instead, requests introduced negative effects, such as bandwidth consumption and collisions.

6.6 Summary

This chapter proposed leader-driven multicast for improving the reliability of the multicast communications in WLANs.

As an enhancement to IEEE 802.11 MAC layer protocol, LDM improves the general reception of both the leader and the non-leader MHs. The leader's improvement is more significant than that of the non-leader MHs. The improvement of the non-leader MHs is more apparent when the PER is high or when a high correlation exists in packet corruptions between the leader and the non-leader MHs. Based on the assumption that the MHs close to each other usually have high correlations in their corrupted packets, it is more beneficial in LDM to select the leader from a group of MHs clustered around the same location than to select an individual MH that is distant from all the other MHs.

Since LDM is a MAC-layer solution, it can be combined with application layer error control methods, such as FEC and EPR, to further improve the quality of the interactive video streaming. When combined with EPR, however, only a small number of the MHs should be appointed as the responders, or the wireless channel may experience traffic congestion, and the advantage of LDM over PM may diminish or even disappear. For the configurations investigated, the simulation results suggested that the number of the responders should be between 3 and 5, depending on the error rate and the packet size.

Chapter 7

Adaptive Strategies

In Chapters 5 and 6, both the application layer and MAC-layer solutions were discussed for the interactive video streaming in WLANs. However, all the solutions discussed so far involve only non-adaptive error control methods. This chapter focuses on the adaptive strategies that can be applied to adjust error control methods at run time.

7.1 Methods of Adaptation

Three basic error control methods (FEC, EPR, LDM) and their variations were combined to create five adaptive strategies for the service of wireless multicast video streaming.

7.1.1 Adaptive FEC

In adaptive FEC, the redundancy rate applied in FEC is dynamically adjusted to achieve high frame reception rate while minimizing the bandwidth consumption. The redundancy rate should be adjusted according to the estimation of each MH's *packet error rate* (PER) in the near future. If the PER is expected to be high, more redundancy is needed to recover from the additional packet losses. On the other hand, if the PER will be low, less redundancy is favored to save bandwidth while still providing adequate coverage for the lost packets.

7.1.2 Adaptive FEC + EPR

Adaptive FEC can be combined with EPR, although the adaptation policy needs some minor adjustment. According to the earlier experiments of this study, EPR is more effective and more efficient than FEC with high redundancy in recovering from large error bursts. Indeed, FEC is useless when the error burst can wipe out an entire FEC group. In an environment where large error bursts occur frequently, FEC with a low FEC rate combined with EPR may be more beneficial to the wireless video streaming service than FEC alone with a high FEC rate, yet at a lower bandwidth cost.

An ideal solution is to estimate the PER that is related only to the short error bursts in the near future. In this way, the adaptive FEC strategy can ignore all the large error bursts and adjust the FEC rate based only on the frequency of the short error bursts. All the lost packets caused by the large error bursts depend on the EPR

mechanism for recovery.

7.1.3 Multiple EPR

In the original design of EPR, each responder could request extra parity packets at most once for each FEC group. As described in Subsection 5.7.1, a delay of 80 milliseconds was introduced to absorb the overhead of sending the EPR and receiving the extra parity packets. However, it was discovered, in most cases, that the extra parity packets arrived only 10-20 milliseconds after the packets were initially sent for each FEC group. With careful scheduling, more than one request could be accommodated for each FEC group before the playback deadline expired. In fact, up to four requests were accommodated within the 80-millisecond delay allowance when multiple EPR was enabled in simulation tests.

In the strategy of *multiple EPR*, each additional request is invoked, either by the detection of the lost parity packets that are in responding to the previous request, or by the expiration of a 20-millisecond timer. This approach improves the reception of the video streaming to any MH at the cost of additional bandwidth consumption and possible channel congestion. To prevent the NAK implosion, global NAK suppression is adopted the same way as in the strategy of regular EPR.

7.1.4 Dynamic Responder in EPR

In contrast to assigning a fixed subset of MHs to act as the responders in EPR, each MH in the *dynamic responder* strategy can be added to or removed from the

responder set [1], depending on each MH's changing condition. In this strategy, the responder set is dynamic.

Since each MH can move freely within the area, a responder may occasionally move to a location where it should no longer act as the responder. For example, the PER of a responder at the periphery of the wireless cell may be extremely high, such that the responder requests extra parity packets for almost every FEC group. In the worst case, this responder cannot benefit from the EPR due to its high PER, but has to introduce extra bandwidth consumption in its attempt to request the extra parity packets for most of the FEC groups.

On the other hand, a responder close to the AP may experience little or no packet loss. While it does not affect the quality of the video streaming service in any way, it does not help either. If the total number of the responders is configured to be constant for some reason, the unhelpful responders should be replaced with the listeners that experience moderate packet losses to improve the general quality of the video streaming service for all the MHs.

7.1.5 Dynamic Leader in LDM

Similar to the strategy of dynamic responder, the identity of the leader in LDM can be dynamic, in contrast to having a fixed MH as the leader for the whole time. If the original leader is no longer suitable for the leader's responsibility, switching the leader to another MH may improve the quality of the video streaming service to all

[1] which is a subset of all the MHs.

the MHs.

For example, if the PER of the original leader is extremely high, the DATA retransmissions at the MAC layer may consume too much bandwidth and even cause traffic congestion. On the other hand, if the PER of the original leader is too low, the leader may not provide any residual benefit to any non-leader MH.

7.2 Information to Coordinate Adaptations

With so many strategies and error control methods applicable to the interactive video streaming service in WLANs, the key issue here was when and how to adapt the error control strategy. The proxy appeared to be the best place to make such decisions.

When an adaptive strategy is applied, all the MHs need to report statistical information to the proxy periodically. The proxy, then, can adjust the streaming strategy at run time based on each MH's information. The responders can piggyback the statistical information in their requests for the extra parity packets, but each listener has to notify the proxy its statistical information by sending a special

This section discusses the types of the statistical information that may be collected by the proxy to make adaptive decisions.

7.2.1 Average signal strength

Every time a packet is received, the signal strength associated with the packet is known by the MH. Although the strength of each signal source is not always constant, it fluctuates only within a small range. Therefore, the changing trend of each MH's

average signal strength, over a period of time, can be used to estimate roughly the PER this MH is expected to have in the near future.

When the AP is the signal source, two thresholds of the average signal strength can be used to determine if an MH is too far from, or too close to, the AP. If the average signal strength of the streaming packets is below the lower threshold, this MH is likely to lose most of the packets, either in the past or in the future. Therefore, the proxy should exclude all the information this MH may provide when it makes an adaptive decision. On the other hand, if the signal strength is above the upper threshold, this MH is likely to receive all the packets successfully. While not causing any trouble for the streaming service to other MHs, it cannot contribute anything to the streaming service and should not be considered as a potential EPR responder or as a potential LDM leader.

7.2.2 Packet error rate (PER)

Each MH can easily calculate the PER over a period of time as long as each packet includes a unique sequence number in its header. If all the packet losses are evenly distributed, the calculated PER, p ($0 \leq p \leq 1$), can be used to determine the optimal FEC rate for this MH as follows:

$$R_{FEC} = \lceil \frac{p}{1-p} \rceil$$

Consider a PER of $p = 50\%$: To produce the best quality of the video streaming service, R_{FEC} needs to be at least 100%, which means the number of the parity

packets should be no less than the number of the original data packets. With half of the packets lost, each MH can still reconstruct each FEC group by FEC decoding.

However, most packet losses are not evenly distributed in the WLANs, but occur in clustered error bursts. FEC alone cannot recover from any large error bursts that wipe out all the packets from the same FEC group. Therefore, the packet losses caused by the large error bursts [2] were excluded in calculating the MH's PER.

7.2.3 Frame reception log

Each MH can maintain a record of the video frames recently received by using a flag for each frame indicating whether or not the frame is received without error, that it requires FEC or EPR to recover from packet losses, or that it is not successfully received before the playback deadline expires. Such information can be analyzed to decide whether or not the most recent adaptation helps the situation. If not, it could be that the adaptive strategy is not working well under the circumstances or the condition of the MH in question has recently changed due to its physical movement. Either way, the proxy can take this information into account when making any new adjustment.

7.3 Effect of Adaptive FEC+EPR

The rest of this chapter is devoted to evaluate the performance of the adaptive strategies by conducting simulations. All of the following tests use the same MPEG-4 video

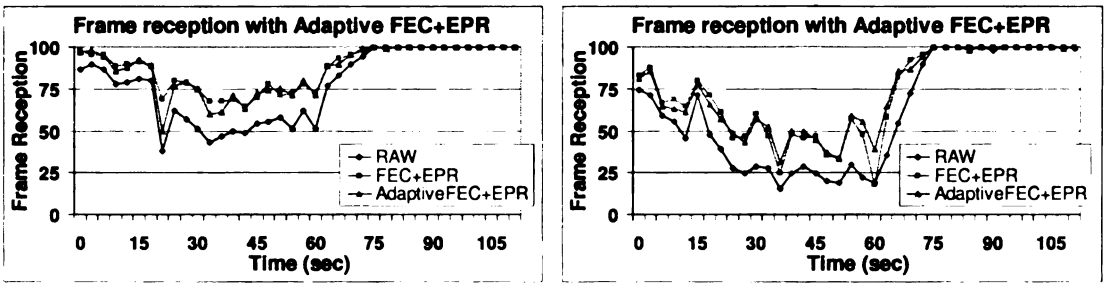
[2] We consider the loss of ten or more packets consecutively as the result of a large error burst

clip previously used in Section 5.7 and 5.8 (120 seconds, 30 fps, DivX-encoded). The time-based model is applied in all of the simulations. To avoid drawing the conclusion from one atypical trial run, five runs were conducted for each testing configuration, each run with different seed value to initialize the random number generator.

The advantage of the adaptive FEC strategy is that it consumes less bandwidth when the PER is low (e.g., the MH is close to the AP) and uses more redundancy to achieve better reception when the PER is high (e.g., the MH is away from the AP). In the following tests, the FEC rate was adjusted according to the information of both the signal strength and the PER collected from each MH.

7.3.1 Using single EPR

Each MH in the following tests was configured to move randomly in the area. The moving style of each MH was as following: each node moved in one direction until it arrived at the destination and stayed there until the next time it moved again. To understand the movement of each MH and how it affected the result, an extra test was conducted by sending the plain video stream (i.e., with no error control) to the MHs, with each MH maintaining the same moving behavior throughout the test. Each MH's frame reception rate of the plain video streaming was referred to as its *raw frame reception rate.* In the following tests, the raw frame reception rate of each MH gave us a general idea of how far the MH was away from the AP. The raw frame reception rate was high when the MH was close to the AP, but might be very low when the MH was far away from the AP.



(a) Mean PER 15%

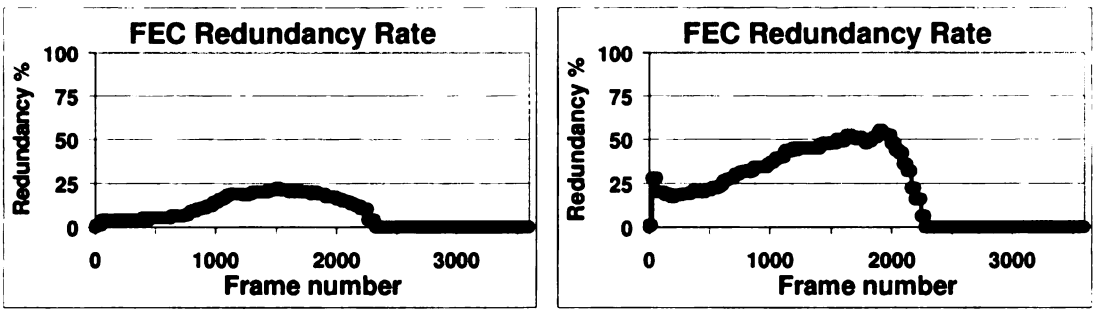
(b) Mean PER 50%

Figure 7.1: Frame reception rate of one moving MH, adaptive FEC+EPR.

To illustrate the difference between the adaptive strategy and the non-adaptive error control methods, the result of another test was used as a reference when both FEC and EPR were used. According to the earlier results described in Chapters 5 and 6, the FEC+EPR combination was considered the most effective non-adaptive mechanism for providing the interactive video streaming service in WLANs. When FEC+EPR was activated in the test, a fixed FEC rate of 50-30% was used for I-P frames, respectively.

Figure 7.1 shows the frame reception rate of the one and only MH involved in the video streaming service in the WLAN. Different error conditions were applied to see how it affected the results. The mean PER of a typical MH at a fixed location was used to describe the error condition when the time-based model is applied in simulation. The mean PER was either 15% or 50%. The responder could request the extra parity packets at most once for each video frame, and the delay allowance for the extra parity packets was 80 milliseconds. Figure 7.2 shows how the FEC rate was adjusted by adaptive FEC in either of the tests.

Although the difference of the frame reception rate between adaptive strategy and the non-adaptive approach is not obviously shown in Figure 7.1, the average frame



(a) Mean PER 15% (b) Mean PER 50%

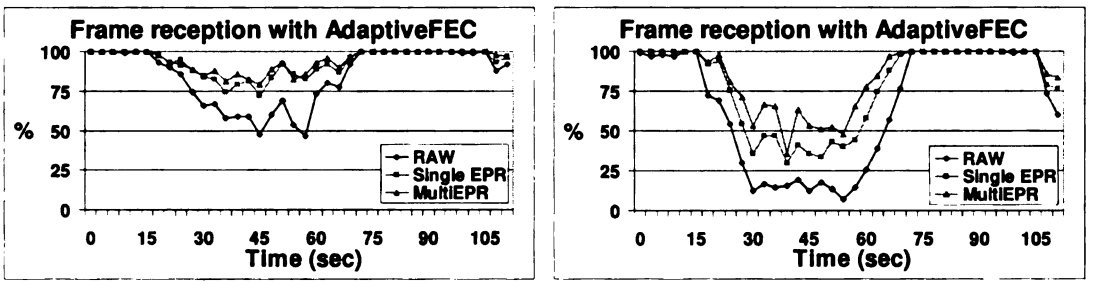
Figure 7.2: FEC rate adaptation of one moving MH, adaptive FEC+EPR.

reception rate of adaptive FEC+EPR is at least the same level as the non-adaptive FEC+EPR combination. Table 7.1 shows the frame reception rate and the bandwidth consumption in all of the tests.

Mean PER 15%	frame rate (%)	total Mbytes	total packets
FEC+EPR	87.8	1.87	20587
Adaptive FEC+EPR	86.8	1.55	17356
reduced by	1.1%	17.3%	15.7%
Mean PER 50%	frame rate (%)	total Mbytes	total packets
FEC+EPR	73.4	2.05	22405
Adaptive FEC+EPR	73.9	1.90	20914
reduced by	-0.7%	7.3%	6.7%

Table 7.1: Bandwidth consumption of using adaptive FEC+EPR.

As shown in Table 7.1, while achieving approximately the same frame reception rate, adaptive FEC+EPR consumes either 7% or 17% less bandwidth than non-adaptive FEC+EPR when the mean PER is either 15% or 50%, averaging over five trial runs for this testing configuration.



(a) Mean PER 15% (b) Mean PER 50%

Figure 7.3: Multiple EPR vs. single EPR, when combined with adaptive FEC.

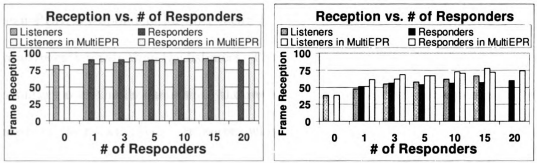
7.3.2 Using multiple EPR

Figure 7.3 shows the result of one MH out of the five moving MHs involved in the video streaming service in the same WLAN. All the five MHs are responders that can send either single EPR or multiple EPRs for each video frame. As shown, multiple EPR improves the frame reception rate over single EPR, and the improvement is more apparent when the PER is higher.

Mean PER 15%	frame rate (%)	total Mbytes	total packets
Single EPR	88.9	1.91	21011
Multiple EPR	91.2	2.09	22791
increased by	2.6%	9.3%	8.5%
Mean PER 50%	frame rate (%)	total Mbytes	total packets
Single EPR	68.9	2.88	30676
Multiple EPR	76.5	3.59	37833
increased by	11%	24.8%	23.3%

Table 7.2: Bandwidth consumption of using adaptive FEC+multiple EPR.

Table 7.2 shows the frame reception rate and the bandwidth consumption in all of the tests. According to the results averaged over five trial runs, multiple EPR improved the frame reception rate by 3% compared to single EPR, at the cost of 9% extra bandwidth consumption when the mean PER was 15%. The improvement of



(a) Mean PER 15%

(b) Mean PER 50%

Figure 7.4: Reception vs. the number of the responders.

the frame reception rate became 11% when the mean PER was 50%, at the cost of 25% extra bandwidth consumption.

7.3.3 Effect of different numbers of responders

To evaluate how different numbers of the responders can affect the performance of either single EPR or multiple EPR, 20 MHs were inserted into the same WLAN as the audience of the interactive video streaming service. Each MH was positioned to be stationary at about the same distance from the AP, but at different directions from each other. The number of the responders varied from 0 (no responder) to 20 (all the MHs are responders). Figure 7.4 shows the average frame reception rate of the responders and the listeners in each test.

Also, Figure 7.4 shows that the frame reception rate of either the listeners or the responders improved when the number of the responders increased, and the improvement was more apparent if the PER was higher. Moreover, the performance of multiple EPR was always better than that of single EPR, as expected.

An interesting point of Figure 7.4 is that at some point (10 responders for 15% mean PER, 5 responders for 50% mean PER), the responders started to have lower frame reception rate than the listeners. Theoretically, the responders should have had advantage over the listeners since the responders could request extra parity packets based on their own needs. However, when the wireless channel was crowded, the feedback sent by the responders might collide with the forward video streaming traffic. Although the general frame reception rate of all the MHs increased when more responders were used, it might not be worth the increasing collisions and the possible congestion in the wireless channel. Therefore, the total number of the responders must be selected carefully, and there should never be as many as the total number of the MHs when a large group of MHs are involved.

7.3.4 Effect of dynamic responder

This group of the tests investigated how the dynamic responder strategy could help the interactive video streaming service. Three MHs, *receiver1*, *receiver2*, and *receiver3* were present in a WLAN, receiving the multicast video stream. Only receiver1 was the responder. Both receiver2 and receiver3 were stationary listeners throughout the test. Receiver1 stayed at its original location for the first 50 seconds, then started moving away from the AP until it reached the periphery of the wireless cell and stayed there till the end of the test. The responder was allowed to use multiple EPRs for each video frame.

The same test was conducted twice, with Receiver1 being the fixed responder

in the first time. In the second time, the dynamic responder strategy was applied, therefore receiver1 could stop acting as the responder when it was sufficiently far away from the AP. Table 7.3 shows the difference in frame reception rate and bandwidth consumption between the dynamic responder strategy and the non-adaptive EPR approach.

	responder (%)	listeners (%)	total bytes	total packets
dynamic responder	47.9	93.1	17803896	19685
non-adaptive EPR	49.8	95.1	24319557	26212
increased by	3.9%	2.1%	36.6%	33.2%

Table 7.3: Effect of the dynamic responder in EPR.

As shown, using non-adaptive EPR instead of dynamic responder consumed more than 30% of extra bandwidth in responding to receiver1's excessive requests for the extra parity packets. However, the frame reception rate of receiver1 improved only 4%. This was because most of the extra parity packets could not be received by receiver1 without corruption when it was away from the AP. As for the two listeners, receiver2 and receiver3, the frame reception rate of each listener improved only 2%, because most of the extra parity packets requested by receiver1 were not needed by either receiver2 or receiver3. Although it may be acceptable in some extreme cases to achieve 4% (or 2%) of the improvement in frame reception at the cost of 30% in extra bandwidth consumption, the advantage of non-adaptive EPR over dynamic responder was not considered to be worth the cost in most cases.

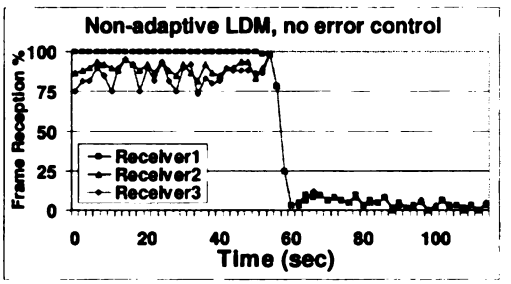
When the strategy of dynamic responder was applied, receiver1 deactivated its responder functionality when it was sufficiently away from the AP, which achieved

a better tradeoff between the bandwidth consumption and the video quality. Furthermore, if either of the listeners became a responder when receiver1 stopped acting as the responder, the frame reception rate of either receiver2 or receiver3 might be further improved at a reasonable cost in terms of bandwidth consumption.

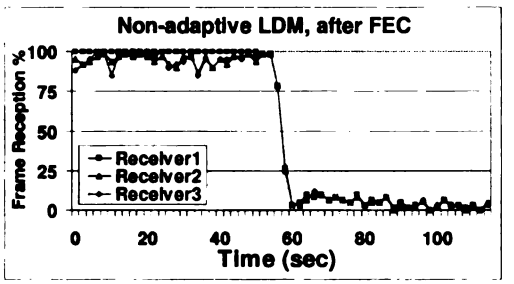
7.4 Dynamic Leader in LDM

In contrast to having a fixed leader, the role of the leader could be switched among MHs when a dynamic leader was applied. The following tests were conducted to evaluate the effect of the dynamic leader strategy. The MPEG-4 video clip (120 seconds, 30 fps, DivX-encoded) was streamed to three MHs (receiver1, receiver2, and receiver3) in a WLAN by using LDM. Both receiver2 and receiver3 were stationary MHs during the test. Receiver1, on the other hand, moved within the area following different predetermined patterns.

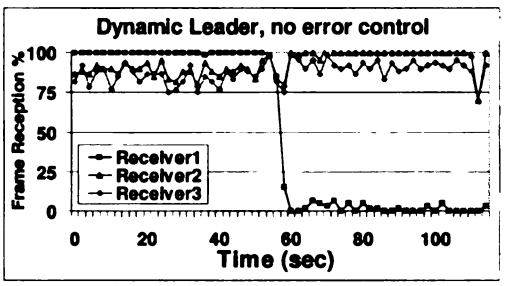
For each of the moving patterns applied to receiver1, the video streaming was conducted twice, both with receiver1 being the leader at the beginning. For the first time, receiver1 was the fixed leader throughout the test. The dynamic leader was used for the second time, so receiver2 could become the leader when receiver1 was no longer suitable for being the leader. It all depended on receiver1's movement to determine when the leader switching occurred in the second run of each testing configuration.



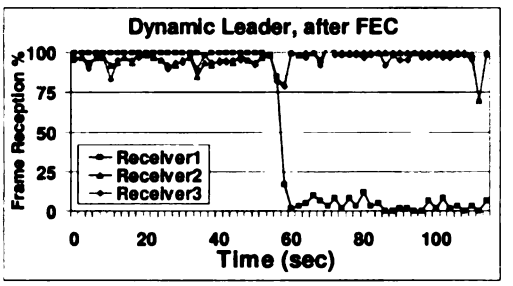
(a) Non-adaptive LDM, no error control



(b) Non-adaptive LDM, after FEC



(c) Dynamic leader, no error control



(d) Dynamic leader, after FEC

Figure 7.5: LDM with 3 MHs. The leader (receiver1) moves away from the AP.

7.4.1 Original leader moving away from AP

For the first test, receiver1 stayed at its original location for the first 50 seconds of the entire 120-second period, then started moving toward the periphery of the wireless cell and stayed there till the end of the test. It was at about 60 seconds from the beginning of the test that the leader was switched from receiver1 to receiver2 when the dynamic leader was applied. Figure 7.5 shows the frame reception rate of all the MHs in all of the tests.

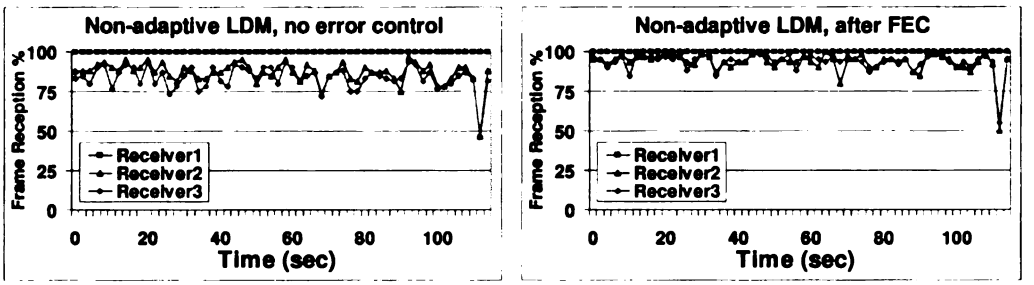
As shown in Figure 7.5 (a) and (b), the non-adaptive LDM produced disastrous results when the leader moved away from the AP. Most of the video frames could not be received on time by any MH, due to the excessive MAC-layer retransmissions and the intolerable long delay they caused. On the other hand, when the dynamic leader

was applied, both receiver2 and receiver3 experienced acceptable reception when the leader was switched from receiver1 to receiver2, as shown in Figure 7.5 (c) and (d). Moreover, the frame reception rate of receiver2, without any error control method, was almost 100% after it became the leader. As for receiver3, the combination of LDM and FEC produced the frame reception rate close to 100% after receiver2 became the leader, which was slightly better than its reception when receiver1 was the leader. This was due to the fact that receiver3 was closer to receiver2 than it was to receiver1.

7.4.2 Original leader moving close to AP

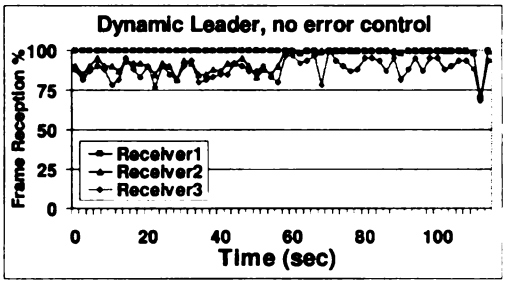
In the next test, receiver1 stayed at its original location for the first 50 seconds, then started moving toward the AP and stayed there till the end of the test. When dynamic leader was applied, the leader switching occurred a few seconds after the receiver1 started moving. Figure 7.6 shows the frame reception rate of all the MHs in all of the tests.

As shown in Figure 7.6, the frame reception rate of receiver1 was always 100%. The frame reception rate of either receiver2 or receiver3 improved after receiver2 became the leader. Receiver2 benefited from being the leader in LDM. Receiver3's reception improved because of the LDM's additional residual effect, since the new leader (receiver2) experienced more packet corruptions than the former leader (receiver1). Although the wireless channel was more crowded after the leader switching, the extra bandwidth consumption was negligible compared to the bandwidth consumed by the entire video streaming, and was considered acceptable in exchange for

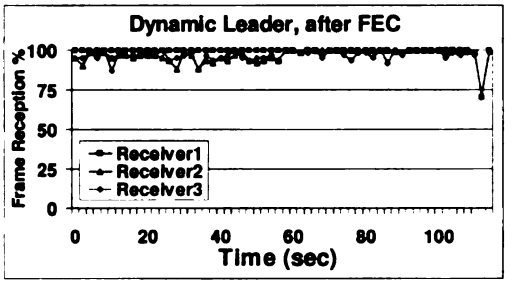


(a) Regular LDM, no error control

(b) Regular LDM, after FEC



(c) Dynamic leader, no error control



(d) Dynamic leader, after FEC

Figure 7.6: LDM with 3 MHs. The leader (receiver1) moving toward the AP.

the improvement in the video quality of both receiver2 and receiver3.

7.5 Summary

This chapter included a discussion of the adaptive strategies in further improving either the quality or the efficiency of the video streaming service in WLANs. The results are summarized as follows.

Adaptive FEC can either save the bandwidth consumption significantly when the noise level is low, or improve the video quality when the noise level is high. Multiple EPR is always more effective than single EPR, at the cost of more bandwidth consumption. Although the video quality may be improved if more responders are used in EPR, it is reasonable to maintain the number of the responders at a low level.

In the dynamic responder strategy, each responder can be deactivated if its PER is too high. This is to prevent the responder from requesting the extra parity packets since most likely the responder cannot benefit from the requests. The video quality may slightly degrade if the number of the responders decreases, but the significant saving of the bandwidth consumption is apparently more desirable. Moreover, some listeners can be activated to act as the responders when some of the responders are deactivated, so that the number of the responders remains the same. If so, the general quality of the interactive video streaming service should be better since the dynamic responder strategy ensures that the most suitable MHs are selected to act as the responders.

The dynamic leader makes it possible to switch the leader in LDM whenever the current leader is no longer suitable to act as the leader. By switching the leader to another MH, the overall performance of the multicast session can be improved significantly, showing that dynamic leader is an effective strategy for LDM adaptation.

Chapter 8

Conclusions

This chapter summarizes the contributions made by this study, followed by the suggested topics that can be further investigated in the future.

8.1 Summary of Dissertation

In this dissertation, the issues related to the interactive video streaming service in WLANs were investigated, with a primary focus on video multicast, where video content must reach multiple MHs at the same time. Both experiments and simulations were conducted in this investigation. For the experiments, a testbed was constructed that involved multiple desktop PCs and multiple laptop computers with wireless connections to the WLAN. For the simulations, the MX simulator was implemented that allowed unmodified applications to be executed atop of a simulated network.

The contributions resulting from this research are as follows.

8.1.1 Time-based model in simulation

In the simulation study, the time-based model was proposed to describe the error behavior in WLANs and was incorporated into the simulations. In the time-based model, each energy source was considered either a signal source or a noise source. Both signals and noises followed the same rule of propagation and energy loss. Also, it was assumed that the BER of an MH depends on the SNR at the MH's location. Although it was complicated and time-consuming to calculate the strength of all the signals and noises at each MH's location at any moment, the time-based model is was adopted as the most accurate model for describing the error behavior in WLANs.

A calibration procedure was conducted to verify the accuracy of the MX simulator and the time-based model. The same tests were conducted both in the real world testbed and in simulation. Although the results of the simulations were not identical to that of the experiments, the distribution of the packet loss bursts in the simulations could be produced the same way as in the experiments, which indicated that both the MX simulator and the time-based model were reasonably accurate compared to conditions in the real-world environment.

8.1.2 FEC-based video error control

Forward error correction (FEC) was commonly used as an error control method in real-time communication. The basic idea of FEC is to introduce redundancy in the data stream, so that receivers may recover some or all of the lost packets without contacting the sender. A block erasure code (one of the FEC techniques that were

used) guaranteed that the same set of the parity packets could be used to recover from any combination of the packet losses, as long as the number of the lost packets was less than or equal to the number of the parity packets. The experiments indicated that FEC was very effective in correcting small burst errors in packet streams, but was ineffective in dealing with large burst errors, which can occur frequently in WLANs.

To further improve the quality of the video streaming service, a proxy-based protocol was proposed that enabled receivers to obtain extra parity packets when FEC alone could not recover from the packet losses. In order to maximize the benefit of FEC, the additional packets sent from the proxy should always be the parity packets generated by the FEC encoder, instead of the original data packets. In this way, one extra parity packet could correct any one-packet loss at any MH. This mechanism was referred to as *extra parity request* (EPR).

The experiments and simulations showed that the FEC+EPR combination could improve the reception of multicast video streaming in WLANs considerably, with a final frame reception rate close to 100% in some cases. Even when severe interference was present in the WLAN, the FEC+EPR combination could still provide noticeable improvement.

8.1.3 Leader-driven multicast

In addition to the application layer error control methods, a new MAC-layer enhancement to the IEEE 802.11 protocol was also invetigated. *Leader-driven multicast* (LDM) leverages the advantages of unicast for the purpose of multicast. To be more

specific, the proxy sends a unicast stream to a *leader* (a designated MH), instead of sending the multicast stream to a group address. Each non-leader MH monitors the unicast stream sent to the leader, collects the packets, and reconstructs the data stream.

Simulation showed that LDM improved the general reception of both the leader and the non-leader MHs. The leader's improvement was more significant than that of the non-leader MHs. The improvement of the non-leader MHs was more apparent when the PER is high or when a high correlation existed in packet corruptions between the leader and the non-leader MHs. Based on the assumption that the MHs close to each other usually have high correlations in their corrupted packets, it was more beneficial in LDM to select the leader from a group of MHs clustered near the same location than to select an individual MH that was distant from all the other MHs.

Since LDM was a MAC-layer enhancement to improve the reliability of the multicast communications in general, it could be combined with application layer error control methods, such as FEC and EPR, to further improve the quality of the interactive video streaming. When combined with EPR, however, the results showed that only a small number of the MHs should be appointed as the responders, otherwise the wireless channel would have experienced traffic congestion, and the advantage of LDM over PM would have diminished or even disappeared. For the configurations investigated, the simulation results suggested that the number of the responders should be between 3 and 5, depending on the channel conditions and the packet size.

8.1.4 Adaptive strategy

The last part of this study focused on the run-time adaptations. All the available mechanisms for error control were studied individually as well as in combination. The study was conducted first using experiments on the testbed, and then was extended to simulations involving more MHs. Dynamic reconfiguration at run-time was adopted to investigate possible adaptive error control strategies. Each adaptive strategy and its effect are summarized as follows.

Using adaptive FEC can either reduce bandwidth consumption when the noise level is low, or improve the video quality when the noise level is high. Multiple EPR is always more effective than single EPR, at the cost of higher bandwidth consumption. Although the video quality may be improved if more responders are used in EPR, the number of the responders should be maintained at a low level.

In the dynamic responder strategy, a responder should be deactivated if its PER is too high. This action is necessary to prevent the responder from requesting the extra parity packets, since most likely the responder cannot benefit from the requests. The video quality may slightly degrade if the number of the responders decreases, but the significant savings in bandwidth may outweigh this factor. Moreover, some listeners can be activated to serve as responders when other responders are deactivated, so that the number of the responders is constant. In this case, the overall quality of the interactive video streaming service should be better since the dynamic responder strategy ensures that the most suitable MHs are selected to act as responders.

Enabling the leader in LDM to change makes it possible to select a more suitable

leader dynamically, depending on conditions. By switching the leader to another MH, the performance of the multicast session can be improved significantly, showing that the dynamic leader is an effective strategy for LDM adaptation.

8.2 Topics in Future Work

The research results illuminated several related issues that are open to further study.

8.2.1 Refined time-based model

The time-based model proposed in this dissertation simplifies the calculation procedure for the strength of signals and noises in simulation. In the current time-based model, the energy loss of the electromagnetic wave solely depends on the distance it has propagated from the source. This simplification is acceptable if the propagation is in free space. However, if the walls and other objects (e.g., furniture, human beings) are involved, the calculation becomes more complicated.

A more accurate calculation needs to take into account many factors that were not considered here. These factors include energy loss, multipath interference, reflection, diffraction, and scattering, which in turn depend on information such as the architecture of the building, the arrangement of furniture, the movement of humans and their objects, the composition of walls, floors, and so on. In the future, a refined time-based model might be constructed using these factors, to describe the error behavior in WLANs in greater detail.

8.2.2 Experimental evaluation of LDM

In this research, the LDM approach was evaluated only in simulations, but not in real-world experiments. To evaluate the performance of LDM in experiments, the wireless *network interface card* (NIC) of each MH would need to be modified to provide the functionalities LDM demands. After modification, the NIC can allow a received packet to be delivered to the higher layer even if the packet is targeted to a different MAC address. Such packets would be discarded by the unmodified NIC. Moreover, if the NIC receives more than one copy of the same packet targeted to a different MAC address, the modified NIC must able to detect the duplication and discard the extra copy.

A possible way to implement LDM is to modify the NIC driver in an open-source operating system such as Linux. Although all of the programs used in this study are based on Microsoft Windows operating system, porting these programs to Linux should be straightforward.

8.2.3 Further study of adaptive strategies

The investigation of the adaptive strategies introduced in this dissertation is still preliminary. Many topics can be extended based on the studies we have conducted so far. For example, in regard to strategies of dynamic responders and dynamic leaders, how to coordinate the dynamic adaptation can be explored further.

Another area of investigation involves requests in multiple EPR. For now, a responder only has the options of sending either one request or as many requests as

possible, if an FEC group appears to be incomplete due to packet losses. A more complete solution would be to provide options so that each responder can choose according to the other information, such as channel conditions, signal strength, and playback deadline.

Finally, the combinations of adaptive strategies should be investigated more thoroughly. The run-time switching among different adaptive strategies can be beneficial to the overall performance if the current situation demands it. The best solution would be to have the proxy implement run-time adaptation in an autonomous way, based on the information it collects from the MHs. The objective is to provide, at any given point in time, the most effective form of error control for interactive video streaming service in WLANs.

Appendices

Appendix A

MX Description Language

For each simulation, a script file need to be read by the MX simulator upon initialization, so that MX understands the configuration of each component as well as the topology of the entire simulated network. We write this script file in *MX description language* (MXDL). This appendix summarize the syntax and usage of MXDL.

A.1 Comments

A line begins with the character '#' is the *comment* line. All the words in the same line will be ignored.

A.2 For loop

Syntax:

```
FOR <id> = <start_value> TO <end_value>
    # Now the script may use $<id> to retrieve its integer value
```

```
NEXT <id>
```

Remarks: Similar to a normal for loop, all the lines between the `FOR` statement and the `NEXT` statement of the same variable name `<id>` will be repeated until the value of `<id>` changes from `<start_value>` to `<end_value>`. For each loop, the value of `<id>` increases by one.

Currently, `FOR` loop can be nested up to 5 levels. The maximum nested level is configured in the codes of MX (the macro `MAX_FORLOOP_NEST`).

A.3 Domain

Syntax:

```
DOMAIN <domain_name> [CSMA/CD <bandwidth>]|[IEEE_802_11 <lasting_time>]
```

Remarks: This command defines a domain of the simulated network, which will be associated with other components, such as hosts, links, interfaces, and so on. Currently, the type of the domain can be either `CSMA/CD` (for Ethernet) or `IEEE_802_11` (for WLAN).

`<bandwidth>` is the primary bandwidth this domain can use in the unit of *Kbps*. For example, the `<bandwidth>` of a 100Mbps Ethernet should be 100000.

`<lasting_time>` is used to indicate how long all the traffic will last (in the unit of second) in a WLAN simulation. This is necessary for MX to know when the simulation can be terminated.

A.4 Host

Syntax:

```
HOST <host_name> <Router|Repeater|Host> <cpu_speed> [FixDataRate]
    [Moving [<moving_radius>]] [Turning]
```

Remarks: This command defines a host in the simulated network, which can be a computer `Host` or a `Router` that connects two or more domains.

`Repeater` is slightly different from `Router`: a `Repeater` relays all the multicast packets to the other domains it connects to, while a `Router` does not.

`<cpu_speed>` is the CPU frequency of this host in the unit of MHz.

`FixDataRate` is a flag used only in WLANs. If not set, the host may drop its data rate if it sustains heavy losses in a WLAN. Otherwise, the data rate is constant regardless of the error condition.

`Moving` is a flag used only in WLANs. If set, the host can move around the *Access Point* (AP) with the maximum distance of `<moving_radius>` in the unit of centimeter. If no `<moving_radius>` is specified, the default radius is 100cm. The value of `<moving_radius>` can be one of the following special cases:

- If `<moving_radius>` is *zero*, move the host toward the AP and stay there.
- If `<moving_radius>` is *negative*, move the host away from the AP and stay at the periphery of the wireless cell.

`Turning` is a flag used only in WLANs. If set, the host is supposed to change its facing direction from time to time, which may affect its signal reception in the

WLAN, therefore the efficiency of this host in receiving signals may vary. Otherwise, the efficiency is constant.

A.5 Noise

Syntax:

`NOISE <domain_name> <noise_name> PERIODIC|PAREX|SENS_WEST <para_list>`

Remarks: This command defines a noise source located in a wireless domain `<domain_name>`. `<domain_name>` must be defined in a previous `DOMAIN` statement. The type of the noise source can be `PERIODIC`, `PAREX`, or `SENS_WEST`. Each noise type describes the error behavior with its own parameter list, `<para_list>`.

- `PERIODIC` noise source has one noisy period in each cycle time. The `<para_list>` is in the format of `<Cycle_time_in_second> <noise_length_in_millisecond>`. Either parameter can be a real number.

- `PAREX` noise source has its noisy period and noiseless period in turn following the ParEx model. The `<para_list>` has three parameters. The first two parameters are P_a and P_k for the pareto distribution, and the third parameter is λ for the exponential distribution.

- `SENS_WEST` noise source is designed to described the abnormal error behavior observed in our laboratory. It needs no extra parameter.

A.6 Topology of the Graph

Syntax:

`TOPOGRAPH <domain_name> <topofile_name>`

Remarks: This command defines the topology of all the components located in a wireless domain `<domain_name>`. `<domain_name>` must be defined in a previous `DOMAIN` statement.

`<topofile_name>` indicates the name of the file maintaining the location information of the AP, each hosts, and each noise sources in the WLAN.

A.7 Network Interface

Syntax:

`INTERFACE <id> ETHERNET|IEEE_802_11|AIRONET <domain_name>`

`<host_name> <IP_addr> <proc_speed> <in_buffer> <out_buffer>`

Remarks: This command defines a *network interface card* (NIC) associated with a host `<host_name>` and a domain `<domain_name>`. `<domain_name>` and `<host_name>` must be defined in a previous `DOMAIN` and `HOST` statement, respectively.

`<id>` is the integer value of the global ID for this NIC in the simulation.

It turns out that Cisco Aironet has many PHY parameters set differently from the generic `IEEE_802_11` protocol, therefore a separate category, `CISCO_AIRONET`, is created to simulate the physical testbed in our study.

`<IP_addr>` is the IP address of this NIC in the format of "xx.YY.zZ.Aa".

`<Proc_speed>` is the frequency of the I/O processor on this NIC in the unit of

MHz . If set to *zero*, the default speed is 500MHz.

<in_buffer> and <out_buffer> are the sizes of the NIC's I/O buffers in the unit of byte. If both set to *zero*, the default size is 64Kbytes for each buffer.

A.8 Loss Model

Syntax:

```
LOSS <loss_id> RANDOM PACKET|BYTE|TIME <loss_rate>

LOSS <loss_id> TWOSTATE PACKET|TIME <para1> <para2>

LOSS <loss_id> PAREX  PACKET|TIME <para1> <para2> <para3>

LOSS <loss_id> TRACE <loss_style (ignored)> <trace_filename>
```

Remarks: This command defines a loss model that can be applied to an individual link. Since a wireless domain in simulation automatically applies the time-based model, which determines the packet corruptions based on each component's location, all the loss models defined here are for the wired links only.

In the first three loss models, the packet loss can be calculated based on either packet or time. As for the per-byte **RANDOM** model, the **loss_rate** is the equivalent probability of a 1000-byte packet being lost. For the per-time style, each time unit is currently set to be 100usec (0.1msec),

<loss_id> is the integer value of the global ID for this loss model in simulation.

For the **TWOSTATE** markov model, <para1> is $P_{0\to1}$ (bad→good) and <para2> is $P_{1\to0}$ (good→bad).

For the **PAREX** model, <para1> and <para2> are P_a and P_k for the pareto distri-

bution, and <para3> is the λ parameter for the exponential distribution.

A.9 Link

Syntax:

LINK <link_id> CABLE <link_length> <domain_name> <loss_id>

Remarks: This command defines a physical link in a wired domain <domain_name>, with a loss model <loss_id> assigned to it. <domain_name> and <loss_id> must be defined in a previous DOMAIN and LOSS statement, respectively. Since a wireless domain in simulation automatically applies the time-based model, which determines the packet corruptions based on each component's location, all the links defined here are for the wired networks only.

<link_id> is the integer ID for this link. Apparently, different domains should be allowed to use the same <link_id> in the simulation, since the pair of <domain_name> and <link_id> should be unique. However, as it has not been thoroughly tested so far, one should always use a unique <link_id> value for each link as a precautionary manner.

<link_length> is the distance between two ends of the link in the unit of meter. The value can be a real number.

A.10 Attach

Syntax:

```
ATTACH <domain_name> <link_id> <interface_id>
```

Remarks: This command connects a link `<link_id>` to a NIC `<interface_id>` in the domain `<domain_name>`. Neither the domain nor the link is for the wireless networks. `<domain_name>`, `<link_id>`, and `<interface_id>` must be defined in a previous `DOMAIN`, `LINK`, and `INTERFACE` statement, respectively.

A.11 Route

Syntax:

```
ROUTE <host_name> <domain_name> <interface_id> <IP_addr> <subnet_mask>
```

Remarks: This command sets up a routing entry in a router `<host_name>`.

A pair of `<IP_addr>` and `<subnet_mask>` defines a group of addresses. The router will forward any packet came from the domain `<domain_name>` with a matching destination address to the specified NIC `<interface_id>`.

`<host_name>`, `<domain_name>`, and `<interface_id>` must be defined in a previous `HOST`, `DOMAIN`, and `INTERFACE` statement, respectively.

A.12 IO Control

Syntax:

```
IOCTL <domain_name> <interface_id> <command> [<argument>]
```

Remarks: This command provides an interface to control the feature of a specified NIC <interface_id> in the domain <domain_name>. The <domain_name> and the <interface_id> must be defined in a previous DOMAIN and INTERFACE statement, respectively.

This command is not fully implemented yet. Currently, the only option for <command> is '2', to reseed the random number generator of this NIC with integer <argument>. If <argument> is not specified, the default value is 0.

A.13 Property

Syntax:

```
PROPERTY NETIFTRACE <domain_name> <interface_id> <in_tracefile>
    <out_tracefile>
PROPERTY DOMAINTRACE <domain_name> <tracefile>
PROPERTY LINKTRACE <domain_name> <link_id> <tracefile>
```

Remarks: Each command specifies the trace file associated with each network component for the purpose of debugging. Each trace file can be "NUL" (no trace), "CON" (standard output), or a regular file name.

Appendix B

Sample MXDL script

Following is a short sample script file written in MXDL.

```
#
# network configuration file
#


# define two domains, one Wired and one Wireless
DOMAIN Wired CSMA/CD 100000
DOMAIN Wireless IEEE_802_11 125


# define a wired link with zero error rate
LOSS 1 RANDOM PACKET 0
LINK 1 CABLE 10 Wired 1
```

```
# four hosts inside a wired network

HOST Sender HOST 2000

HOST Proxy HOST 2000

HOST Chopin HOST 2000

HOST AccessPoint REPEATER 1000


# define the NIC for each wired host, and connect them with link 1

INTERFACE 1 ETHERNET Wired Sender 35.9.26.100 100 0 0

ATTACH Wired 1 1

INTERFACE 2 ETHERNET Wired Proxy  35.9.26.101 100 0 0

ATTACH Wired 1 2

INTERFACE 1000 ETHERNET Wired Chopin 35.9.26.161 100 0 0

ATTACH Wired 1 1000

INTERFACE 3 ETHERNET Wired AccessPoint 35.9.26.102 100 0 0

ATTACH Wired 1 3


# define the other NIC of AccessPoint, which connects to a wireless network

INTERFACE 4 AIRONET Wireless AccessPoint 35.9.20.160 100 0 0


# set up the route table of AccessPoint

ROUTE AccessPoint Wired 3 35.9.26.0 255.255.255.0

ROUTE AccessPoint Wireless 4 35.9.20.0 255.255.255.0
```

```
# set up twelve noise sources in the wireless network
FOR Y=1 TO 12
NOISE WIRELESS NOISESOURCE$Y PAREX 1.4 .8 0.035
NEXT Y

# specify the topology file used by the wireless network
TOPOGRAPH WIRELESS WLAN_TOPOGRAPH.TOPO

# set up the trace file for debugging purpose
PROPERTY NETIFTRACE Wireless 4 AccessPoint.csv
PROPERTY DOMAINTRACE wireless wireless.csv

# END
```

Bibliography

Bibliography

[1] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications, high-speed physical layer extension in the 2.4ghz band. September 1999.

[2] Wireless Application Protocol – White Paper. June 2000.

[3] IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *ISO/IEC 8802-11:1999(E)*, August 1999.

[4] Brian P. Crow and Jeong Geun Kim. IEEE 802.11 wireless local area networks. *IEEE Communications*, September 1997.

[5] Richard van Nee and Geert Awater. New High-rate Wireless LAN Standards. *IEEE Communications*, December 1999.

[6] Aman Kansal and U B Desai. Mobility Support For Bluetooth Public Access. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 5, pages 725–728, INSPEC Accession Number: 7454584, May 2002.

[7] B. R. Badrinath, A. Bakre, R. Marantz, and T. Imielinski. Handling mobile hosts: A case for indirect interaction. In *Proc. Fourth Workshop on Workstation Operating Systems*, Rosario, Washington, October 1993. IEEE.

[8] Yatin Chawathe, Steve Fink, Steven McCanne, and Eric Brewer. A proxy architecture for reliable multicast in heterogeneous environments. In *Proceedings of ACM Multimedia '98*, Bristol, UK, September 1998.

[9] K.C. Huang and Kwang-Cheng Chen. Interference analysis of nonpersistent CSMA with hidden terminals in multicell data networks. In *Proceedings of IEEE PIMRC, 1995*, September 1995.

[10] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications, high-speed physical layer in the 5ghz band.

[11] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications, higher speed physical layer (phy) extension to ieee 802.11b. June 2003.

[12] Trade-Off Analysis (802.11e versus 802.15.3 QoS mechanism) White Paper. July 2002.

[13] CCITT H.261 Standard. 1993.

[14] CCITT H.263 Standard. 1998.

[15] Joan L. Mitchell, Didier Le Gall, and Chad Fogg. *MPEG Video Compression Standard.* Chapman & Hall, 1996.

[16] Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s (MPEG-1). *ISO/IEC 11172:1993*, 1993.

[17] Information technology – generic coding of moving pictures and associated audio information (MPEG-2). *ISO/IEC 13818:2000*, 2000.

[18] Information technology – coding of audio-visual objects (MPEG-4). *ISO/IEC 14496:2001*, 2001.

[19] Chad Fogg. Questions that should be frequently asked about MPEG. April 1996. Berkeley Multimedia Research Center, available at http://bmrc.berkeley.edu/frame/research/mpeg/mpeg2faq.html.

[20] Hayder Radha, Yingwei Chen, Kavitha Parthasarathy, and Robert Cohen. Scalable Internet video using MPEG-4. *Signal Processing: Image Communication*, 15:95–126, September 1999.

[21] M. van der Schaar, Hayder Radha, and C. Dufour. Scalable MPEG-4 Video Coding with Graceful Packet-Loss Resilience over Bandwidth-varying Networks. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, New York City, New York, July 2000.

[22] M. van der Schaar and J. Meehan. Robust Fine-Granularity-Scalability for Wireless Video. In *Proceedings of IEEE International Packet Video Workshop, 2002*, April 2002.

[23] Distance Vector Multicast Routing Protocol, IETF RFC 1075. 1988.

[24] Host Extensions for IP Multicasting, IETF RFC 1112. 1989.

[25] Internet Group Management Protocol, IETF RFC 2236. 1997.

[26] Protocol Independent Multicast-Sparse Mode (PIM-SM), IETF RFC 2236. 1998.

[27] Steven R McCanne. Scalable compression and transmission of internet multicast video. Technical Report CSD-96-928, 7, 1997.

[28] Philip A. Chou, Alexander E. Mohr, Albert Wang, and Sanjeev Mehrotra. Error Contronl for Receiver-Driven Layered Multicast of Audio and Video. *IEEE Transactions on Multimedia*, March 2001.

[29] Michael Luby, Vivek Goyal, Simon Skaria, and Gavin Horn. Wave and Equation Based Rate Control Using Multicast Round Trip Time. In *Proceedings of IEEE SIGCOMM, 2002*, August 2002.

[30] Sergio D. Servetto, Rohit Puri, Jean-Paul Wagner, Pierre Scholtes, and Martin Vetterli. Video Multicast in (Large) Local Area Networks. In *Proceedings of IEEE InfoCom, 2002*, June 2002.

[31] Mesquite software, product csim. http://www.mesquite.com/products/csim19.htm.

[32] Introduction to directshow. http://www.microsoft.com/Developer/PRODINFO/directx/dxm/help/ds/default.htm.

[33] Multimedia Systems Coursework. Divx: The movie industry's mp3?

[34] Andrew Hawkesworth Department. Divx: Dvd quality movies on a cd-r?

[35] Chiping Tang and Philip K. McKinley. MX: A tool for emulation and simulation of distributed applications and protocols. Technical Report MSU-CSE-01-19, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, June 2001.

[36] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving simulation for network research. Technical Report 99-702b, University of Southern California, March 1999. revised September 1999, to appear in IEEE Computer.

[37] The network simulator - ns-2. http://www.isi.edu/nsnam/ns.

[38] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.

[39] Daji Qiao and Kang G. Shin. A two-step adaptive error recovery scheme for video transmission over wireless networks. In *Proceedings of IEEE Infocom, 2000*, March 2000.

[40] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig. A markov-based channel model algorithm for wireless networks. In *Proceedings of Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2001).*, pages 28–36, July 2003.

[41] Giao Thanh Nguyen, Randy H. Katz, Brian Noble, and M. Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *Winter Simulation Conference*, pages 597–604, 1996.

[42] David A. Eckhardt and Peter Steenkiste. Improving wireless LAN performance via adaptive local error control. *Proceedings of IEEE ICNP '98*, 1998.

[43] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics.* Pearson Education, 2001.

[44] Chiping Tang. Adaptive Reliable Multicast in Wireless Local Area Networks. Master's thesis, Michigan State University, 2002.

[45] N. Golmie, R. E. Van Dyck, and A. Soltanian. Interference of bluetooth and ieee 802.11: simulation modeling and performance evaluation. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 11–18, 2001.

[46] Chiping Tang and Philip K. McKinley. Modeling Multicast Packet Losses in Wireless LANs. Technical Report MSU-CSE-02-12, Department of Computer Science, Michigan State University, East Lansing, Michigan, April 2002.

[47] Peter B. Danzig. Flow Control for Limited Buffer Multicast. *IEEE Transactions on Software Engineering*, 20(1):1–12, January 1994.

[48] Anthony J. McAuley. Reliable broadband communication using a burst erasure correcting code. In *Proc. ACM SIGCOMM '90; (Special Issue Computer Communication Review)*, pages 297–306, September 1990.

[49] Luigi Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM Computer Communication Review*, 27(2):24–36, April 1997.

[50] Philip K. McKinley, Chiping Tang, and Arun P. Mani. A study of adaptive forward error correction for for wireless collaborative computing. *IEEE Transactions on Parallel and Distributed Systems*, September 2002.

[51] Shirish Karande and Hayder Radha. Rate-Constrained Adaptive FEC for Video over Erasure Channels with Memory. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, September 2004.

[52] Luigi Rizzo and Lorenzo Vicisano. A Reliable Multicast data Distribution Protocol based on software FEC techniques. In *The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97)*, Sani Beach, Chalkidiki, Greece, June 1997.

[53] Luigi Rizzo and Lorenzo Vicisano. RMDP: An FEC-based Reliable Multicast Protocol for Wireless Environments. *Mobile Computing and Communications Review*, 2(2), April 1998.

[54] Philip K. McKinley and Arun P. Mani. An experimental study of adaptive forward error correction for wireless collaborative computing. In *Proceedings of the IEEE 2001 Symposium on Applications and the Internet (SAINT-01)*, San Diego-Mission Valley, California, January 2001.

[55] Hang Liu and Magda El Zarki. Adaptive Source Rate Control for Real-Time Wireless Video Transmission. *Mobile Networks and Applications*, 3(1):49–60, 1998.

[56] Dongyan Xu, Baochun Li, and Klara Nahrstedt. Qos-directed error control of video multicast in wireless networks. In *Proceedings of IEEE International Conference on Computer Communications and Networks*, October 1999.

[57] Peng Ge and Philip K. McKinley. Experimental evaluation of error control for video multicast over wireless LANs. In *Proceedings of the Third International Workshop on Multimedia Network Systems*, Phoenix, Arizona, April 2001.

[58] Nick Feamster and Hari Balakrishnan. Packet Loss Recovery for Streaming Video. In *Proceedings of IEEE International Packet Video Workshop, 2002*, April 2002.

[59] Abhik Majumdar, Daniel Grobe Sachs, Igor V. Kozintsev, Kannan Ramchandran, and Minerva M. Yeung. Multicast and Unicast Real-Time Video Streaming over Wireless LAN. *IEEE Transactions on Circuits and Systems for Video Technology*, June 2002.

[60] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, December 1997.

[61] Joy Kuri and Sneha Kasera. Reliable multicast in multi-access wireless LANs. In *Proceedings of IEEE Infocom, 1999*, March 1999.

[62] Ken Tang and Mario Gerla. MAC Layer Broadcast Support in 802.11 Wireless Networks. In *Proceedings of IEEE MILCOM, 2000*, October 2000.

[63] Ken Tang and Mario Gerla. Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks. In *Proceedings of IEEE WCNC, 2000*, September 2000.

[64] Ken Tang and Mario Gerla. MAC Reliable Broadcast in Ad Hoc Networks. In *Proceedings of IEEE MILCOM, 2001*, October 2001.

[65] Min-Te Sun, Lifei Huang, Anish Arora, and Ten-Hwang Lai. Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks. *Wiley Wireless Communications and Mobile Computing on Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing*, 2003.

[66] IEEE standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications . *ISO/IEC 8802-3:2000(E)*, 2000.