Multicast MAC Extensions for High Rate Real-Time Traffic in Wireless LANs

Dissertation

Zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technische Fakultät I
- Mathematik und Informatik -
der Universität des Saarlandes

von

Zhao Li

Saarbrücken, Deutschland 2011

Dekan:	Prof. Dr. Holger Hermanns, Universität des Saarlandes
Vorsitzender:	Prof. Dr. Markus Bläser, Universität des Saarlandes
Gutachter:	Prof. Dr. Thorsten Herfet, Universität des Saarlandes
	Prof. Dr. Rüdiger Kays, Technische Universität Dortmund
Beisitzer:	Dr. Jörg Baus, Universität des Saarlandes
Kolloquiums:	22. 09. 2011

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, daß ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen order indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Diese Arbeit wurde bisher weder im In- noch im Ausland in gleicher order ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgeleget.

Saarbrücken, __.__.____

____

(Zhao Li)

Acknowledgments

My research project and thesis would not have been accomplished without the help and support from my colleagues, friends and family.

First and foremost I would like to thank my supervisor, Prof. Dr.-Ing. Thorsten Herfet, chair of the Telecommunications Lab at the Saarland University. His wide knowledge, suggestions and guidance have been highly valuable for my work. Greatest gratitude to his continued support and encouragement.

I am also grateful to Prof. Dr.-Ing. Rüdiger Kays, who has kindly accepted the task of reviewing my thesis.

My special thanks then go to my Master supervisor, Prof. Baohua Zhao, Dept. Computer Science, the University of Science and Technology of China, for his support and valuable advices to my research work.

Many thanks to Jochen Miroll for a fruitful cooperation during the research work. He has been of an enormous help in contributing with ideas and working on the tests and measurements presented in the thesis. I gratefully acknowledge the friendly help from Manuel Gorius. And I wish to thank Zakaria Keshta and Diane Chlupka for their kind assistance. I also appreciate all of the other colleagues for their help and companionship.

Furthermore, I would like to thank the Doctoral Training Program between the Chinese Academy of Science and the Max Planck Society for the support.

Finally, I would like to express my loving thanks to my wife Yuanyuan Shi for her endless support and love. My special gratitude also goes to my parents and my sister for their loving support.

Kurze Zusammenfassung

Zurzeit vollzieht sich ein schneller Wechsel vom vorwiegend textbasierten zum multimediabasierten Internet. Die weitverbreiteten IEEE 802.11 Drahtlosnetzwerke sind vielversprechende Kandidaten, um das Internet für Nutzer überall, jederzeit und auf jedem Gerät verfügbar zu machen. Die Unterstützung gruppenorientierter Echtzeit-Dienste in drahtlosen lokalen Netzen ist jedoch immer noch eine Herausforderung. Das liegt daran, dass aktuelle Protokolle keinen Multicast unterstützen. Sie senden Multicast-Pakete vielmehr in einer „Open Loop"-Strategie als Broadcast-Pakete, d. h. ohne jegliche Rückmeldung (feedback) oder Paketwiederholungen. In der vorliegenden Arbeit, anders als in den auf Teilnehmereinzelabfragen (polling) basierenden Ansätzen, die unter langen Verzögerungen, geringer Skalierbarkeit und geringer Effizienz leiden, versuchen wir, Multicast-Feedback bestehend aus positiven (ACK) und negativen Bestätigungen (NACK) auf MAC-Layer im selben Zeitfenster zu bündeln. Die übrigen Empfänger können NACK-Frames senden, um das ACK des Leaders zu zerstören und Paketwiederholungen zu veranlassen. Basierend auf einem Feedback-Jamming Schema schlagen wir zwei MAC-Layer-Protokolle für den Fehlerschutz im Multicast vor: Das SEQ-getriebene Leader Based Protocol (SEQ-LBP) und das Hybrid Leader Based Protocol (HLBP). SEQ-LBP ist eines Automatic Repeat reQuest (ARQ) Schema. HLBP kombiniert ARQ und paketbasierte Forward Error Correction (FEC). Wir evaluieren die Leistungsfähigkeit von ACK/NACK jamming, SEQ-LBP und HLBP durch Analysis, Simulationen in NS-2, sowie Experimenten in einer realen Testumgebung mit handelsüblichen WLAN-Karten. Die Testergebnisse bestätigen die Anwendbarkeit der Feedback-Jamming Schemata und die herausragende Leistungsfähigkeit der vorgestellten Protokolle SEQ-LBP und HLBP. SEQ-LBP ist durch seine kurze Verzögerung, seine Effektivität und seine Einfachheit für kleine Multicast-Gruppen nützlich, während HLBP auf Grund seiner hohen Effizienz und Skalierbarkeit im Bezug auf die Größe der Empfänger eher in großen Multicast-Gruppen anzuwenden ist.

Short Abstract

Nowadays we are rapidly moving from a mainly textual-based to a multimedia-based Internet, for which the widely deployed IEEE 802.11 wireless LANs can be one of the promising candidates to make them available to users anywhere, anytime, on any device. However, it is still a challenge to support group-oriented real-time multimedia services, such as video-on-demand, video conferencing, distance educations, mobile entertainment services, interactive games, etc., in wireless LANs, as the current protocols do not support multicast, in particular they just send multicast packets in open-loop as broadcast packets, i.e., without any possible acknowledgements or retransmissions. In this thesis, we focus on MAC layer reliable multicast approaches which outperform upper layer ones with both shorter delays and higher efficiencies. Different from polling based approaches, which suffer from long delays, low scalabilities and low efficiencies, we explore a feedback jamming mechanism where negative acknowledgement (NACK) frames are allowed from the non-leader receivers to destroy the acknowledgement (ACK) frame from the single leader receiver and prompts retransmissions from the sender. Based on the feedback jamming scheme, we propose two MAC layer multicast error correction protocols, SEQ driven Leader Based Protocol (SEQ-LBP) and Hybrid Leader Based Protocol (HLBP), the former is an Automatic Repeat reQuest (ARQ) scheme while the later combines both ARQ and the packet level Forward Error Correction (FEC). We evaluate the feedback jamming probabilities and the performances of SEQ-LBP and HLBP based on theoretical analyses, NS-2 simulations and experiments on a real test-bed built with consumer wireless LAN cards. Test results confirm the feasibility of the feedback jamming scheme and the outstanding performances of the proposed protocols SEQ-LBP and HLBP, in particular SEQ-LBP is good for small multicast groups due to its short delay, effectiveness and simplicity while HLBP is better for large multicast groups because of its high efficiency and high scalability with respect to the number of receivers per group.

Zusammenfassung

Zurzeit vollzieht sich ein schneller Wechsel vom vorwiegend textbasierten zum multimediabasierten Internet. Die weitverbreiteten IEEE 802.11 Drahtlosnetzwerke sind vielversprechende Kandidaten, um das Internet für Nutzer überall, jederzeit und auf jedem Gerät verfügbar zu machen. Die Unterstützung gruppenorientierter Echtzeit-Dienste, wie Video-on-Demand, Video-Konferenzen, Fernunterricht, mobile Unterhaltungsdienste, interaktive Spiele, etc. in drahtlosen lokalen Netzen ist jedoch immer noch eine Herausforderung. Das liegt daran, dass aktuelle Protokolle keinen Multicast unterstützen. Sie senden Multicast-Pakete vielmehr in einer „Open Loop"-Strategie als Broadcast-Pakete, d. h. ohne jegliche Rückmeldung (feedback) oder Paketwiederholungen. In der vorliegenden Arbeit betrachten wir zuverlässige Multicast-Ansätze auf MAC-Layer, die Mechanismen auf höheren Ebenen durch kürzere Verzögerungen und höhere Effizienz übertreffen.

Anders als in den auf Teilnehmereinzelabfragen (polling) basierenden Ansätzen, die unter langen Verzögerungen, geringer Skalierbarkeit und geringer Effizienz leiden, versuchen wir, Multicast-Feedback bestehend aus positiven (ACK) und negativen Bestätigungen (NACK) auf MAC-Layer im selben Zeitfenster zu bündeln. Ein Empfänger wird hierbei als Leader ausgewählt. Die übrigen Empfänger können NACK-Frames senden, um das ACK des Leaders zu zerstören und Paketwiederholungen zu veranlassen. Auf Grund des sogenannten Capture-Effekts kann das ACK-Frame die Kollision mit den NACKs überstehen, insbesondere wenn die Empfangsstärke der NACKs geringer ist. Wir evaluieren die ACK/NACK Jamming-Wahrscheinlichkeit durch theoretische Analysis, NS-2-Simulationen, sowie Messungen in realen Testumgebungen mit handelsüblichen WLAN-Karten. Durch Experimente der Gesamtlänge von einhundert Stunden unter verschiedenen Szenarien haben wir herausgefunden, dass die Hardware-ACK/NACK Jamming-Wahrscheinlichkeit bis zu 0.99+ für übliche Szenarien (etwa 0.90+ für Worst Cases mit nur zwei Empfängern

erreichen kann, wenn ein einfacher, dynamischer Leader-Auswahlalgorithmus (Leader selection) benutzt wird. Das Ergebnis bestätigt, dass Hardware-ACK/NACK Jamming als effektives und effizientes Multicast-Feedback im Design von zuverlässigem MAC-Layer Multicast angewendet werden kann.

Basierend auf einem Feedback-Jamming Schema schlagen wir zwei MAC-Layer-Protokolle für den Fehlerschutz im Multicast vor: Das SEQ-getriebene Leader Based Protocol (SEQ-LBP) und das Hybrid Leader Based Protocol (HLBP). SEQ-LBP erweitert das existierende Leader Based Protocol (LBP) in Form eines Automatic Repeat reQuest (ARQ) Schemas mit höherer Effizienz und der Einführung eines SEQ-Frames. HLBP wirkt der begrenzten Skalierbarkeit reiner ARQ-Schemata durch die Einführung von paketbasierter Forward Error Correction (FEC) entgegen. Wir evaluieren die Leistungsfähigkeit von LBP, SEQ-LBP und HLBP durch Analysis, basierend auf dem Gilbert-Elliot (GE) Kanal-Modell, sowie Simulationen in NS-2. Die Simulationsergebnisse verifizieren die theoretischen Betrachtungen und demonstrieren die Vorteile der vorgestellten Protokolle. Folglich ist SEQ-LBP durch seine kurze Verzögerung, seine Effektivität und seine Einfachheit für kleine Multicast-Gruppen nützlich, während HLBP auf Grund seiner hohen Effizienz und Skalierbarkeit eher in großen Multicast-Gruppen anzuwenden ist.

Des Weiteren implementieren wir ein SEQ-LBP mit dynamischer Leader-Selection und Multicast-Management auf Treiber-Ebene in einer realen Testumgebung und evaluieren seine Leistungsfähigkeit im Wiederherstellen von Paketverlusten im Multicast. Wir evaluieren auch ein rein auf NACK-Jamming basierendes ARQ-Schema in unserer Testumgebung und untersuchen die Falsch-Positiv-Rate reiner NACK-Aggregation. Außerdem diskutieren wir hochauflösende Codierung auf MAC-Layer, Mechanismen zur Ratenadaption und protokollschichtübergreifende Mechanismen, die das vorgestellte Feedback-Jamming-Protokoll erweitern können.

Abstract

Nowadays we are rapidly moving from a mainly textual-based to a multimedia-based Internet, for which the widely deployed IEEE 802.11 wireless LANs can be one of the promising candidates to make them available to users anywhere, anytime, on any device. However, it is still a challenge to support group-oriented real-time multimedia services, such as video-on-demand, video conferencing, distance educations, mobile entertainment services, interactive games, etc., in wireless LANs, as the current protocols do not support multicast, in particular they just send multicast packets in open-loop as broadcast packets, i.e., without any possible acknowledgements or retransmissions. In this thesis, we focus on MAC layer reliable multicast approaches which outperform upper layer ones with both shorter delays and higher efficiencies.

Different from polling based approaches, which suffer from long delays, low scalabilities and low efficiencies, we try to aggregate MAC layer multicast feedback, acknowledgement (ACK) and negative ACK (NACK), in the same time slot. In particular, one receiver is selected as the leader, NACK frames are allowed from the non-leader receivers to destroy the ACK frame from the leader receiver and prompt retransmissions from the sender. Due to the capture effect, the ACK frame may survive the collision with NACK frames especially when the receiving powers of NACK frames are lower. We evaluate the ACK/NACK jamming probabilities through theoretical analyses, NS-2 simulations, as well as measurements on a real test-bed built with consumer wireless LAN cards. By hundred hours of tests (each one lasts for several hours) under various scenarios, we find that the hardware ACK/NACK jamming probability can be as high as 0.99+ for normal scenarios (about 0.90+ for the worst case with only two receivers which even experience nearly the same channel condition) when a simple dynamic leader selection algorithm is used. As a result, it is confirmed that, for the first time, the hardware ACK/NACK jamming can be applied as an effective and efficient multicast feedback in the design of MAC layer reliable Multicast.

Based on the feedback jamming scheme, we propose two MAC layer multicast error correction protocols, SEQ driven Leader Based Protocol (SEQ-LBP) and Hybrid Leader Based Protocol (HLBP). As an Automatic Repeat reQuest (ARQ) scheme, SEQ-LBP enhances an existing protocol, Leader Based Protocol (LBP), with a higher efficiency by the adoption of a SEQ frame. HLBP overcomes the scalability limitation of pure ARQ schemes by the introduction of a packet level Forward Error Correction (FEC). We evaluate the performances of LBP, SEQ-LBP and HLBP through analyses over the Gilbert-Elliott (GE) channel model as well as simulations in NS-2. The simulation results verify the theoretical analyses and show the advantages of the proposed protocols. Due to the SEQ frame, SEQ-LBP avoids the problems of LBP and is more efficient in various scenarios. Due to the block coding and block feedback, HLBP is much more efficient than both LBP and SEQ-LBP and has a superior scalability with respect to the number of receivers per multicast group. In conclusion, SEQ-LBP is good for small multicast groups due to its short delay, effectiveness and simplicity while HLBP is better for large multicast groups because of its high efficiency and high scalability.

We implement a driver level SEQ-LBP with dynamic leader selection and multicast management on a real test-bed and evaluate its performance for recovering multicast packet losses. This driver level SEQ-LBP can replace the normal MAC broadcast in the *Madwifi* driver and provide a reliable MAC layer multicast service for real applications. We also evaluate a pure NACK jamming based ARQ scheme on the test-bed and investigate the fake detection problems of pure NACK aggregation. Moreover, we discuss the fine granularity MAC layer coding, data rate adaptation mechanisms and cross-layer issues which could enhance the proposed feedback jamming based protocols.

Contents

List of Figures

List of Tables

Abbreviations

3GPP	Third-generation Partnership Project
3GPP2	Third-generation Partnership Project 2
ACK	Acknowledgment
ACKs	Acknowledgments
AL	Application Layer
AODV	Ad Hoc on Demand Distance Vector
AP	Access Point
ARF	Automatic Rate Fallback
ARQ	Automatic Repeat request
AV	Audio-Video
BCMCS	BroadCast MultiCast Services
BER	Bit Error Rate
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access

CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DCF	Distributed Coordination Function
DIFS	Distributed Inter-Frame Space
DLS	Dynamic Leader Selection
DSR	Dynamic Source Routing
DVB	Digital Video Broadcast
DVMRP	Distance Vector Multicast Routing Protocol
E-MBMS	Evolved MBMS in 3GPP LTE
ESS	Extended Service Set
FCS	Frame Check Sequence
FEC	Forward Error Correction
FMI	Future Media Internet
GCR	GroupCast with Retries
GE	Gilbert-Elliott
HAL	Hardware Abstraction Layer
HARQ	Hybrid Automatic Repeat request
HARQ I	Hybrid Automatic Repeat reQuest Type I
HARQ II	Hybrid Automatic Repeat reQuest Type II

HARQ III	Hybrid Automatic Repeat reQuest Type III
HDTV	High Definition Television
HEC	Hybrid Error Correction
HLBP	Hybrid Leader Based Protocol
HSDPA	High Speed Downlink Packet Access
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
i.i.d	Independent and Identically Distributed
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPTV	IP Television
IR	Incremental Redundancy
ITU	International Telecommunication Union
ITU-T	ITU's Telecommunication Standardization Sector
LAN	Local Area Network
LBP	Leader Based Protocol

LDPC	Low Density Parity Check
LTE	3GPP Long Term Evolution
MAC	Medium Access Control
MAODV	Multicast Ad-hoc On-demand Distance Vector
MBMS	Multimedia Broadcast/Multicast Services
MIMO	Multiple Input Multiple Output
MLD	Multicast Listener Discovery
MPDU	MAC Protocol Data Unit
NACK	Negative Acknowledgment
NAV	Network Allocation Vector
NS-2	Network Simulator version 2
OAR	Opportunistic Auto Rate
OFDM	Orthogonal Frequency Division Multiplex
OFDMA	Orthogonal Frequency Division Multiplex Access
OMACK	OFDMA based Multicast ACK
PBC	Periodic Broadcas
PCF	Point Coordination Function
PER	Packet Error Rate
PGM	Pragmatic General Multicast

PHY	Physical
PIM	Protocol Independent Multicast
PLCP	Physical Layer Convergence Protocol
PLR	Packet Loss Ratio
QoS	Quality of Service
RA	Receiver Address
RBAR	Receiver Based Auto Rate
RI	Redundancy Information
RS	Reed-Solomon
RSSI	Received Signal Strength Indication
RTP	Real-time Transport Protocol
RTCP	Real-time Transport Control Protocol
RTS	Request to Send
RTT	Round Trip Time
SEQ-LBP	SEQ driven Leader Based Protocol
SGE	Simplified Gilbert-Elliott
SIFS	Short Inter-Frame Space
SINR	Signal to Interference-plus-Noise Ratio
SNR	Signal to Noise Ratio

SW	Stop-and-Wait
TA	Transmitter Address
TCL	Tool Command Language
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
Tgaa	IEEE 802.11aa Task Group
UCF	Unary Channel Feedbacks
UDP	User Datagram Protocol
UDP-Lite	User Datagram Protocol Lite
UNF	Unary Negative Feedback
VTS	Video Transmission Study
WHN	Wireless Home Networks
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access

Notations

α	Transmitting probability from Bad state to Bad state in the GE channel model
β	Transmitting probability from Good state to Good state in the GE channel model
ρ	Temporal error correlation in the GE channel model
$BlockLR(k,x)$	The block loss probability after x retry rounds in HLBP with a block code (n,k)
d	Distance for a station to the AP
$DataLR(k,x)$	The data packet loss rate after x retry rounds with a data block size k in HLBP with a block code (n,k)
D_{LBP}	The MAC layer maximum delay in LBP
$D_{SEQ-LBP}$	The MAC layer maximum delay in SEQ-LBP
D_{HLBP}	The MAC layer maximum delay in HLBP
ET_{LBP}	The expected channel holding time per data packet in LBP
$ET_{SEQ-LBP}$	The expected channel holding time per data packet in SEQ-LBP
ET_{HLBP}	The expected channel holding time per data packet in HLBP
$E(x)$	Expected value of variable x
$Expt_{LBP}$	The expected number of transmissions per data packet in LBP

$Expt_{SEQ-LBP}$	The expected number of transmissions per data packet in SEQ-LBP
$Expt_{HLBP}$	The expected number of transmissions per data packet in HLBP
$FECPLR(k,x,y)$	The probability that y packets are still missing to decode the current block after $k+x$ packets have been transmitted in HLBP with a block code (n,k)
JP	Feedback jamming probability of an ACK frame against a single NACK frame
$JP(Th_{cap},R)$	Feedback jamming probability given receiver number R and Packet capture threshold Th_{cap}
L	Packet length
m	Retry limit
(n,k)	FEC block code
N_t	The thermal noise
p	Packet loss probability; p also denotes transmitting/receiving power if stated specially
$P[a,b]$	The probability of a errors in a sequence of b symbols over the GE channel model
$P_B[a,b]$	The probability of a errors in b transmissions with the channel ending in state B over the GE channel model
$P_G[a,b]$	The probability of a errors in b transmissions with the channel ending in state G over the GE channel model
P_{ber}	Bit Error Rate
P_B	Error probability in Bad state of the GE channel model
p_e	Symbol error rate or called erasure probability
$P_e(l)$	The error probability of a l bytes data

P_G	Error probability in Good state of the GE channel model
PLR^{l}_{LBP}	The final packet loss ratio at the leader receiver in LBP
PLR^{nl}_{LBP}	The final packet loss ratio at the non-leader receivers in LBP
$PLR^{l}_{SEQ-LBP}$	The final packet loss ratio at the leader receiver in SEQ-LBP
$PLR^{nl}_{SEQ-LBP}$	The final packet loss ratio at the non-leader receivers in SEQ-LBP
PLR^{l}_{HLBP}	The final packet loss ratio at the leader receiver in HLBP
PLR^{nl}_{HLBP}	The final packet loss ratio at the non-leader receivers in HLBP
$Pr(X)$	Occurring probability of even X
p_{res}	Residual output error rate
RI_{LBP}	The average redundant information (transmission) per data packet in LBP
$RI_{SEQ-LBP}$	The average redundant information (transmission) per data packet in SEQ-LBP
RI_{HLBP}	The average redundant information (transmission) per data packet in HLBP
T_{ACK}	The transmission time of an ACK frame
T_{ARQ}	The multicast delay in ARQ schemes
T_{cc}	The channel contention time
T_{CTS}	The transmission time of a CTS frame
$T_{DATA}(L)$	The transmission time of a data frame (L bytes)
T_{DIFS}	The Distributed Inter Frame Space time
T_{FEC}	The multicast delay in FEC schemes

Th_{cap}	Packet capture threshold
T_{HEC}	The multicast delay in HEC schemes
$T_{LBP}(L)$	The channel holding time of one transmission with data packet length L in LBP
T_{PLCP}	The transmission time of the PLCP preamble and header
T_{RTS}	The transmission time of a RTS frame
T_{RTT}	The bound of the Round Trip Time
T_{SEQ}	The transmission time of the SEQ frame
$T_{SEQ-LBP}(L)$	The channel holding time of one transmission with data packet length L in SEQ-LBP
T_{SIFS}	The Short Inter Frame Space time
T_d	Packet transmission time, packet-length/capacity
T_s	Packet interval for multicast loads

Chapter 1

Introduction

Nowadays we are rapidly moving from a mainly textual-based to a multimedia-based Internet, where rich audio/video content, 3D representations, virtual and mirror worlds, serious games, lifelogging applications, etc., become a reality. The exponential increase of multimedia applications involving the transmission of audio/video content over communication networks has led to the birth of the Future Media Internet (FMI) [FMI10] concept. High Definition TV (HDTV[1]) is revolutionizing the world and Digital Video Broadcasting (DVB[2]) is running over all kinds of traditional transmission media. It has been also claimed that Internet Protocol (IP) Television (IPTV) is the killer application for the next-generation Internet [Xia07]. Most of these real-time multimedia applications typically tolerate a residual Packet Loss Ratio (PLR) but are bound by strict delay constraints. Furthermore, with the increase in the demands for group-oriented real-time multimedia services such as video-on-demand, video conferencing, distance educations, mobile entertainment services, interactive games, etc., the support of multimedia multicast is required in the next-generation IP-based wireless networks. Multicast is an efficient paradigm for transmitting data from a sender to a group of receivers as it limits the transmissions of redundant packets and saves bandwidth as well as energy, which becomes

[1] http://www.timefordvd.com/tutorial/DigitalTVTutorial.shtml

[2] Digital Video Broadcasting (http://www.dvb.org)

of a greater value when wireless medium is concerned due to the fact that wireless medium is of broadcast in its nature. The multimedia multicast has been already supported in current mobile networks, such as Multimedia Broadcast/Multicast Services (MBMS) [3gp05a] [3gp05b] in the Third Generation Partnership Project (3GPP), BroadCast MultiCast Services (BCMCS) [3gp06] in the Third Generation Partnership Project 2 (3GPP2) and Evolved MBMS (E-MBMS) in 3GPP Long Term Evolution (LTE), which provide the capability to distribute real-time multimedia services for mobile users via IP multicast data over point-to-multipoint radio bearers.

Widely deployed IEEE 802.11 wireless LANs can be one of the promising candidates for delivering multicast multimedia. Furthermore, the adoption of the IEEE 802.11n specification [11n09], supporting a maximum PHY (physical layer) data-rate of 600Mbps (raw data-rate) using advanced PHY techniques such as MIMO and channel bonding, may further boost the acceptance of wireless LANs for wireless multimedia applications in environments such as home, office, hot-spots, airports, universities, etc., in which it is a logical extension to enable digital TV distribution using wireless LAN. The characteristics of wireless networks can be summarized as a bandwidth variation and terminal heterogeneity plus a high degree of packet losses. The Quality of Services (QoS) requirements for multimedia services in the other hand include a certain amount of reliability, short delays and low delay jitters, in some combinations, varying from application to application. The success of multimedia multicast in wireless networks depends on the support for these requirements, as well as the support for reliable quality, real-time, device mobility, AV (audio/video) content protection, content adaptation, and scalability [Sin08]. However, the current IEEE 802.11 standards do not comply with many of these requirements. In particular, the current MAC layer forwards multicast packets in an open-loop manner as broadcast packets, i.e., without any feedback or timely delivery mechanisms [IEEE07]. How to support real-time multimedia multicast in wireless LANs in various environments is still a challenge.

The goal of this thesis is to design and evaluate multicast error control schemes and protocols in the Medium Access Control (MAC) layer to support real-time multimedia multicast in wireless LANs. The main problem is to minimize the cost of multicast error correction while guaranteeing the quality of multicast delivery under both PLR and delay constraints.

1.1. State of the Art

Multicast has the capability to support point-to-multipoint communications between a single sender and multiple receivers. This capability can be employed at different layers such as application layer, IP layer and MAC layer in a network protocol stack. Numerous multicasting routing algorithms are proposed at the IP layer, such as Distance Vector Multicast Routing Protocol (DVMRP) [Wai88] and Protocol Independent Multicast (PIM) [Wei99]. Also, to support multicast routing protocols, Internet routers use a group membership protocol - Internet Group Management Protocol (IGMP) [Cai02] for IPv4 and Multicast Listener Discovery (MLD) [Vid04] for IPv6. However, multicasting at the IP layer requires all intermediate Internet routers to upgrade. Moreover, the most common transport layer protocol to use multicast addressing is UDP, which is not reliable by its nature - messages may be lost or delivered out of order. Reliable multicast protocols such as Pragmatic General Multicast (PGM) [Spe01] have been developed to add loss detection and retransmission on top of IP multicast for both IPv4 and IPv6. Therefore, more realistic proposals focus on multicasting overlay at the application layer, called end-to-end multicast. On the other hand, due to the broadcasting nature of link layer communications, there have been a little research works on multicasting at the MAC layer in the literature. Wireless LANs, the last mile extension of the Internet, bring forth new challenges to support reliable multicast for multimedia applications.

For wireless LANs, the current IEEE 802.11 distributed coordinate function (DCF), which is the basic media access protocol used for unicast communication, is reliable because of its carrier sensing scheme and retransmission scheme. This protocol uses Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) to facilitate medium sharing between contending transmitters. (Figure 1.1 illustrates the 802.11 DCF mechanism.) The transmitter, after sensing the medium to be idle (for DIFS), sends a Request To Send (RTS) frame with the transmitter and receiver addresses and the duration for which the medium is to be reserved. Any node other than the receiver, who hears the RTS, sets its network allocation vector (NAV) up to the time period mentioned in the RTS. When the intended receiver receives the RTS frame, and senses the medium to be free, it replies with a Clear To Send (CTS) frame after waiting for a small inter frame space (SIFS) period. Any node other than the transmitter, who hears the CTS and had not heard the RTS before, would set its NAV to the time period mentioned in the CTS. The successful reception of CTS by the transmitter indicates that the medium has been reserved. The transmitter waits for a SIFS duration and transmits the data frame. On receiving DATA successfully, the receiver waits for a SIFS duration and replies with an acknowledgement (ACK). When the transmitter receives the ACK it knows that the DATA was successfully delivered. The losses of CTS or ACK lead the transmitter to repeat the whole procedure until an ACK is received or the retry limit is reached. However, the IEEE 802.11 standards do not comply with multicast data requirements. In particular, the current MAC layer forwards multicast packets in open-loop as broadcast packets (shown in Figure 1.1), i.e., without any collision avoidances or acknowledgements. It is not possible to extend the DCF carrier sensing scheme and feedback scheme directly to multicast scenarios because the simultaneous ACKs from multiple receivers will result in a collision at the transmitter. So, efficient multicast protocols are required for wireless LANs to support multimedia multicast applications.

Currently, the reliability of multicast in wireless LANs is achieved through upper layers (e.g. the application layer) in an end-to-end basis. Typically, there are three types of error correction techniques: Automatic Repeat reQuest (ARQ) [Non98a] [Tan07a], Forward Error Correction (FEC) [Non98b] [Riz97a] erasure coding and Hybrid Error Correction (HEC) [Qia00] [Car97] [Ada04] [Tan07b] [Tan09] which is the integration of ARQ and FEC. Although the application layer error correction schemes can provide reliability to some multicast applications, they are not a perfect option when applications such as multimedia conferencing or streaming video/audio are concerned due to the excessive delay they might cause, e.g. upper layer protocol waiting, MAC layer queuing, etc. On the other hand, link layer error recovery operates on a considerably smaller time-scale. Investing the link layer error recovery is worthwhile because it improves the quality of multimedia applications as seen by the end-user. For end-to-end reliable multicast applications like multicast file transfer, distributed computing, chat and whiteboard applications, link layer reliability saves time as well as network and end system resources. Link layer approaches can operate on individual frames, using implicit transparent link fragmentation. Frames may be much smaller than IP packets, and repetition of smaller frames containing lost or erroneous parts of an IP packet may improve the efficiency of the error recovery process and the efficiency of the link. A link multicast error recovery procedure may be able to use local knowledge that is not available to end hosts, to optimize delivery performance for the current link conditions. This information can include the state of the link and channel, e.g., knowledge of the current available transmission rate, the prevailing error environment, or available transmitting power in wireless links.

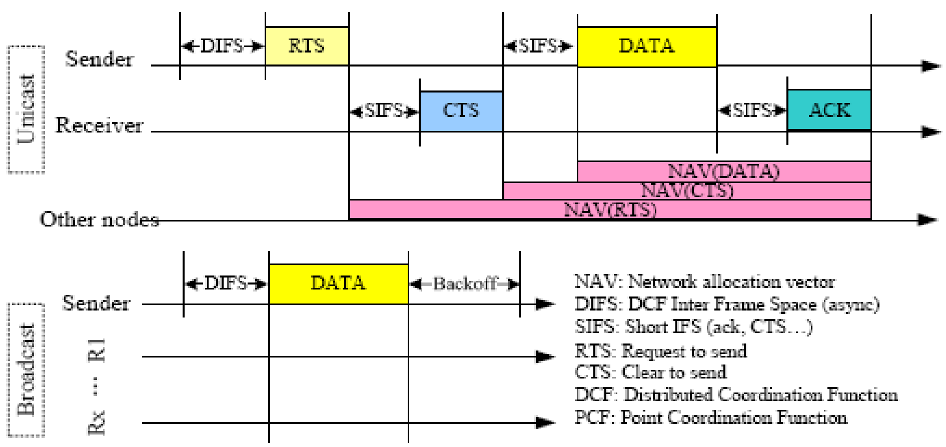

Figure 1.1: IEEE 802.11 DCF Unicast and Broadcast

Moreover, for multi-hop multicast in wireless networks or hybrid networks with both wired networks and wireless LANs which are common scenarios for wireless multimedia applications, the need for additional transmissions due to the errors in wireless LANs puts unnecessary processing burden on the original remote sender and the entire network. If the access points (AP) (or base stations) were to take the responsibility of supplying retransmissions rather than the original sender, then the load of supplying retransmission gets distributed across access points and takes a shorter time [Kur99]. Furthermore, even for scenarios where applications themselves do not demand multicast, several higher layer protocols rely heavily on reliable link layer multicast, for instance Dynamic Source Routing (DSR) and Ad Hoc on Demand Distance Vector (AODV) routing protocols [Sun02].

There are several major challenges in the design of MAC layer multicast for wireless LANs, shown as follows:

- **Feedback Implosion and Scalability**

  In order to ensure reliable transmissions from the sender to receivers, it is important for the receivers to send feedback either by acknowledging every packet that is received by sending an ACK message or by sending a negative acknowledgement (NACK) message for a packet that is detected as lost. In case of a large number of receivers, an ACK for every packet from every receiver (scheduled in different time slots) would lead to a so–called “ACK implosion” at the sender, which would occupy most of the channel time (the base transmission rate is used for ACK/NACK). Another problem with an ACK–based approach is that the sender has to maintain a list, for each packet, of the receivers from which it received an ACK. This could be an important overhead for the MAC layer, especially when the number of receivers is very large. Due to the possibility of ACK implosion and the requirement of maintaining the identity of each receiver, an ACK–based approach for reliable multicast is generally not considered to be scalable [Tow97]. A NACK–based approach is more practical because the number of NACKs is likely to be much less than the number of ACKs. Also, with a NACK based approach the sender does not need to be aware of the number and identity of receivers; the responsibility of loss recovery is now on the receiver – a receiver wishing a retransmission simply sends a NACK to the sender. However, when the network is large and the loss probabilities are high we can have a NACK implosion too.

Random timers can be used for the purpose of NACK suppression [Flo97] [Tow97]. A receiver, on detecting a loss, waits for a random amount of time and multicasts a NACK to all the group members. While waiting, if a receiver receives a NACK for the same packet, it suppresses its own NACK. Moreover, the sender can also wait a little bit to aggregate the NACKs instead of retransmitting upon receiving a NACK. However, currently those feedback suppression approaches are only applicable to upper layer or end-to-end multicast. How to schedule and suppress multicast feedback in the MAC layer is still a challenge.

- **Loss Detection**

The loss detections in the upper layers (e.g. the application layer) usually are based on timers or sequence check. In the timer based mechanisms, a receiver regards it as a loss when the expecting packet has not arrived until the timer expires. In the sequence check based mechanisms, a receiver regards it as a loss when it receives a packet with a higher sequence number than the expecting one. In the MAC layer, a packet cannot be trusted and is thrown away when it fails the Cyclic Redundancy Check (CRC). As it is in a very small and fine time granularity and the channel is shared with other nodes, a single timer or sequence check cannot be used for loss detection at receivers side in the MAC layer. Normally, we have to poll each receiver to detect a loss which is a heavy time consuming work. However, if we just poll once and each receiver replies feedback in its allocated time lost, the problem could be relieved a little, but it requires high-precision time synchronization which is another hard work.

- **Time Precision**

As the channel contention (CSMA/CA) is a time consuming work with respect to data transmission time, the feedback had better be sent out following the data frame as a control frame instead of being handled as a data packet which has to compete the channel again. This requires high time precision. In 802.11 DCF, an ACK frame can be sent after a SIFS since the reception of the data packet. However, a NACK frame cannot be sent like this. This is because the packet cannot be trusted when it fails the CRC check, even though there is maybe just a bit error, and hence the destination for the NACK is unknown. Moreover, if the frame is severely damaged, it is impossible to find the right time to reply feedback. Normally, we have to poll each receiver to request feedback.

A task group in IEEE 802.11 identified as task group 802.11aa (Tgaa) [Tgaa10] has been formed after a project approval request made by the video transmission study group (VTS) [Har08] in 2008. Its goal is to standardize MAC layer enhancements for more reliable multicast transmission of real-time multimedia streams over wireless LANs. Several mechanisms have been discussed in Tgaa, namely those for multicast error correction via retransmission (ARQ), which are summarized in 802.11aa under the term “GroupCast with Retries” (GCR). In GCR, a group is a Multicast group receiving one (e.g. high throughput video) stream. Group membership detection may be achieved via IGMP snooping as is typically done for multicast in 802.11. Hence, a set of wireless stations receiving the same stream in 802.11aa GCR are subject to an error correction that applies to the whole group of stations or a subset thereof (with implies lack of feedback from the rest of the group). Currently, Tgaa specifies in its draft [Tgaa10] a Groupcast Block-ACK polling mechanism. In such a scheme, each addressed receiver is individually requested to provide a bit map of previously correctly received data frames. Upon this information the sender can consolidate frame retransmissions.

As a polling based scheme, when every station is addressed, the GCR block-ACK polling scales linearly with the number of receivers and consumes more time, although in this case it provides near perfect reliability on the MAC layer, similar to unicast traffic [Tgaa10]. Hence, the GCR block-ACK polling may not satisfy the delay constraints of real-time applications where the multicast group size could be dozens or even hundreds, e.g. in the last mile scenarios such as video conference, hot-spot, gaming etc. In this thesis, we focus on ACK/NACK jamming based approaches, in which, one receiver is selected as the leader, NACK frames from non-leader receivers are used to cancel the ACK frame, if any, sent by the leader. Destruction of the leader's ACK by NACKs would then trigger a retransmission. If based on ACK/NACK jamming, such protocols could achieve low latency and high throughput at a predictable reliability, as well as a superior scalability with respect to the number of receivers. Contrast to the GCR block-ACK polling, in our schemes, the feedback from all receivers could be sent within the same time slot as long as a single legacy 802.11 ACK. So, the time consumed for one round of gathering feedback is short and independent of the number of receivers. Since real-time multicast applications typically require strict delay constraints but can tolerate a certain residual error rate, the proposed ACK/NACK jamming based protocols could be a suitable option.

1.2. Thesis Contributions

While working on the problems described above, we have extended the current state of the art with the following main contributions.

- We explore the potential feedback mechanisms for MAC layer multicast in IEEE 802.11 wireless LANs. Besides the polling scheme, feedback (ACK or NACK) aggregation is a potential candidate. However, pure NACK aggregation has fake detection problems which cause high residual error rates or severe unnecessary retransmissions. The feedback jamming, which is the aggregation of an ACK and NACKs in the same time slot, avoids the fake detection problems and could be a good candidate. In particular, one receiver is selected as the leader, NACK frames are allowed from the non-leader receivers to destroy the ACK frame from the leader receiver in the same time slot and prompt retransmissions from the sender.
- Based on the feedback jamming mechanism, we enhance a Leader Based Protocol (LBP) with a MAC control frame carrying the Sequence number (called SEQ-LBP). Initially, LBP is not reliable for the non-leader receivers and has poor performance at high error rates due to no guiding frame or sequence check. SEQ-LBP solves the problems of LBP well. All the non-leader receivers can send feedbacks according to

the timers set based on the SEQ frame. Both the leader receiver and non-leader receivers reply ACK and NACK respectively based on sequence check, hence it avoids the unnecessary transmissions in LBP. SEQ-LBP needs the minimum number of redundancy transmissions among all pure ARQ based schemes.

- To overcome the scalability limitation of pure ARQ schemes, we combine SEQ-LBP and packet level FEC and propose a Hybrid LBP (HLBP). Using a block coding, parity packets are generated from a block of original data packets. HLBP transmits a block of original data packets using raw broadcast and retransmits parity packets using an improved SEQ-LBP which is based on block feedback. HLBP is much more efficient than both LBP and SEQ-LBP especially for large multicast groups. HLBP needs the near-minimum number of redundancy transmissions among all packet level schemes. LBP, SEQ-LBP and HLBP are all back compatible to legacy 802.11 stations.
- We analyze the performances of LBP, SEQ-LBP and HLBP over the simplified Gilbert-Elliot (SGE) channel model [Gil60] [Ell63] [Mus89]. We also evaluate their performances on NS-2[1]. The simulation results verify the theoretical analyses and show the advantages of the proposed protocols. Due to the SEQ frame, SEQ-LBP avoids the problems of LBP and is more efficient in various scenarios, especially for large multicast groups. Due to block coding and block feedback, HLBP is much more efficient than both LBP and SEQ-LBP and has a superior scalability with respect to the number of receivers per multicast group. Moreover, simulation results confirm that SEQ-LBP outperforms the application layer ARQ schemes with both a shorter multicast delay and a higher efficiency. Meanwhile, with the same delay constraints, HLBP is more efficient than the application layer HEC schemes. SEQ-LBP is good for small multicast groups due to its short delay, effectiveness and simplicity while HLBP is better for large multicast groups because of its high efficiency and high scalability.
- We confirm the feasibility of ACK/NACK jamming through theoretical analyses, NS-2 simulations, as well as measurements on a real test-bed. Using *Atheros* chipset

[1] The Network Simulator ns-2. http://www.isi.edu /nsnam/ns.

along with the *Madwifi*[1] driver, we design a flexible software platform that runs in real-time Linux. Our platform supports microsecond precision and packet transmissions at a configurable time and frame format by not triggering hardware level CSMA contention or backoff schemes. Based on the platform, we implement a driver level dynamic leader selection algorithm and a multicast management approach. By hundred hours of tests (each one lasts for several hours) under various scenarios, we found that the hardware ACK/NACK jamming probability can be as high as 0.99+ for normal scenarios (about 0.90+ for the worst case with only two receivers which even experience nearly the same channel condition) when a simple dynamic leader selection algorithm is used. As a result, it is confirmed that, for the first time, the hardware ACK/NACK jamming can be used as a multicast feedback in the design of MAC layer reliable Multicast.

- We have implemented a driver level SEQ-LBP with dynamic leader selection and multicast management on the test-bed and confirm its performance for recovering multicast packet losses. Based on this we assume that a SEQ-LBP implementation, if incorporated into the wireless modem and thus with more precise timing, is an effective and efficient MAC layer multicast ARQ mechanism. Our driver level SEQ-LBP can replace the normal MAC broadcast in the *Madwifi* driver and provide a reliable MAC layer multicast service for real applications.
- We also evaluate a pure NACK jamming based ARQ scheme on the test-bed and explore the fake detection problems of pure NACK aggregation. Busy tone and physical subcarrier based multicast schemes suffer from the same or similar limitations as well. These limitations should be considered for the design of MAC layer multicast and cross MAC and Physical layer multicast for wireless networks. Moreover, we also talk about the fine granularity MAC layer coding, data rate adaptation mechanisms and cross-layer issues which could enhance the proposed feedback jamming based protocols.

[1] The Madwifi Project. http://madwifi-project.org/.

1.3. Thesis Organization

The organization of the rest thesis is as follows. Chapter 2 gives an overview of existing and evolving multicast error recovery technologies: ARQ, FEC and HEC. We also discuss the use of these technologies in different network layers in practice, such as the physical layer, MAC layer and application layer. Related work about MAC layer multicast is also reviewed in detail.

In Chapter 3, we introduce two channel models: the Independent Identical Distribution (i.i.d) channel model and the Gilbert-Elliott (GE) channel model, which are often used for evaluating the performances of different error recovery schemes over erasure error channels in literature. We also discuss the delay budget for end-to-end multicast error recovery schemes (ARQ, FEC and HEC) and MAC layer approaches.

In Chapter 4, we first discuss the potential feedback mechanisms for MAC layer multicast and compare the feedback jamming scheme with other feedback aggregation schemes. Then the ACK/NACK jamming based protocols are described: LBP, SEQ-LBP and HLBP. We also discuss some related issue for these protocols, such as the introduction of FEC in the MAC layer.

Chapter 5 is focused on performance analyses. We first analyze the ACK/NACK jamming probability over the Rayleigh channel model and then calculate the performances of LBP, SEQ-LBP and HLBP over both the i.i.d channel model and the GE channel model. We also discuss how to choose the protocol parameters to optimize their performances.

In Chapter 6, we first present the NS-2 simulation environment and then show the simulation results of the feedback jamming probability and protocols (LBP, SEQ-LBP and HLBP). Next, we compare the simulation results with the analysis results based on Chapter 5 to verify the analyses. The performances of these schemes are compared in various scenarios: different jamming probabilities, different channel conditions, different number of receivers, etc. We also compare these schemes with the block-polling based scheme and with some application layer multicast error recovery approaches.

Chapter 7 presents our work on a hardware test-bed. First, we introduce our *Madwifi* test-bed and the implementations on it: the ACK/NACK jamming scheme, a MAC layer multicast management scheme and a dynamic leader selection algorithm. Then, we present the measure results on the test-bed that confirm the feasibility of ACK/NACK jamming. Finally, we describe the implementation of the driver level SEQ-LBP and present its experiment results.

In Chapter 8, we talk about some accessory techniques for the feedback jamming based protocols. We first present the experiment results of feedback aggregation through a pure NACK jamming based MAC layer multicast approach. Then a MAC layer FEC coding with a fine granularity is discussed. Moreover, we also talk about the potential data rate adaptation mechanisms and crossing layer issues for the proposed feedback jamming based protocols.

Finally in Chapter 9, we make conclusions about the presented work, discuss some directions for future work and present a list of publications resulted from this work.

Chapter 2

Overview of Multicast Error Recovery Techniques

Typically, there are three types of error correction techniques for multicast delivery: Automatic Repeat reQuest, Forward Error Correction erasure coding and Hybrid Error Correction which is the integration of ARQ and FEC. This chapter gives an overview of these techniques. We also discuss the use of these technologies in different protocol layers in practice, such as the physical layer, MAC layer and application layer. Moreover, the related work about the MAC layer multicast is reviewed in detail.

2.1. Automatic Repeat Request

2.2.1 Basic Scheme

Automatic Repeat Request is a basic error control technique for data transmissions which uses feedback, timer and retransmissions. The source will automatically retransmit a packet if it is not convinced that the destination has received the packet correctly. Transmission errors are examined at receivers via an error detecting code such as CRC. Detection of a packet loss at the sender may be via a protocol timer, by detecting missing positive acknowledgement, by receiving explicit negative acknowledgement and/or by polling the receiver status. In a multicast scenario, after knowing which data packet is lost, the sender will multicast the lost data packets to all of the receivers for recovering the missing data packets. There are two basic ARQ schemes: Stop-and-Wait ARQ (Figure 2.1) and sliding-window ARQ [Lin93] [Pet03] [Fai02] (Figure 2.2 and Figure 2.3).

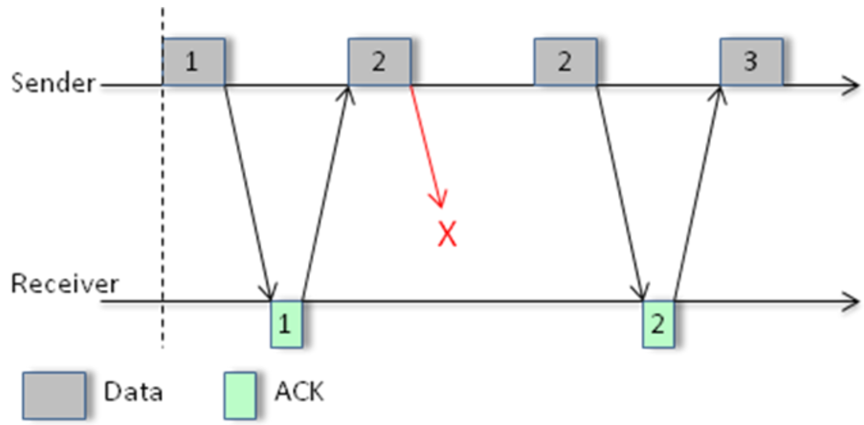

Figure 2.1: Stop-and-Wait ARQ

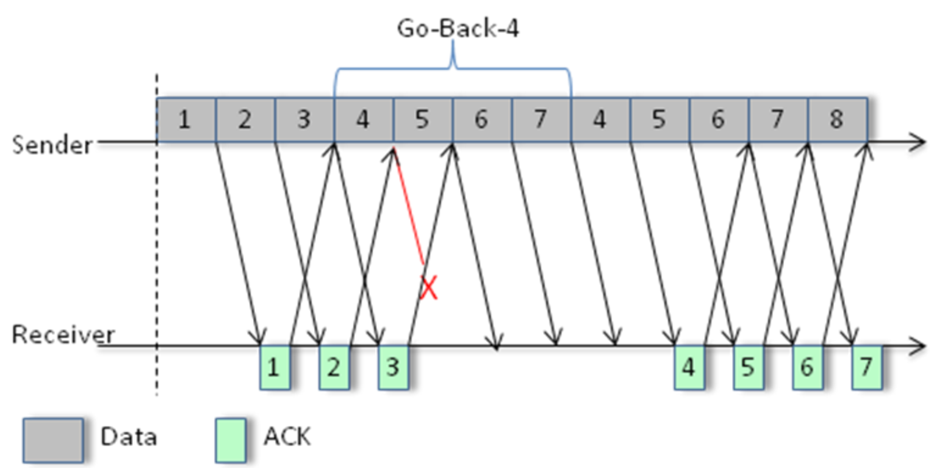

Figure 2.2: Go-Back-N ARQ

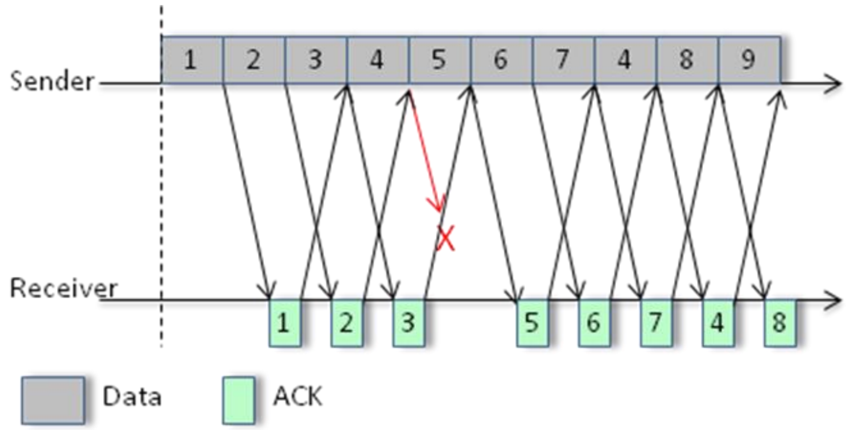

Figure 2.3: Selective Repeat ARQ

In the Stop-and-Wait ARQ, the sender transmits a packet and then it stops to wait for an acknowledgement from the receiver. The sender will perform a retransmission if the timer for this packet expires, which indicates either the loss or the corruption of the packet. As the sender must wait for the acknowledgement, there can be only one packet in transmitting at a given time. Therefore, the Stop-and-Wait scheme is inefficient. It is, however, easy to implement and thus popular for that reason.

Contrary to Stop-and-Wait, sliding-window ARQ allows several packets to be in transit at a time. Sequence numbers are used to number packets. Both the sender and the receivers have a window of packets. The sender’s window indicates the next packet it is allowed to send and the receiver’s window indicates the packet it is expecting. Two basic sliding-window ARQ approaches are Go-Back-N (Figure 2.2) and Selective Repeat (Figure 2.3), and the second one is also known as selective reject ARQ. In the Go-Back-N ARQ, if an erroneous packet is detected (by sequence check), receiver cancels its ACK or sends a NACK for that packet. When the sending window is full or receiving a NACK, the sender retransmits this packet and all succeeding packets that had been already sent after the erroneous packet. Selective Repeat ARQ retransmits those packets that are negatively acknowledged or are not acknowledged in time. With selective repeat, the amount of retransmissions is minimized, but on the other hand it is more complex than Stop-and-Wait and Go-Back-N.

About the choice between ACK and NACK, for common wireless scenarios, the link error rates are far less than 50% (e.g. the worst PLR is only about 10% in wireless LAN with IEEE 802.11 [Fuj04]), so NACK based schemes are more efficient than ACK based ones. Due to this reason, most real-time multicast transport protocols for high speed networks proposed to use NACK based mechanisms rather than ACK based ones [Arm92] [Ott04] [Pej96] [Pin94]. However, please note that ACK is a more effective way to indicate the success of receiving the data packet. This is because the disappearance of NACK does not mean the data packet is 100% received correctly as the NACK may be lost due to channel errors. Contrast to NACK, ACK is an explicit and doubtless indication.

As aforementioned in chapter 1, in case of a large number of receivers, an ACK for every packet from every receiver would lead to a so–called “ACK implosion” at the sender, which would be busy with processing a large number of ACKs and would have little time to transmit data. Another problem with an ACK–based approach is that the sender has to maintain a list, for each packet, of the receivers from which it received an ACK. This could be an important overhead, especially when the number of receivers is very large. Due to the possibility of ACK implosion and the requirement of maintaining the identity of each receiver, an ACK–based approach for reliable multicast is generally not considered to be scalable [Tow97]. A NACK–based approach is more practical because the number of NACKs is likely to be much less than the number of ACKs. Also, with a NACK based approach the sender does not need to be aware of the number and identity of receivers; the responsibility of loss recovery is now on the receiver – a receiver wishing a retransmission simply sends a NACK to the sender. However, when the network is large and the loss probabilities are high we can have a NACK implosion too. Hence one of the most important challenges in designing reliable multicast is to deal with NACK implosion.

To deal with feedback implosion, receivers can multicast the feedback in the group instead of just unicasting it to the sender alone. A receiver, on detecting a loss, waits for a random amount of time and multicasts a NACK to all the group members. If another receiver hears the NACK and determines that its own pending NACK is subsumed, then it cancels its own

NACK. However, for application layer multicast, using multicast for feedback transmission is less reliable than unicast because unicast is reliable in the MAC layer while multicast is not. Moreover, due to the random timer and as the feedback has to be transmitted to the whole multicast group, feedback multicast takes a longer time than unicast in multi-hop wireless networks. Multicasting NACKs is proved to be useful when receivers are locally concentrated with small propagation delay; when receivers are far apart with large propagation delay, however, multicasting NACKs is proved to have a negative impact on delay properties, due to the large propagation delay of NACKs to other receivers and to the time that receivers must wait to achieve the desired suppression of NACKs [Pej96]. Consequently, using unicast or multicast for feedback is a tradeoff between feedback suppression and performance on reliability and delay. Please also note that for multi-hop multicast, feedback can be aggregated by building trees or hierarchies of receivers, routers, or servers with the sender at the root of the tree (i.e., at the highest level of hierarchy). Here, receivers send feedback to their parent nodes at the next higher level of hierarchy. The parent nodes aggregate feedback before forwarding them up the tree towards the sender.

The difference between ARQ and FEC is that ARQ is inherently channel adaptive, as only lost packets are retransmitted, while the introduction of FEC adds overhead even if the channel is clean. However, on poor channels ARQ may introduce significant delays due to the roundtrip propagation time of the retransmission request and its response. These delays significantly limit the applicability of ARQ to multimedia communications.

Furthermore, ARQ scales very badly to large sets of receivers. For a packet loss rate of p, and a set of R receivers experiencing independent losses, the probability that every single data packet needs to be retransmitted is $1-(1-p)^R$, and this value quickly approaches unity as R gets large. This also results in a high average number of transmissions per packet. The situation does not improve if losses at different receivers are partly correlated: when the group becomes large, so does the number of subsets of receivers with uncorrelated losses, and the same phenomena appears, only at a different scale. Scalability problems also exist in handling feedback from the receivers: in fact, the sender must handle distinct error reports from every receiver, resulting in a high number of average reports per transmitted packet if plain ACK/NACK is used.

Due to the simplicity and effectiveness, ARQ is widely used for data communication in both wired and wireless networks. TCP is a typical go-back-N ARQ based protocol. It has been proved that TCP can work very well for guaranteeing the reliability of transmissions for non-real time services. However, TCP does not provide any mechanism to guarantee the end-to-end transmission delay so that it is not suitable for real-time services with strict delay constraints. As shown in Chapter 1, 802.11 DCF is a typical Stop-and-Wait ARQ based protocol. Moreover, both TCP and DCF are applicable only for unicast. The applications of ARQ for multicast in different layers are discussed in the following subsection.

2.2.2 Related Work

Besides LBP, there are a few proposals for the MAC layer multicast in wireless LANs. Polling is a straight way to apply ARQ error control scheme for multicast in the MAC layer [Tgaa10]. After transmitting the data frame, the sender polls each receiver to expect an ACK frame. Any loss of ACK leads the sender to retransmit the data frame until the retry limit is reached. A more efficient polling scheme is that the sender sends a request acknowledgement frame to arrange for each receiver to reply feedback at a scheduled time. Block acknowledgement can improve the efficiency of polling further. Sequence numbers are used to number frames. A block of data frames can be transmitted at a time once the channel is granted. Then the sender polls all receivers and all receivers reply block acknowledgement which indicates the transmission result of each data frame of the block (may be based on bitmap). Block polling is more efficient than single polling but with a higher delay. Although MAC layer polling scheme is effective to recover the multicast losses, it is just a pure ARQ protocol anyhow and has a poor scalability with respect to the number of receivers per multicast group.

Standard probabilistic approaches [Kur99] [Ake04] can also be used to tackle the feedback collision problem of MAC layer multicast as we described in Chapter 1. In the "delayed feedback" scheme, receivers wait a random time before replying feedback (CTS or ACK). This scheme can also be based on NACK. In this case, hearing a NACK can lead other receivers to cancel their own NACK if they have it [Ake04]. This feedback suppression scheme reduces load on the network especially for large multicast groups. In the "probabilistic feedback" based schemes, instead of waiting for a random number of time slots to send feedback, the group members send out feedback in the same time slot but with a certain probability. Although probabilistic approaches can reduce the feedback collision, they cannot avoid feedback collision completely. And the potential feedback collisions cause additional delay and reduce channel utilization. Moreover, the choice of right parameters for waiting times and probability of sending feedback is difficult because it is dependent upon the number of group members and even the channel condition of each receiver.

Another approach to tackle the feedback collision and to recover multicast losses is to select one of the receivers as the leader or called collision detector which is responsible for replying multicast transmission result to the sender [Tou98] [Pen02]. This approach takes a short time but it cannot guarantee the reliability at the other receivers because feedback is gathered only from the leader/detector, the choice of which affects the total multicast performance.

Gupta et al. proposed another different way to reply multicast transmission results using busy tone (CTS-tone and NACK-tone) which is in an additional channel [Gup03]. When the RTS is sent, the sender node does not expect to receive a CTS packet. Instead, it listens on the signaling channel to see if any node is transmitting an NCTS tone. If no such tone is sensed, the sender begins the transmission of the multicast data packet. At the end of the packet transmission, the node senses the signaling channel again to see if any node is transmitting a NACK tone. If there is no NACK tone, the sender assumes that the data transmission was successful. However, this approach requires an extra channel for the busy tone, which is not practical for common wireless LANs. Moreover, it is hard to distinct a tone from collision or interference because the tone is not a frame that can be interpreted and protected by a CRC check. Basalamah et al. proposed a similar approach based on pure NACK [Bas06], where each receiver replies a NACK in the same time slot following the data frame if necessary and a single NACK or a detection of a collision prompts the sender to retransmit the data frame. This approach suffers from a similar challenge of collision detection. We will also evaluate the pure-NACKs jamming based ARQ scheme in this thesis and discuss the collision detection.

Another approach called the orthogonal frequency division multiplex access (OFDMA) based multicast ACK (OMACK) [Dem06] [Kim08] is similar to the busy tone based scheme but using single wireless channel. OMACK uses one OFDM symbol for the ACKs from all receivers, and each receiver indicates its packet reception status by utilizing a subcarrier tithing method (denoting ACK or NACK, similar to bitmap). OMACK can significantly reduce the overhead of multicast feedback, as a consequence, improves the performance of wireless networks. However, this approach needs new hardware supports and even a cross layer (MAC & PHY) design. Moreover, similar to tone based schemes, it suffers from the challenge to interpret the symbol (to distinct it from collision) as it is not a frame which can be interpreted and protected by a CRC check.

Collision avoidance schemes also can reduce the multicast losses, especially the collision losses. Sobrinho et al. proposed a blackburst scheme in [Sob96] to avoid collisions in multicast distribution. The length of the blackburst is determined by the station’s waiting time. The station with the longest blackburst will gain access to the channel. An early multicast collision detection solution is proposed by Nilsson et al. in [Nil02], which sends an early multicast collision detection packet and works in an ideal Extended Service Set (ESS) with no hidden stations. As there is no retransmission phase, these two schemes cannot correct path fading losses and even cause constant overhead in the network.

Currently, application layer ARQ based schemes are commonly used to recover multicast losses in wireless LANs [Non98a] [Tan07a] [Tan09]. It is shown that ARQ schemes are effective to repair multicast packet losses for small groups with low error rates, even though they always result in long multicast delays due to application layer protocol waiting, MAC layer queuing, hardware handling, etc.

In summary, all these approaches discussed above are pure ARQ schemes and hence they are not efficient for large multicast groups due to the limitation in scale. However, for small multicast groups, ARQ is still a good choice due to the effectiveness and simplicity to be implemented. We will evaluate the performance of some pure ARQ schemes in this thesis, such as MAC layer block-ACK polling and LBP. We also introduce the application layer ARQ scheme to compare it with our MAC layer approaches.

2.2. Forward Error Correction

2.2.1 Basic Scheme

Another method for multicast error control is forward error correction. Forward error correction is applied to a block of source data packets to produce extra parity packets that are sent along with the data packets. The FEC code is carefully designed to be able to protect the data against channel erasures under most circumstances. FEC therefore provides error resilience by increasing the amount of data to be sent. FEC does not require a return channel and is typically not adaptive to the current state of the channel. Also, FEC techniques do not guarantee that the data will arrive to the receiver without errors. FEC operation is illustrated in Figure 2.4. For conventional FEC, a set of data packets are transformed into fixed number of coded packets. If the number of erased packets is less than the decoding threshold for the FEC code, the original data can be extracted intact. One popular class of erasure correction codes, Reed-Solomon (RS) codes [Mor02] [Riz97a], have desirable optimality properties. Due to the decoding complexity, the most commonly used RS codes operate on symbols of bytes and restrict the code parameters as $k \le n \le 255$ [Riz97a] which can satisfy most practical real-time applications with strict delay constraints. Other classes of erasure correction codes offering longer block lengths exist, including a family of very fast and rateless but suboptimal (requiring on average $(1+\varepsilon)k$ - where $\varepsilon > 0$ - packets to recover the k source packets) Low Density Parity Check (LDPC) codes [Mor02]: Tornado codes [Lub97], Luby Transform codes [Lub02], Raptor codes [Sho06], etc.

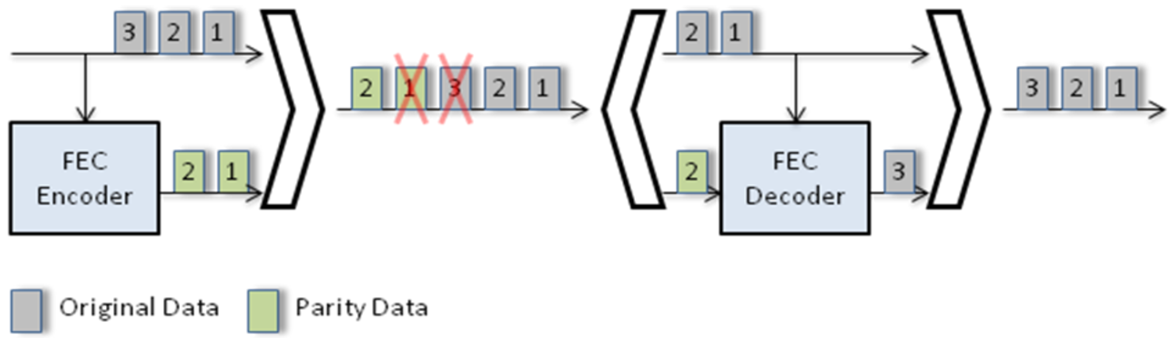

Figure 2.4: FEC

Reed-Solomon codes are an instance of a larger class of linear block codes. They are systematic codes, which means that the code words contain the original data in unmodified form along with added parity symbols. Reed-Solomon codes can be described in terms of two numbers, (n,k), where n is the length of the code word, and k is the number of data symbols in that code word. (Therefore, a (255,205) Reed-Solomon code consists of 205 data words and 50 parity words.) Each symbol is drawn from a finite field of 2^s elements, where s is the number of bits to be represented in each symbol. Typically, 8 bits per symbol are used. The total number of words in the code is equal to $2^s - 1$; however, by replacing some data symbols with known values, we can realize smaller codes. For example, by replacing 105 data symbols in a (255,205) code with zeros, we create a (150,100) code.

Although Reed-Solomon codes can be used to correct errors, erasures, or both, particularly efficient decoding algorithms based on Vandermonde matrices [Riz97b] exist if only erasures are to be corrected. In this case, each parity symbol can correct one missing data symbol. This means that the original codeword (and, therefore, the original data) can be recovered if at least k of the original n symbols are received intact.

Block codes can be easily used to create packet erasure codes by simply striping the code words across packets, so that each packet contains one symbol from each of a large number of Reed-Solomon code words. If this is done, a packet erasure will erase one symbol from each code word; this can be corrected using the parity symbols contained in one parity packet. In this way, a (n,k) packet erasure code, where all k data packets can be decoded if at least k packets arrive, can be created.

Implementing FEC is computationally expensive, since the entire data stream must be processed to produce the encoded packets, each one conveying information on a number (possibly as large as k) of source data packets. This is not a concern in telecommunication systems, where the encoder/decoder is usually implemented as a dedicated piece of hardware and is usually much cheaper than having a feedback channel. But in computer communications, the feedback channel is often available and implementing FEC means a noticeable overhead for the main processor.

In multicast protocols, however, the use of FEC techniques has completely different motivations. The encoding is mainly used to remove the effect of independent losses at different receivers: thanks to the encoding, as long as a receiver collects a sufficient number of different packets, reconstruction of the original data is possible independently of the identity of the received packets. This makes protocols scale much better irrespective of the actual loss pattern at each receiver. As an additional bonus, the dramatic reduction in the residual loss rate (after decoding) largely reduces the need to send feedback to the sender, thus minimizing the use of the uplink channel, and simplifying feedback handling. The main limitation of FEC techniques is that they cause constant overhead for the networks even when the channel is in good conditions. Moreover, FEC techniques do not guarantee that the data will arrive to the receiver without errors. Adaptive FEC [Mor02] [Tan09] can relieve both of the limitations where the code is adaptive to the measured or predicted channel condition.

2.2.2 Related Work

FEC coding (convolutional coding) is used in the physical layer of 802.11a/g with a coding rate of 1/2, 2/3, or 3/4 (802.11b does not use FEC.). A new convolutional code rate of 5/6 is added to 802.11n standard which uses LDPC code based FEC [11n09]. Although physical layer FEC is effective to protect the data bits, it cannot guarantee the packet reception ratio. The residual packet error rates are still high, e.g. the PLR in the worst case can be as high as 10% in wireless LAN with IEEE 802.11 [Fuj04]. We do not discuss physical layer error correction in detail in this thesis.

FEC coding can also be used in the MAC layer to protect the data. A MAC layer bytes level FEC has been proposed by Choi et al. in [Cho06]. The MAC payload is split into multiple blocks, which are encoded using block FEC codes. Parity bytes are carried following each block. And when the errors cannot be recovered by the FEC, MAC layer retransmission is also used. Although it is effective to correct the bit errors, MAC layer bytes-level FEC causes fixed overhead even under good channel conditions. Moreover, the existing MAC layer FEC schemes (with retransmissions) are only for unicast.

Packet level FEC are always used in the application layer for multicast error correction. Raptor codes based FEC have been standardized by DVB for IPTV applications [Ets07] and by 3GPP for MBMS services [3gp05a]. To reduce the constant overhead of FEC, rate adaptive FEC have also been developed which change the code rate adaptive to the measured or predicted channel conditions [Ana07]. Anyhow, pure FEC based schemes cannot guarantee the final packet receipt ratio as there is no feedback and retransmit process. As a result, FEC are always combined with ARQ and correct the multicast error cooperatively, which will be discussed in detail in the following section.

2.3. Hybrid Error Correction

2.2.1 Basic Scheme

The integration of FEC and ARQ are often referred to as Hybrid Error Correction, HEC, Hybrid ARQ or HARQ. First ideas for systems combing error correction and ARQ were from the work of Wozencraft and Horstein in 1960 [Woz60] [Woz61]. A historic overview of further development of such techniques, now known as HARQ, can be found in [Cos98] [Lot07]. Afterwards, many researchers studied plenty kinds of bit-level and packet level HARQ. The studies [Qia00] [Car97] [Tan07b] indicate that HEC schemes are much more efficient for recovering multicast data packets than the schemes with either FEC or ARQ alone.

We first consider bits level HEC schemes. In standard ARQ, error-detection information bits are added to data to be transmitted (such as CRC). In Hybrid ARQ, FEC bits are also added to the existing Error Detection bits (such as Reed-Solomon code or Turbo code). As a result Hybrid ARQ performs better than ordinary ARQ in poor signal conditions, but in its simplest form this comes at the expense of significantly lower throughput in good signal conditions. There is typically a signal quality cross-over point below which a simple Hybrid ARQ is better, and above which a basic ARQ is better.

The simplest version of HARQ, Type I HARQ, encodes the data bits and CRC with FEC. When a packet is found to be in error, a feedback will be sent to the sender and the sender will then retransmit the same packet until the packet is successfully decoded at the receiver or a maximum retransmission limit is reached. In the case where the erroneous packets are stored in a buffer and the corresponding soft values at bit level are combined according to the weights of received signal to noise ratio (SNR), the method is also known as Hybrid ARQ Type I with Packet Combining or Chase Combining [Beh07]. Note that in HARQ Type I, the code rate used in retransmissions is the same as used in the first original transmission. HARQ Type II is also known as full Incremental Redundancy (IR) and this technique gradually decreases coding rate in each retransmission by sending additional redundancy bits. These bits will then be combined with the previously received packets which were stored at the buffer of the receiver to form more powerful error correction codes. HARQ Type III is known as partial IR. This method decreases coding rate by sending additional redundancy bits while maintaining self-decodability in each retransmission. The retransmitted packet can be chase combined with the previous packet to increase the diversity gain. HARQ Type I is adopted in 3GPP (R99), WiMAX[1], HSDPA[2] and ITU-T G.hn[3], a high-speed Local area network.

Now we consider packet level HARQ technologies. Currently, there are two main packet level HARQ schemes used in the application layer [Car97] [Non97] [Den95]. One possibility for combing packet level ARQ and FEC, referred to as HARQ Type II (shown in Figure 2.5), is not to send any parity packets (redundant packets) with the first transmission, but to send parity packets when a retransmission is required (e.g. receiving a NACK). Note that error recovery by multicast retransmission of a single parity packet allows all receivers to recover their different single lost packet. As shown in [Non97], this approach is very bandwidth-efficient for reliable multicast to a large number of receivers. Another approach that combines ARQ and FEC, referred to as HARQ Type I, immediately sends a certain amount of parity packets following the original data packet in the first transmission. If the loss rate obtained after reconstruction at the receiver is still too high, more parity packets are retransmitted. Many studies show that HARQ Type I & II are efficient to correct multicast losses for a large number of receivers [Car97] [Qia00] [Tan07b] [Tan09]. In the rest of this thesis, we refer to HARQ or HEC as only packet level HARQ approaches if not otherwise stated.

---

[1] WiMAX forum, http://www.wimaxforum.org/

[2] Introduction of HSPDA, http://en.wikipedia.org/wiki/HSPDA

[3] A forum for ITU-T G.hn, http://homegridforum.typepad.com/

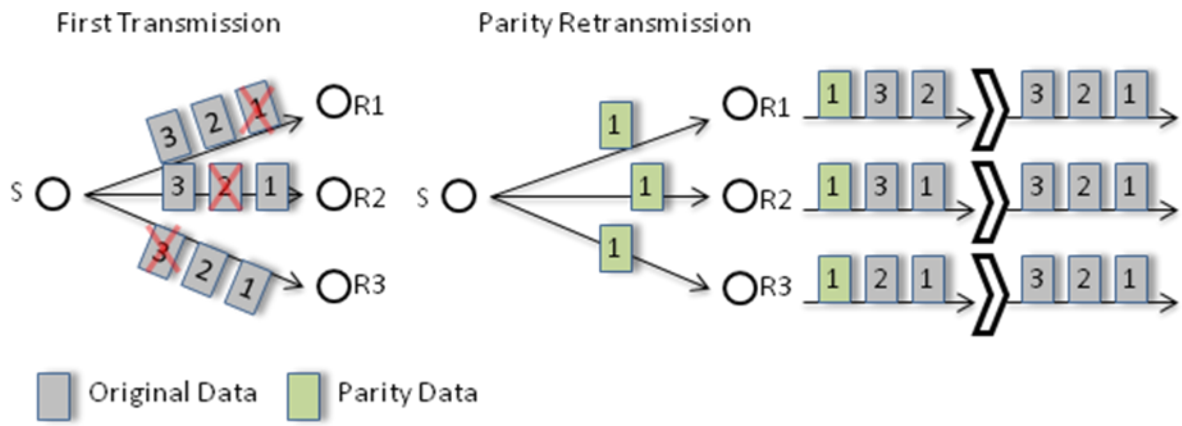

Figure 2.5: HARQ Type II

The maximum number of transmissions per codeword is usually determined by relevant constraints (e.g., delay) in the system [Lot07]. Implementation complexity is the key reason for restricting the maximum number of transmissions. The data packets must be stored in the receiver up until the last transmit process, and the memory requirements at the receiver are bound by this maximum number of transmissions. Although the worst-case delay can be limited by restricting the number of transmits, a solution that disregards complexity issues would be to allow the transmitter to make a QoS tradeoff (i.e. delay vs. SNR efficiency by controlling transmit power). Frequently, the transmitter can decide on the maximum number of transmissions appropriate for a given QoS class, but within the limits of a maximum number of transmissions that keeps the implementation complexity bound.

Another interesting issue arising is this context is the ACK/NACK loss. Each HARQ transmission cycle has an associated ACK/NACK transmission, with appropriate erasure/error thresholds on its decoding. An ACK/NACK error leads to small performance losses, as in the case of an ACK error, the transmitter retransmit the data packet which is unnecessary, while in the case of a NACK loss, the transmitter terminating the transmission of that packet, which results in data losses, as the block has not been successfully decoded yet. This problem can be relieved by some special enhancement for the transmission of ACK/NACK. For example, the base transmission rate is used for the transmission of

ACK/NACK frames in the MAC layer [Kur99] [Li09] and MAC layer unicast can be used for the transmission of ACK/NACK [Pej96] [Tan07a].

2.2.2 Related Work

Recent interest in the HARQ scheme comes from the quest for reliable and efficient transmission under fluctuating conditions in wireless networks. An information-theoretic analysis of some HARQ protocols, concerning throughput and average delay for block-fading Gaussian collision channels have been reported in [Cai01] [Tun02] [Ses04]. Another line of recent work on HARQ is concerned with the mother code and its puncturing. Given the number of parity bits which are at each stage omitted from the mother code (i.e., punctured and not transmitted), their identity is determined by a puncturing pattern. The throughput of HARQ schemes is strongly affected by the power of the mother code used in the system and the family of codes obtained by puncturing. Thus recently proposed HARQ schemes use powerful turbo codes, and the design of puncturing patterns is an important issue [Row00] [Nar97] [Aci99].

Deng et al. proposed a Type-I hybrid ARQ system which automatically adjusts the error correcting code rate to match the current channel bit error rate [Den95]. Joe et al. designed a hybrid ARQ scheme with concatenated FEC for wireless ATM networks [Joe97], and the key idea is the adaptation of the code rate to channel conditions using incremental redundancy to maximize the throughput efficiency. In [Qia00], Qiao and Shin proposed a two-step adaptive hybrid ARQ scheme for transmitting H.263 video sequences over wireless LANs, which, (i) based on both the wireless channel conditions and the deadline constraint, adaptively selects the best error correction code by looking to an optimal code table which is predetermined before starting the video service, and, (ii) based on the actual frame loss events, adaptively uses the prebuilt optimal code table to guarantee certain quality of service in terms of frame loss rate. Tan et al. developed formulas to optimize the performance of

HARQ Type II while guaranteeing the required PLR under strict delay constraints [Tan07b] [Tan09].

2.4. Conclusion

In this chapter we reviewed the multicast error recovery techniques of ARQ, FEC and HEC and their applications in wireless networks. In general, for reliable multicast delivery under strict delay constraints, HARQ schemes are more bandwidth efficient than pure ARQ or FEC scheme alone. However, single FEC schemes could be the best scheme in some special cases that a critical short delay is required (e.g. gaming), or even no retransmission round is allowed (e.g. no return channel). Similarly, for some other special cases, pure ARQ could be the best scheme, e.g. there are only a few receivers in a multicast scenario or the entire receivers having high correlation. In summary, the design of multicast error recovery schemes has to be combined with the practical multicast scenarios.

The work presented in this thesis is focused on MAC layer multicast error recovery in wireless LANs for real-time multimedia multicast applications, which always require strict time constraints but can tolerate a certain residual loss rate. We develop two approaches for small multicast groups and large multicast groups respectively. The first approach, called SEQ-LBP, is a pure ARQ based scheme. Based on ACK/NACK aggregation technique in the MAC layer, SEQ-LBP is efficient, more scalable than general ARQ schemes, and of short delays. Due to its good performance and simplicity, SEQ-LBP is a good choice for small multicast groups such as Wireless Home Networks (WHN). Our second approach, called HLBP, is a MAC layer HARQ scheme which combines packet level FEC and SEQ-LBP together. Although with implementation complexity, HLBP is a good choice for large multicast groups such as video conference, due to its high efficiency and scalability.

Chapter 3

Channel Model and Delay Budget

As introduced in Chapter 1, the wireless channels are error-prone and error recovery schemes have to be applied to provide acceptable quality for multicast services in wireless LANs. Usually, bit-wise channel coding is used in the physical layer to recover bit errors. However, the bit-wise channel coding cannot recover burst errors longer than code words and hence the residual error rates in the physical layer are still high. Consequently, more error recovery schemes have to be applied in upper layers. Normally, the channel errors are considered as packet erasure error in the MAC and upper layers as the erroneous packets that fail the CRC check will be thrown away by the MAC layer even there may be only one bit error in the packet. In this chapter, we discuss two packet erasure channel models to support the performance evaluation for the multicast error recovery approaches that will be discussed in the following chapters. The first channel model is the independent and identically distributed channel model, which is often used due to its simplicity, while the other one is the Gilbert-Elliott channel model, which is more accurate but of a higher analysis complexity. We also discuss the delay budget of end-to-end multicast error recovery approaches and MAC layer approaches in this chapter.

3.1. Erasure Channel

We first discuss the features of the erasure channel. In the case of erasure channel, the input symbol is not erroneously received but rather deleted (Figure 3.1). Note that the transport protocol must provide methods to detect such a deletion. Following the calculation in [FMI09], the capacity of the erasure channel is $1-p_e$, where p_e is the symbol error rate or called erasure probability. So it is concluded that the channel capacity of the erasure channel is linearly dependent on the erasure probability. From this result the required redundancy of information transmission over the erasure channel can be derived as $p_e/(1-p_e)$. Note that this rate only includes the redundancy required for the correction of the packet losses. The practically required redundancy information additionally includes the means for detection and identification of packet losses.

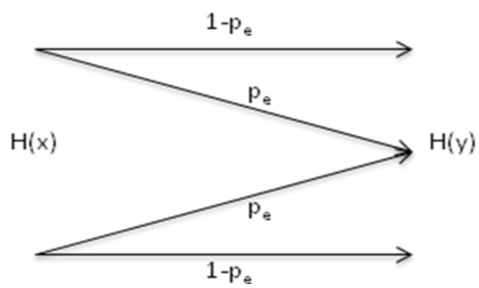

Figure 3.1: Erasure Channel

As shown before, multimedia applications typically are loss tolerant, which means that correction of the transport down to an arbitrarily small residual error rate is not required. Figure 3.2 shows the model for an erasure channel with residual errors. The overall channel can be simply considered as the cascade of two erasure channels, whereas the overall packet loss rate equals the packet loss rate of the overall channel and the packet loss rate of segment two equals the allowed residual error rate. The overall packet loss, given the model depicted in Figure 3.2, is $p_e = 1-(1-p_e[1])(1-p_e[2]) = 1-(1-p_e[1])(1-p_{res})$ and therewith

$p_e[1] = (p_e - p_{res})/(1 - p_{res})$ given $p_e \geq p_{res}$. So the first segment of the cascade can now be interpreted as being an erasure channel with a residual output loss rate of p_{res}.

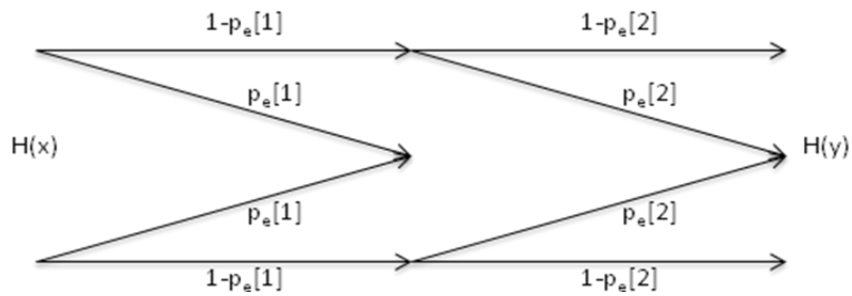

Figure 3.2: Erasure Channel with Residual Error

3.2. i.i.d Channel Model

The i.i.d channel model (sometimes referred to as canonical model) is a memory-less channel, where each packet error has the same probability distribution as the others and all are mutually independent. This feature (or assumption) of i.i.d channel model simplifies the underlying mathematics of performance analysis for protocols over the packet erasure model. Even though the i.i.d channel may not be realistic in practical wireless environments, it is still often used to analyze and compare the performances of error recovery schemes due to its simplicity [Kur99] [Tow97] [Tan07a]. In this thesis, we adopt the i.i.d channel model as one candidate of two channel models to support the performance analyses and comparison of our proposed MAC layer multicast error recovery approaches.

Now, we take two examples to show how to use the i.i.d channel model, which are the basic mathematics of the theoretical analyses for our approaches. First, let us consider a single ACK based ARQ scheme (Stop-and-Wait) for one sender and one receiver (like 802.11 DCF) over the i.i.d channel model. The sender transmits data packets to the receiver over a i.i.d channel with a error probability p. If the data packet is received correctly, the receiver replies an ACK frame which is assumed to be reliable. The sender retransmits each data packet if no ACK is received until the retry limit m is reached. Given the parameters p

and m, we can calculate that the final residual error rate at the receiver is p^{m+1} and the average number of transmissions per packet is $\left(1-p^{m+1}\right)/\left(1-p\right)$. In other words, the redundancy transmission for one packet is $\left(p-p^{m+1}\right)/\left(1-p\right)$.

For the other example, let us consider a single FEC scheme over the i.i.d channel model with a packet error probability p. The packet level FEC code is (n,k), which means $n-k$ parity packets are generated from k original data packets and the original k data packets can be decoded out from any k correct packets of the block of n packets. Given the parameters n, k and p, the probability that there is i erroneous packets in the block is shown in formula (3.1). And the block reception rate, which is the probability that the original k data packets can be decoded out, is calculated as in formula (3.2).

$$\Pr(i \text{ packets lost } | n \text{ packets block}) = \binom{n}{i} p^i (1-p)^{n-i} \tag{3.1}$$

$$\Pr(\leq n\text{-}k \text{ packets lost } | n \text{ packets block}) = \sum_{i=0}^{n-k}\left(\binom{n}{i} p^i (1-p)^{n-i}\right) \tag{3.2}$$

3.3. GE Channel Model

3.3.1. GE Channel Model

The GE channel model [Gil60] [Ell63] [Mus89] is a two-state Markov chain shown in Figure 3.3 which was first used by Gilbert [Gil60] to characterize error sequences generated by data transmission channels. In the Good state (G) errors occur with (low) probability P_G while in the Bad state (B) they occur with (high) probability P_B.

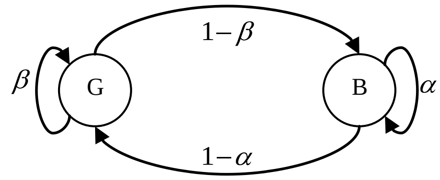

Figure 3.3: GE Channel Model

The errors occur in clusters or bursts with relatively long error-free intervals (gaps) between them. The state transition is summarized by its transition probability matrix in formula (3.3).

$$P = \begin{bmatrix} \beta & 1-\beta \\ 1-\alpha & \alpha \end{bmatrix} \tag{3.3}$$

This model can be used to generate sequences of symbol errors, in which case, it is common to set $P_G = 0$ and $P_B = 0.5$ [Yee95]. However, in situations where a code (such as Reed-Solomon code) is used (e.g. in the physical layer), it is more appropriate for the model to generate m-bit symbol errors, e.g. bytes level or packet level. In this case, the most reasonable choices for the two symbol error probabilities are $P_G \approx 0$ and $P_B \approx 1$ [Yee95]. This model is always referred to as the simplified GE model, which was proved to be accurate for modeling the burst packet losses in wireless LANs [Kar03] [Kha03] [Tan09]. The analyses and simulation experiments in the following chapters are based on the simplified GE model.

The steady state probabilities of being in states G and B are $\pi_G = (1-\alpha)/(2-\alpha-\beta)$ and $\pi_B = (1-\beta)/(2-\alpha-\beta)$ respectively. So the average packet loss rate produced by the GE channel model is

$$p = P_G \pi_G + P_B \pi_B = \frac{P_G(1-\alpha) + P_B(1-\beta)}{(1-\alpha+1-\beta)} \tag{3.4}$$

For the simplified GE channel model, the PLR will be

$$p = \frac{(1-\beta)}{(1-\alpha+1-\beta)} \tag{3.5}$$

Following [Yee95], the variance of the error symbol Z is $\sigma^2 = E(Z-\overline{p})^2 = p(1-p)$. So the correlation coefficient ρ of two consecutive error symbols Z_1 and Z_2 can be calculated as in formula (**3.6**3.6), which is referred to as the temporal error correlation.

$$\rho = \frac{E((Z_1-p)(Z_2-p))}{\sigma^2} = \frac{E(Z_1Z_2)-pE(Z_1)-pE(Z_2)+p^2}{p(1-p)}$$
$$= \frac{p\alpha-p^2}{p(1-p)} = \alpha+\beta-1 \tag{3.6}$$

Finally, the two parameters of the simplified model, α and β, can be expressed in terms of the more meaningful quantities p and ρ by solving formulas (3.5) and (3.6). This yields

$$\alpha = p+\rho(1-p) \tag{3.7}$$
$$\beta = (1-p)+\rho p \tag{3.8}$$

The transition probability matrix then becomes

$$P = \begin{bmatrix} 1-p(1-\rho) & p(1-\rho) \\ (1-p)(1-\rho) & 1-(1-p)(1-\rho) \end{bmatrix} \tag{3.9}$$

And the I -step transition probability matrix is:

$$P^I = \begin{bmatrix} 1-p(1-\rho^I) & p(1-\rho^I) \\ (1-p)(1-\rho^I) & 1-(1-p)(1-\rho^I) \end{bmatrix} \tag{3.10}$$

Furthermore, the error-burst-length X and the error-free-length Y can be defined as random variables [Tri82], shown in formula (3.11) and (3.12), which are the number of all potential transition sequences that contain exactly $j-1$ times one state and 1 time the other state.

$$p_X^j = \Pr(X = j) = \alpha^{j-1}(1-\alpha), \qquad j \in \{1,2,\dots\} \tag{3.11}$$

$$p_Y^j = \Pr(Y = j) = \beta^{j-1}(1-\beta), \qquad j \in \{1,2,\dots\} \tag{3.12}$$

Finally, the expected values of the error-burst-length X and the error-free-length Y are:

$$E(X) = \sum_{j=1}^{\infty} j \cdot p_X^j \overset{\alpha<1}{=} \frac{1}{1-\alpha} \tag{3.13}$$

$$E(Y) = \sum_{j=1}^{\infty} j \cdot p_Y^j \overset{\beta<1}{=} \frac{1}{1-\beta} \tag{3.14}$$

3.3.2. Sequence Analysis

We first analyze the correlation coefficient ρ for the simplified GE model. From formula (3.6), we can see that the normalized ρ can take all values between $[-1,1]$ since α and β are conditional probabilities that can only take values between $[0,1]$. Let's have a look into some special cases [FMI09].

- Assume no correlation (e.g. $\alpha+\beta=1$):

  In this case the GE channel acts as a memoryless channel for which the error probability follows eq. (3.5). This special case does not have any correlation between successive error events. Those error events are then i.i.d.

- Assume extreme positive correlation ($\rho=1$):

In this special case, both α and β equal to 1. There are two possibilities: a sequence of all Good states ($p=0$) or a sequence of all Bad states ($p=1$). The successive error events are extremely correlated.

- Assume extreme negative correlation ($\rho=-1$):

  This case means both α and β equal to 0. The successive error events are also extremely correlated but it is in a negative way. This case is a sequence of Good and Bad states where each state is followed by its opposite state, such as GBGBGB… or BGBGBG..., and the average error rate is 0.5.

To further explain the correlation coefficient, Figure 3.4 shows the *i*-step transition probability for a state trellis starting in the "B" state and ending in "B" state for a packet loss probability of 50% and various correlations between -1..1.

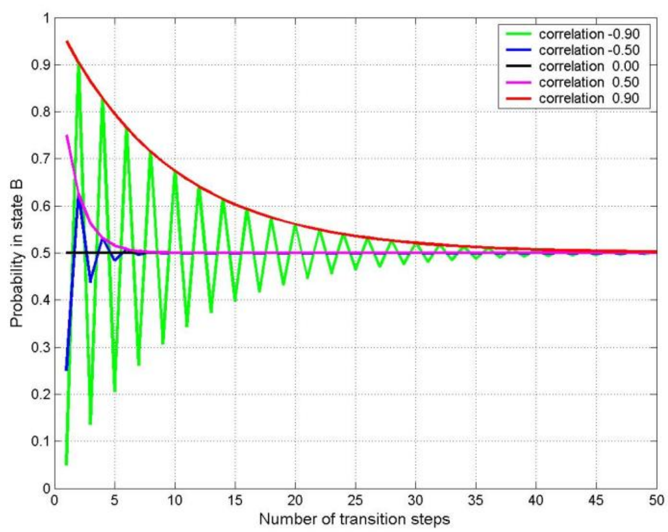

Figure 3.4: Example transition probabilities

It is shown in [Tan07b] [Gor07] that the correlation coefficient is about 0.01~0.05 in normal environments for wireless LANs.

Now we compute $P[a,b]$, the probability of a errors in a sequence of b symbols following [Yee95]. Let $P_G[a,b]$ be the probability of a errors in b transmissions with the channel ending in state G. Similarly, let $P_B[a,b]$ be the probability of a errors in b transmissions with the channel ending in state B. Then

$$P[a,b] = P_G[a,b] + P_B[a,b] \tag{3.15}$$

For b=1, 2, 3 … and a=0, 1, 2 … b, assuming the simplified GE channel, then

$$P_G[a,b] = P_G[a,b-1]\beta + P_B[a,b-1](1-\alpha) \tag{3.16}$$

$$P_B[a,b] = P_B[a-1,b-1]\alpha + P_G[a-1,b-1](1-\beta) \tag{3.17}$$

The initial conditions for the recursion are

$$P_G[0,0] = (1-\alpha)/(2-\alpha-\beta) \tag{3.18}$$

$$P_B[0,0] = (1-\beta)/(2-\alpha-\beta) \tag{3.19}$$

and $P_G[a,0] = P_B[a,0] = 0$ for $a \neq 0$. Note that with these initial conditions, all numerical values computed by the recursion will be steady state results.

3.3.3. Discussion

As described in previous sections, the i.i.d channel model is a special case of the SGE channel model when there is no any correlation between successive error events. The SGE channel model was proved to be accurate for modeling the burst packet losses in wireless LANs [Kar03] [Kha03] [Tan09].

We also simulate a typical 802.11a wireless LAN and probe the error character of MAC broadcast. One AP broadcasts packets periodically at a load of 4.4Mbps. Meanwhile, each receiver transmits packets to a random station and the total load is about 1.6Mbps which are considered as background traffics. All nodes move randomly at a speed of 10m/s. The average PLR per 100ms for the MAC broadcast at each receiver is shown in Figure 3.5. From the results, we can see that the broadcast errors at each receiver burst occasionally, and hence they can be modeled by the SGE model.

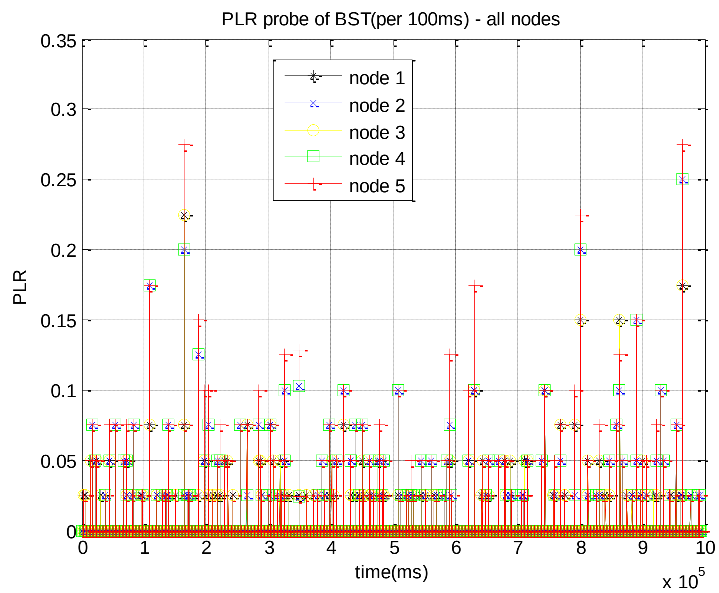

Figure 3.5: Average PLR per 100ms of MAC broadcast

The analyses and simulation experiments in the following chapters are based on the SGE model if not otherwise stated.

3.4. Delay Budget

As discussed in Chapter 1, the real-time multimedia applications always require strict delay constraints. Moreover, in this thesis, we will compare the MAC layer multicast error recovery approaches with application layer end-to-end ones. Consequently, we need to analyze the delay budgets for different error recovery schemes in different layers.

3.4.1. End-to-End Delay

We first consider the end-to-end delay bounds for ARQ, FEC and HEC over a general model: the underline RTT is bound by a value T_{RTT} and the delay for each direction is $T_{RTT}/2$. Each packet retransmission incorporates one RTT, i.e. the time it takes to transmit a feedback packet from the receiver to the sender plus the time for sending back a repetition of the lost packet. As for end-to-end ARQ, RTT is significantly larger than twice the pure delay of physical communications. Here, RTT also includes the time that the host requires for processing feedback and retransmission. Another parameter we have to consider is the packet interval (or called load interval) T_s, which is derived from the average data rate (or called load rate) $\bar{R}$ and the average packet length $\bar{L}$: $T_s = \bar{L}/\bar{R}$. The packet transmission time (packet-length/channel-capacity) is T_d . According to the information theory, in order to guarantee the reliability of transports, the system must satisfy $T_d < T_s$. Otherwise, it is impossible to guarantee the reliability of transmissions due to insufficient bandwidth which leads to congestions in the connection. Now we consider the delay bound of a simple Stop-and-Wait ARQ scheme which is shown in Figure 3.6. Given the retry limit m, the upper bound of the delay can be calculated in formula (3.20).

$$T_{ARQ} \le T_s + mT_{RTT} + T_{RTT}/2 \tag{3.20}$$

In the packet-level FEC the size of the interleaver represents the coding delay: It is the amount of data that has to be collected at the receiver until it is able to recover lost packets [FMI09]. Due to virtual interleaving, the delay can be assumed to only appear at the receiver. It is assumed that the redundant packets do not contribute to the delay. This is true as long as the capacity of the network link is not exceeded, i.e. the redundant packets just raise the date rate and therefore squeeze the packet interval. For simplification, we assume that the effect to the overall FEC delay vanishes. The transmission of FEC block, with code

(n,k), is shown in Figure 3.7. From this figure, the block delay of FEC scheme can be calculated as in formula (3.21).

$$T_{FEC} = kT_s + T_{RTT}/2 \tag{3.21}$$

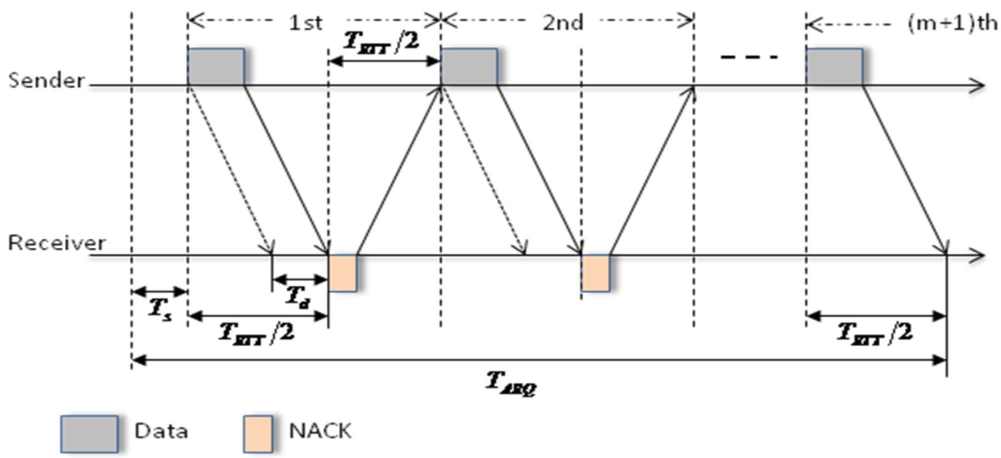

Figure 3.6: End-to-End Delay of Stop-and-Wait ARQ

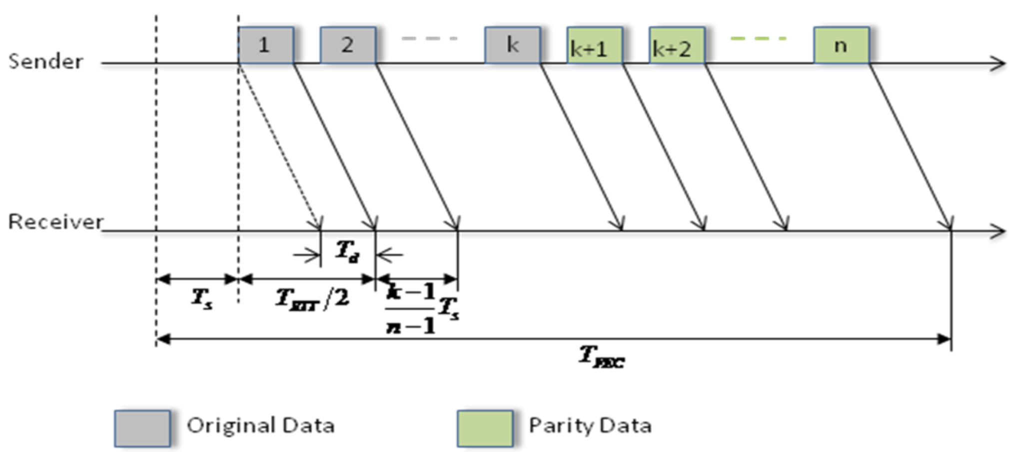

Figure 3.7: End-to-End Delay of FEC

We now consider a simple HEC scheme, which is a basic HARQ Type II scheme as introduced in chapter 2. As shown in Figure 3.8, it is not to send any parity packets (redundant packets) with the first transmission, but to send parity packets when a retransmission is required (e.g. NACK). As FEC and ARQ operate sequentially, the over HEC delay is calculated straight forward as the sum of their single delays. Combining the delay calculation for ARQ and FEC, given the retry limit m, the upper bound of the delay can be calculated in formula (3.22).

$$T_{HEC} \le kT_s + mT_{RTT} + T_{RTT}/2 \tag{3.22}$$

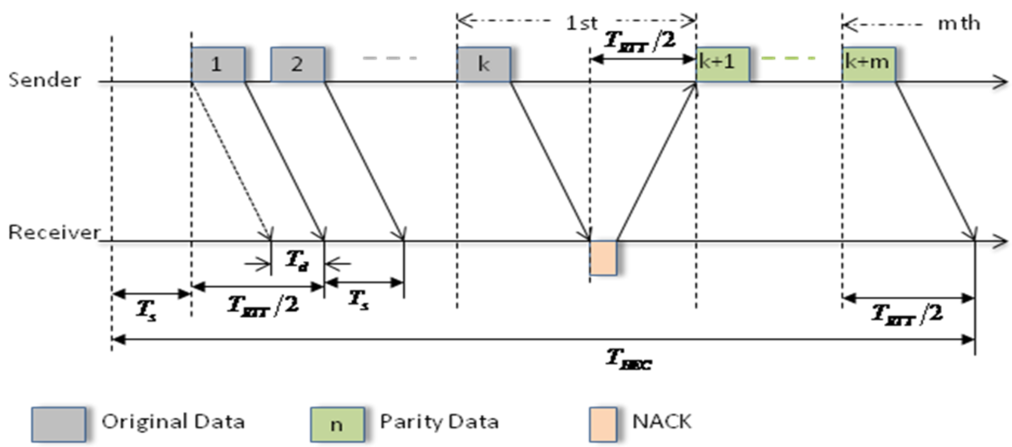

Figure 3.8: End-to-End Delay of HARQ Type II

These are foundational delay analyses for basic end-to-end ARQ, FEC and HEC schemes and will be used for the delay analyses of our proposed schemes in this thesis later on.

3.4.2. MAC Layer Delay

For the class of multicast applications having strict end-to-end delay requirements (e.g. multimedia conferencing), error recovery on an end-to-end basis is not a good option because it takes a long time and sometime may not meet the delay constraints. For example, TCP, which supports reliable end-to-end reliable transmissions, cannot guarantee the transmission delay and hence it is not suitable for real-time applications. However, link level or MAC level error recovery operates on a considerably smaller time scale, and is therefore a viable approach. To focus on the wireless links, we consider an end-to-end multicast scenario (shown in Figure 3.9) with loss-free wired networks and assume that losses take place only on wireless links. Using end-to-end error recovery approaches, the need for additional transmissions (e.g. retransmissions in ARQ related schemes) due to errors in the wireless links puts unnecessary processing burden on the original sender. These additional transmissions go over the entire wired multicast tree and also the wireless links, taking a long time, wasting bandwidth and also leading to processing of unwanted redundant retransmissions at those receivers which might have already received the packet. If the base-station or the AP were to take the responsibility of supplying retransmissions (e.g. MAC layer approaches) rather than the original sender, then the load of supplying retransmissions gets distributed across base-stations which are restricted only within the local area and taking a shorter time.

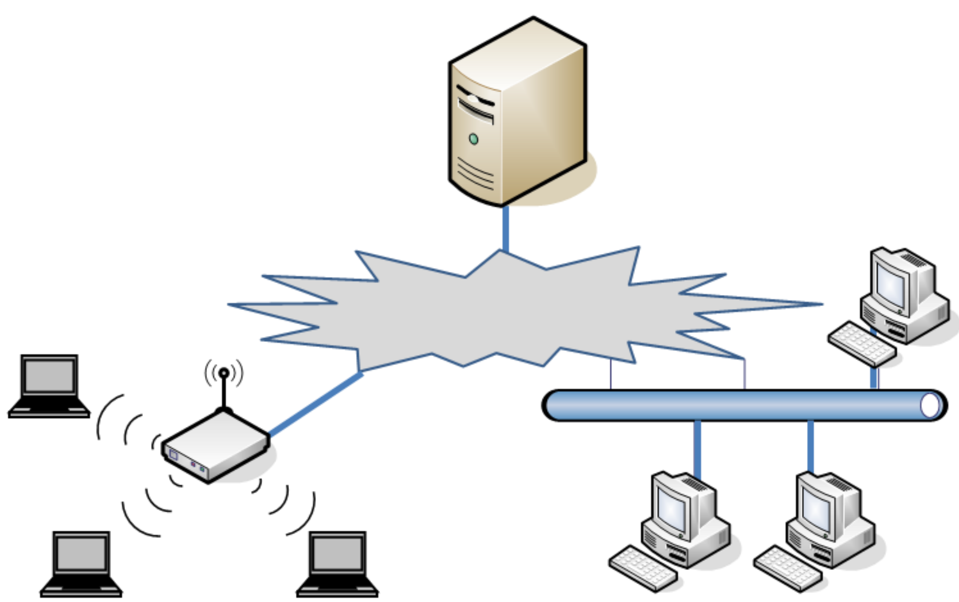

Figure 3.9: End-to-End multicast scenario

Moreover, as the link level exchange between the sender and receivers is in a smaller time scale with strict time synchronization, special approaches of multicast feedback can be performed and hence shorten the final end-to-end delay further. The ACK/NACK jamming based multicast feedback is one of the approaches, where the ACK and NACKs are aggregated in the same time slot of the legacy ACK. The delay of ACK/NACK jamming based MAC layer multicast approaches will be discussed in detail in the following chapters.

3.5. Conclusion

In this chapter, we discussed two packet erasure channel models to support the performance evaluation for the multicast error recovery approaches that will be discussed in the following chapters. The first channel model is the independent and identically distributed channel model, which is often used due to its simplicity, while the other one is the Gilbert-Elliott channel model, which is more accurate but of a higher analysis complexity. The performance analyses and simulations for the proposed schemes will be based on both of these channel models.

We also discuss the delay budgets of end-to-end multicast error recovery approaches: FEC, ARQ and HEC. Moreover, the delay budget of MAC layer approaches have been discussed and compared with end-to-end approaches.

Chapter 4

Feedback Jamming based Protocols

To design a feedback mechanism for MAC layer multicast, we first talk about feedback aggregation in this chapter. It is found that pure NACK aggregation has some fake decision problems while the feedback jamming mechanism, which allows ACK and NACKs in the same time slot, can mitigate the fake decision problems. As all NACK frames are aggregated in the same time slot as the ACK frame, the feedback jamming based protocols can achieve low latency and high throughput at predictable reliability, as well as superior scalability with respect to the number of receivers. However, the exploration of feedback jamming is still not thorough, neither the feasibility nor the protocol itself. In this thesis, we explore the feedback jamming thoroughly and develop enhanced protocols. We first describe our enhanced protocols, SEQ-LBP and HLBP, in this chapter and then explore the feasibility of feedback jamming and evaluate the performance of the protocols in the following chapters.

4.1. Feedback Aggregation

We first talk about the potential feedback mechanisms for MAC layer multicast. Based on the positive feedback in IEEE 802.11 DCF unicast, a direct multicast feedback is polling as described in chapter 2. However, the polling based mechanisms are not a perfect option as they suffer from the feedback implosion problem which is quite heavy for large multicast groups. Moreover, as discussed in Chapter 2, probabilistic approaches, such as "delayed feedback" and "probabilistic feedback" based ones, can also be used to handle with multicast feedback. However, they cannot avoid feedback collision completely and they still cause much longer delays than unicast. So here we focus on feedback aggregation. Unlike the positive feedback ACK, negative feedback NACKs could be aggregated in the same time slot because the collision of NACKs could be treated as a negative feedback as well. A potential mechanism based on pure NACK is shown in Figure 4.1 and explained as follows.

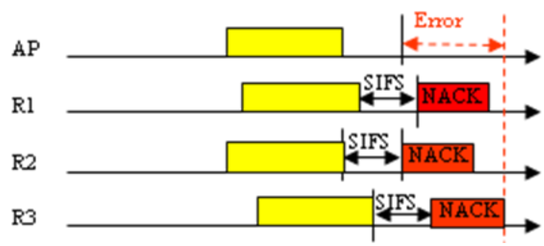

Figure 4.1: Pure NACK based multicast feedback

As shown in Figure 4.1, after receiving the data frame or the feedback request frame from the sender, each receiver replies a NACK if it has not received the corresponding data frame. Either a correctly received NACK or a detection of collision can prompt the sender to retransmit the data frame. Intuitively, this mechanism suffers from two main problems. First, if the data frame or the feedback request frame are completely destroyed (e.g. due to interference), receivers do not know any information about the sender, and hence they cannot reply NACK. As a result, the sender will detect a clean channel in the feedback time slot and treat it as a successful transmission. This fake positive feedback results in high residual error rates at receivers. The other problem is about fake collision detection. In particular, the interference from other networks may be detected by the sender as NACKs collisions which will lead to unnecessary retransmissions.

A potential approach to mitigate the problems of pure NACK aggregation is to allow a positive feedback ACK from a receiver delegate (or called leader receiver) in an additional time slot as shown in Figure 4.2. In this case, after receiving the data frame or the feedback request frame from the sender, the leader receiver replies an ACK frame if it has received the corresponding data frame correctly while each non-leader receiver replies a NACK frame if it has not received the data frame correctly yet. The sender retransmits the data frame when it has not received an ACK or has detected a collision during the ACK and NACK time slots respectively. The data transmission is accounted successful only when an ACK is received in the ACK time slot and no collision is detected in the NACK time slot. Apparently, this mechanism mitigates the problem of fake positive feedback. However, it still suffers from the fake collision detection. From the tests results in our test-bed, which will be shown in the following chapters, it is observed that the fake collision detection is severe and even causes 50 percents of unnecessary retransmissions.

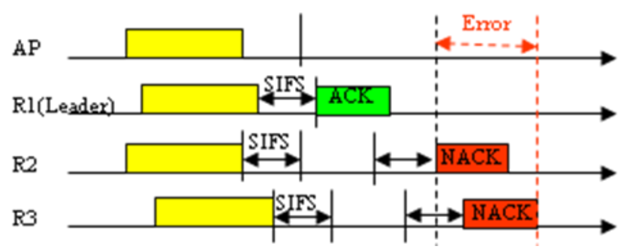

Figure 4.2: ACK/NACKs jamming in different time slots

To avoid the problem of fake collision detection, we try to aggregate the ACK and NACKs in the same time slot as shown in Figure 4.3. In this case, after receiving the data frame or the feedback request frame from the sender, the leader receiver replies an ACK frame if it has received the corresponding data frame correctly while each non-leader receiver replies a NACK frame in the same time slot if it has not received the data frame correctly yet. The NACK frames from non-leader receivers will destroy the ACK (called as feedback jamming), if any, sent from the leader receiver. The data transmission is accounted as successful when the sender receives an ACK frame correctly. Otherwise, the sender will retransmit the data frame. Intuitively, this mechanism avoids both fake positive feedback and fake negative feedback. However, due to the capture effect [Had02] [Li06] it is not sure that NACK frames can destroy the ACK frame completely in every case. The main task is to test the feedback jamming probability which will be done in the following chapters. In this chapter, we will present our proposed protocols based on feedback jamming.

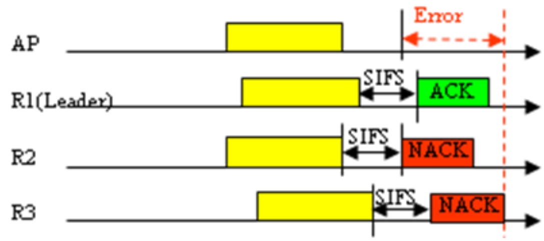

Figure 4.3: ACK/NACKs jamming in the same time slot

Due to the different distances from receivers to the sender, the time synchronization should be considered for feedback jamming based protocols. Fortunately, the feedback jamming scheme does not need strict time synchronization. The ACK frame can be destroyed when the NACK frames arrive within the receiving time of the ACK frame. In particular, the receiving time of an ACK frame (length 31bytes including the PLCP header) is: 31*8b/6Mbps. And the corresponding distance is 300Mmps*31*8b/6Mbps=12.4km. For an indoor wireless LAN environment, where the transmission distance is no more than 100m, the feedback jamming does not need any additional time synchronization. Moreover, due to the low requirement for time synchronization, the feedback jamming based multicast feedback mechanism could be used for other wireless networks with long transmission distance, e.g. the mobile communication networks.

4.2. LBP

4.2.1. Protocol

The MAC layer reliable multicast LBP, SEQ-LBP and HLBP require a slight modification to the legacy IEEE 802.11 protocols. LBP was proposed in [Kur99]. As shown in Figure 4.4, LBP works as follows. A receiver is selected as the leader for the multicast group. The AP first sends a RTS frame to all receivers, and only the leader receiver replies a CTS frame. The AP is then assured that the channel is granted and starts the transmission of the data frame. The leader receiver sends an ACK in reply if the data is received correctly, or does nothing otherwise. If any non-leader receiver detects a transmission error, a NACK is sent. This NACK frame will collide with the ACK, if any, sent by the leader receiver. The ACK/NACK jam is referred to as feedback jamming or ACK/NACK jamming in this thesis. And if the AP receives an ACK, this transmission is complete. Otherwise, the AP repeats the whole procedure until an ACK is received or the retry limit is reached.

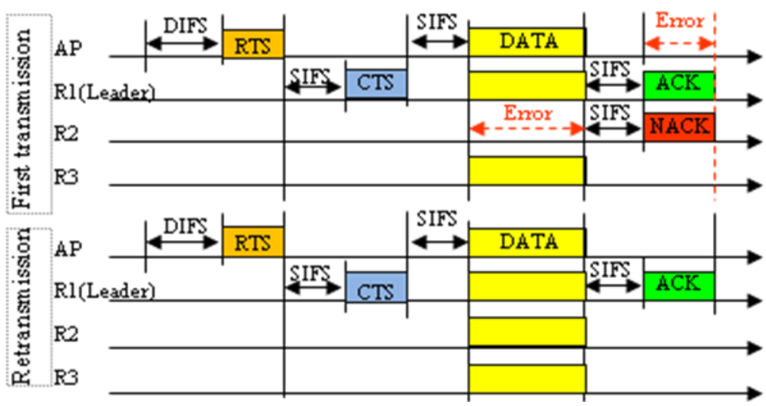

Figure 4.4: LBP

A simple leader election scheme was also proposed by Kuri et al. for LBP [Kur99]: When a receiver r sends a link-level join-group message to join multicast group G. The AP checks the table to find out whether or not group G already has a leader and replies with the message that r will be a non-leader or leader for group G respectively. When the leader sends a link-level leave-group message to leave group G or leaves without any announcement, the AP stops forwarding messages addressed to group G. If there are other group members that are still interested in G, they will eventually time out and start the process of subscribing to group G anew.

It was noted that it is possible to reduce the amount of control traffic flow for leader election purposes when a higher layer group management protocol like the IGMP [Cai02] is running above the link layer [Kur99]. In this case, explicit link-level join-group messages may be suppressed, and leader election carried out by “snooping” IGMP packets. Under IGMP, receivers send explicit IGMP-level join-group messages upon joining a group. These join-group messages must pass through the AP. Hence, it is possible for the AP to become aware of one or more group members in the cell. The AP can then assign one of these members the task of a leader by sending a message to this member.

4.2.2. Discussion

Intuitively LBP has two main problems. First, when the data frame is lost or fails the CRC check, the non-leader receivers cannot reply NACKs because they do not know when or how to send them, as the data frame cannot be trusted (even only with a bit error) and the destination is unknown. As a result, LBP is not reliable for the non-leader receivers and in practice application layer multicast error control schemes have to be used to correct the high residual errors in LBP. Second, LBP has poor performance when the channel error rates are high. The non-leader receivers send NACKs whenever the received frame is in error, regardless of whether this erroneous frame has been received correctly before or not. This is because the receivers in LBP cannot access the sequence number of the data frame before the data frame is received, as there is no such field in the structure of RTS/CTS frames for multicast or even the RTS/CTS has not been turned on. So the AP has to retransmit data packets until all receivers receive the data frame correctly at the same time. There are a lot of unnecessary transmissions, and hence LBP is not efficient particularly for lossy channels.

Compared with polling based schemes (e.g. block-ACK polling and IEEE 802.11 PCF), ACK/NACK jamming requires less strict synchronization because the only requirement for feedback jamming is that NACK frames come during the ACK time to collide with the ACK frame. Moreover, based on ACK/NACK jamming, many feedback based automatic rate adaptation schemes and channel prediction schemes for unicast can apply to multicast scenarios, such as Automatic Rate Fallback (ARF) [Kam97] etc.

However, as described in the previous sections, due to the capture effect, the feedback jamming probability is not always 100 percent and will only get to a certain value in a given scenario. Intuitively, the selection of the leader has great impact on the total feedback jamming probability. As a result the design of leader selection algorithm has to be considered with increasing the feedback jamming probability. We will develop a dynamic leader selection algorithm and evaluate its performance on our test-bed which is built using consumer wireless LAN cards.

4.3. SEQ-LBP

4.3.1. Protocol

SEQ-LBP enhances LBP with a MAC layer control frame called SEQ shown in Figure 4.5. Besides the same fields in RTS/CTS frames, such as frame control header, transmission duration, receiver address (RA), transmitter address (TA) and frame check sequence (FCS), the SEQ frame also includes the sequence number of the following data frame. The use of the SEQ frame is to lead receivers to set timers and to announce the sequence number of the following data frame. The SEQ frame also reserves the channel as RTS frame does, particularly when RTS/CTS exchange is not triggered for small size multicast data packets.

2	2	6	6	2	4
Frame Control	Duration	RA	TA	Sequence Control	FCS

Figure 4.5: Format of the SEQ frame in SEQ-LBP

SEQ-LBP is shown in Figure 4.6. The RTS/CTS exchange between the sender and the leader receiver is still optional like in LBP and is omitted here for simplicity. Unlike in LBP, the AP broadcasts a SEQ frame before the data frame. On receipt of the SEQ frame, each receiver sets a timer according to the SEQ frame. After receiving the data frame, the leader receiver replies an ACK frame if the data is correct or it has already got the data based on sequence check, or does nothing otherwise. When the timer expires, each non-leader receiver replies a NACK if the data is erroneous and it has not received it correctly yet based on sequence check, or does nothing otherwise. If no ACK is received, the AP repeats the whole procedure and retransmits the data until the retry limit is reached. If instead the AP receives an ACK, the transmission of this data packet is taken as successfully completed. For example, in the retransmission phase in Figure 4.6, although this time the data frame is lost, the leader receiver still replies an ACK because it knows that this data frame has been received correctly already in the first transmission, thanks to the SEQ frame.

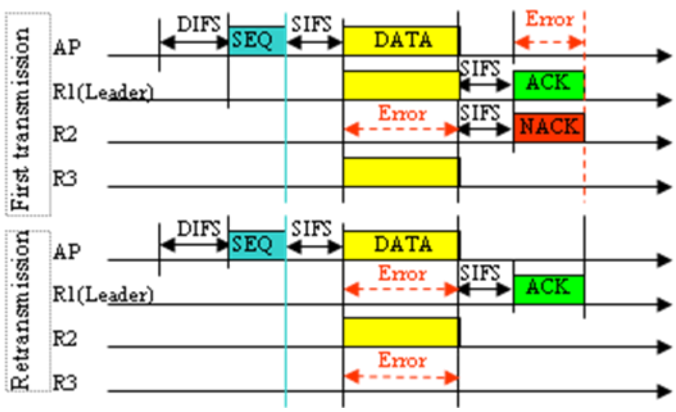

Figure 4.6: SEQ-LBP (First Version)

Furthermore, if the channel reservation function of the SEQ frame can be omitted, SEQ-LBP can work in a different way as shown in Figure 4.7, where the SEQ frame is sent after the data frame. The AP broadcasts a SEQ frame after the data frame to request feedback from all receivers. The leader receiver and non-leader receivers reply ACK and NACKs respectively as described above. Please note that, the first version is a better choice as the SEQ frame can reserve the channel and relieve collision or interference errors. As discussed in the first section in this chapter, the feedback jamming scheme has low requirements for time synchronization. For wireless LANs, the first version works well. For other networks with long distance receivers, the alternative version, in which the ACK/NACK frames are sent following the SEQ frame (after a SIFS), is a better choice. As the location of the SEQ frame has no impact on the feedback jamming probability, in this thesis we refer to SEQ-LBP as the first version if not otherwise stated.

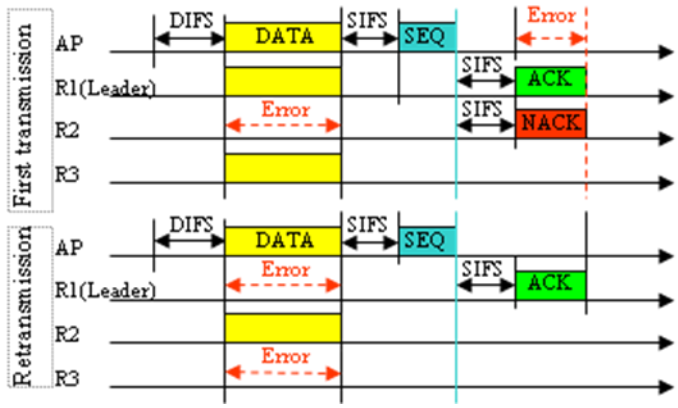

Figure 4.7: SEQ-LBP (Alternative Version)

4.3.2. Discussion

Intuitively, SEQ-LBP solves the problems of LBP very well. All the non-leader receivers can send feedbacks when the timers expire which are set based on the SEQ frame. Both the leader receiver and non-leader receivers send ACK and NACK respectively based on sequence check thanks to the SEQ frame, hence it avoids the unnecessary transmissions in LBP. SEQ-LBP is more efficient than LBP and has a higher scalability. As using the same DATA format, SEQ-LBP is compatible to legacy IEEE 802.11 protocols, where the legacy stations can even share the retransmissions for multicast members although they cannot join the multicast group by themselves.

However, SEQ-LBP is still a pure ARQ scheme and the error recovery is based on retransmission for each single data packet. As a result, SEQ-LBP is still not efficient for large multicast groups due to the limitation to scale. For example, for a multicast group with 10 receivers and the average independent error rate 0.10 for each receiver, roughly SEQ-LBP needs at least one retransmission for each data packet. Block feedback and FEC coding are good enhancements to increase the scalability and efficiency, which will be discussed in the following sections. Due to its simplicity and effectiveness, SEQ-LBP is still a good option for wireless LANs with small multicast groups.

4.4. HLBP

4.4.1. Protocol

To overcome the scalability limitation of pure ARQ schemes, we combine SEQ-LBP and packet level FEC and develop HLBP. The format of the SEQ frame in HLBP is shown in Figure 4.8, which is similar to the SEQ frame in SEQ-LBP. Instead of the sequence control field of the SEQ frame in SEQ-LBP, the SEQ frame in HLBP includes the block number and the packet index (in the block) of the following data frame. The use of the SEQ frame is to lead the non-leader receivers to set timers and to announce the block number and the packet index of the following data frame. The format of the data frame in HLBP is shown in Figure 4.9, which has a block number field and a packet index field instead of the sequence number field in the original data frame in 802.11 DCF.

2	2	6	6	1	1	4
Frame Control	Duration	RA	TA	Block Number	Packet Index	FCS

Figure 4.8: Format of the SEQ frame in HLBP

2	2	6	6	6	1	1	0-2308	4
Frame Control	Duration	RA	TA	BSSID	Block Number	Packet Index	Data	FCS

Figure 4.9: Format of the DATA frame in HLBP

As shown in Figure 4.10, HLBP uses a packet level FEC code (n,k) in the MAC layer. $n-k$ parity packets are generated from k original data packets. Similar to the MAC layer block transmission, the AP transmits the first $k-1$ data packets without feedback request and then transmits the k th data packet using an improved SEQ-LBP as follows: after transmitting the k th data frame, the AP broadcasts a SEQ frame to request feedback from all receivers. The leader receiver replies an ACK frame if it has already got at least k correct packets for the current block, or does nothing otherwise. Each non-leader receiver replies a NACK if it has got less than k correct packets for the current block, or does nothing otherwise. Then if the AP receives an ACK, this transmission is complete. Otherwise, the AP repeats the whole SEQ-LBP procedure and retransmits different parity packets until an ACK is received or the retry limit is reached.

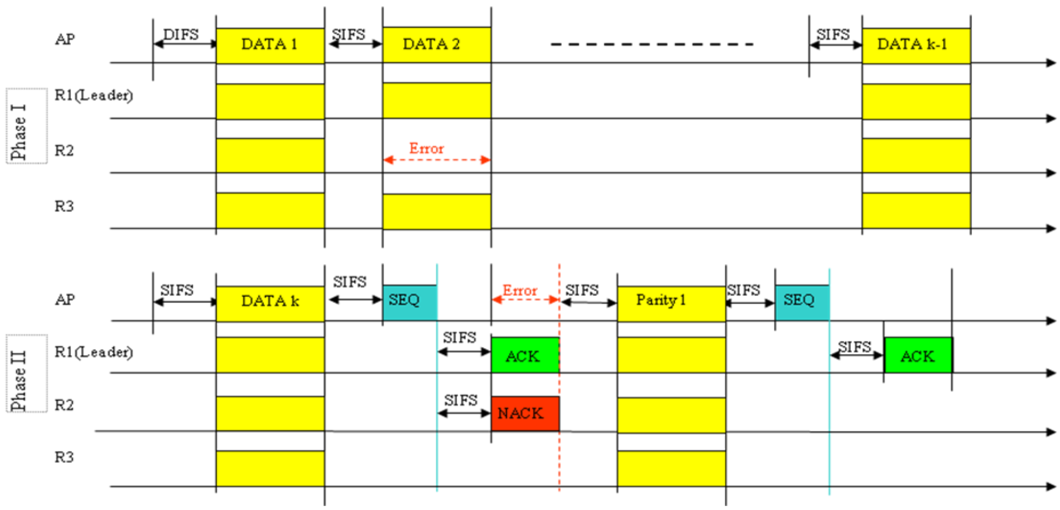

Figure 4.10: HLBP

Please note that here the alternative version of SEQ-LBP is used for HLBP, in which the ACK/NACK frames are sent following the last data frame after a SIFS but the channel reservation function of the SEQ frame is lost. The SEQ frame can also be scheduled before the last data frame in HLBP, which requires a longer timer but performs the channel reservation function of the SEQ frame. As the channel is already reserved for a block of packets, it is not needed for the SEQ frame to reserve the channel. As a result, in this thesis, we just talk about the current version as depicted above (Figure 4.10).

4.4.2. Discussion

We first note that SEQ-LBP is just a special case of HLBP when $k=1$. Both SEQ-LBP and HLBP achieve complete feedback suppression thanks to the feedback jamming scheme. However, the failure of feedback jamming and the loss of SEQ frames will decrease the performance (reception rate) at the non-leader receivers in both SEQ-LBP and HLBP. Fortunately, the SEQ frames are much more reliable (nearly error free) than data frames because they are much smaller and are transmitted using a lower data rate, like other control frames in 802.11 (e.g. RTS, CTS, ACK). Moreover, if RTS/CTS signaling is turned on, the SEQ frames also avoid collision losses. We will explore the feedback jamming probability under various scenarios in the following chapters.

Similar to SEQ-LBP, HLBP is also compatible to legacy IEEE 802.11 protocols. The data frames in HLBP can be interpreted by the legacy stations that do not use HLBP because the combination of the block number and packet index fields just equals the original sequence control field. And the one octet fields for block number (0-255) and packet index (0-255) are just big enough for RS code in wireless LANs, whose block length is typically no more than 255. Moreover, LBP, SEQ-LBP and HLBP can run without RTS/CTS exchanges for small data frames just like IEEE 802.11 DCF unicast. Although our discussion is in the context of IEEE 802.11 DCF, LBP, SEQ-LBP and HLBP are actually applicable to all ACK/retransmission based MAC protocols, such as 802.11 PCF (Point Coordination Function) etc.

About the time overhead of the FEC encoding and decoding in the MAC layer, actually all the parity packets can be generated out before the start of the whole transmitting and so it can satisfy the strict time constraint in the MAC layer. In practice, the FEC encoding and decoding can even be performed in the driver level (software level), in which way the FEC function will not put any burden in the wireless LANs hardware. Moreover, the parity packets can also be generated in upper layers (e.g. the application layer), which leads to cross-layer approaches. In this thesis, it is assumed that the FEC coding is a fundamental configure and we do not talk about its implementations in detail.

4.5. Conclusion

In this chapter, we first talked about the potential feedback mechanisms for MAC layer multicast, in particular various feedback aggregation schemes. It is found that pure NACK aggregation has some fake decision problems while the feedback jamming mechanism, which allows ACK and NACKs in the same time slot, can mitigate the fake decision problems.

We then described the LBP, SEQ-LBP and HLBP protocols. The improvement of SEQ-LBP and HLBP to LBP is discussed. In summary, LBP is not a mature protocol due to several severe problems. SEQ frame is used to solve the problems and to improve LBP. Packet level FEC coding is introduced to the MAC layer to improve the efficiency for large multicast groups. Because of simplicity and effectiveness, SEQ-LBP is a good option for wireless LANs with small multicast groups while HLBP is a better option for large multicast groups. In the following chapters, we will explore the feasibility of feedback jamming and evaluate the performances of the proposed protocols through theoretical analyses, NS-2 simulations and experimental tests on a real test-bed built using consumer wireless LANs cards.

Chapter 5

Performance Analysis

As discussed in the last chapter, due to the capture effect, the feedback jamming probability is not always 100 percents and will only get to a certain value in a given scenario. Moreover, the channel condition of the leader receiver has great impact on the total feedback jamming probability. In this chapter we calculate the feedback jamming probability over the Rayleigh channel model, which is commonly used to model wireless LAN scenarios [Rap96] [Stu02].

Given a feedback jamming probability, then we analyze the performance of LBP, SEQ-LBP and HLBP over both the i.i.d channel model and the simplified GE channel model. The used metrics include the final residual error rate (or PLR), the average redundancy transmission and the average channel holding time [Kur99], where the last one is a natural criterion because the reciprocal of the average channel holding time provides a measure of throughput. The channel holding time is obtained by summing up the time, to access the channel and to actually transmit data or feedback, associated with successful transmission of the tagged data packet to all group receivers. Moreover, we also calculate the upper bounds of the multicast delay in those protocols.

5.1. Feedback Jamming Probability

5.1.1. Performance Analysis

We first analyze the ACK/NACK jamming performance theoretically. The used propagation model takes the deterministic power loss and multipath fast fading of signal into account. For the purpose of analytical tractability, it is assumed that there is no direct path between the AP and the receiver, in other words the envelope of transmitted signal is Rayleigh-faded [Rap96]. Therefore, its instantaneous power is exponentially distributed according to:

$$f_{p_i}(p) = \frac{1}{p_{0i}} \exp\left(-\frac{p}{p_{0i}}\right), p > 0 \tag{5.1}$$

where p_{0i} represents the local-mean power of the transmitted frame at the receiver and can be calculated using the typical log-distance path loss model [Rap96], shown in formula (5.2). This model gives the path loss P_l at a distance d from the transmitter based on the path loss at some close-in reference distance d_0.

$$P_l(d) = P_l(d_0) + n10\log_{10}(d/d_0) \tag{5.2}$$

where n, the path loss exponent, determines the rate of loss. A number of values for n have been proposed for different environments. We use $n = 3$, which is commonly used to model losses in an urban environment [Rap96]. To estimate $P_l(d_0)$, we use the Friis free space propagation model [Rap96] as follows:

$$P_r(d_0) = \frac{P_t G_t G_r \lambda^2}{(4\pi d_0)^2 L} \tag{5.3}$$

where P_r and P_t are the receiving and transmit powers (in Watts), G_t and G_r are the transmit and receive antenna gains, λ is the carrier wavelength (in meters), and L is a system loss factor ($L=1$ in our simulations).

Noise was modeled as a combination of the noise floor of the interface and the aggregate energy of neighboring transmissions that were too weak to cause a collision. The noise floor was computed by first calculating the thermal noise N_t using the well known equation [Rap96]:

$$N_t = kTB_t \tag{5.4}$$

where k is Boltzmann's constant (1.38e23 Joules/Kelvin), T is the temperature (in Kelvin), and B_t is the unspread bandwidth of the interface; and then factoring in the published noise figure of the interface.

During simultaneous transmissions of multiple stations, a receiver captures a frame if the power of the detected frame sufficiently exceeds the joint power (incoherent addition) of the interfering contenders by a certain threshold factor for the duration of a certain time period (over which instantaneous power is assumed to remain approximately constant). Thus, the capture probability is the probability of signal-to-interference ratio γ exceeding the product capture threshold Th_{cap}. Similar to the calculation of the capture probability in [Kim99] [Had02] [Li06], the jamming probability of one ACK frame (denoted as l) and $R-1$ NACK frames can be calculated as in formula (5.5).

$$\begin{aligned} JP(Th_{cap}, R) &= 1 - Prob(\gamma > Th_{cap}) \\ &= 1 - \int_0^\infty dp_1 f_{p_1}(p_1) \ldots \int_0^\infty dp_{R-1} f_{p_{R-1}}(p_{R-1}) \int_{Th_{cap}(p_1+\ldots+p_{R-1}+N_0)}^\infty f_{p_l}(p_l) dp_l \end{aligned}$$

$$= 1 - \exp\left[-Th_{cap}\frac{N_0}{p_{0l}}\right]\prod_{i=1}^{R-1}\frac{1}{1+Th_{cap}\frac{p_{0i}}{p_{0l}}} \tag{5.5}$$

where p_l and p_i, $1 \le i \le R-1$ represent the receiving power of the ACK frame and NACK frames at the AP respectively. Similarly, p_{0l} and p_{0i}, $1 \le i \le R-1$ denote the local-mean receiving power of the ACK frame and NACK frames at the AP respectively.

5.2. LBP Performance

5.2.1. Over the i.i.d Channel Model

We first analyze the performance of LBP over both the i.i.d channel model and the SGE channel model. Please also note that here it is assumed that the data frames are only partially damaged and non-leader receivers can reply NACKs based on the data frames (feedback time and destination). As a result the performances of LBP are just upper bounds and might not be always hold in practice. However, this assumption has no impact on the performance comparison with other protocols as LBP has the worst performance.

Given the jamming probability JP of an ACK frame against a single NACK frame, we consider the performance of LBP under the assumption that the channel model is identical for each receiver, e.g. the same average channel loss rate, and error events at different receivers are independent. It is also assumed that only the data frame experiences lossy channel and both ACK and NACK frames are reliable, which is a common assumption for protocol analysis [Tow97] [Kur99] [Tan09]. We first consider the performance of LBP over the i.i.d channel model given the average loss rate p, receiver number R and retry limit m. Firstly, the residual error rate for the leader receiver can be calculated as in formula (5.6).

$$PLR_{LBP}^{l}(i.i.d, p, m) = p^{m+1} \tag{5.6}$$

For the residual error rates at non-leader receivers, we first consider the scenario with only two receivers ($R = 2$). The residual error rate for the non-leader receiver can be calculated as in formula (5.7). The first term $p(1-p)(1-JP)$ (in the first line) denotes the probability that the non-leader receiver loses the packet while the leader receiver gets it correctly and the NACK from the non-leader fails to destroy the ACK from the leader, as a result the whole transmission is finished and the error cannot be recovered any more. The second term $\left(pp + p(1-p)JP\right)$ is the probability that both receivers lose the packet or the NACK from the non-leader receiver destroys the ACK from the leader successfully, both of which lead to the retransmission from the sender. Similarly, the following pairs of terms denote the probabilities in each retransmission round.

$$
\begin{aligned}
&PLR_{LBP}^{nl}\left(i.i.d, p, R=2, m, JP\right) = p(1-p)(1-JP) + \left(pp + p(1-p)JP\right)* \\
&\left(p(1-p)(1-JP) + \left(pp + p(1-p)JP\right)\left(...\left(p(1-p)(1-JP) + \left(pp + p(1-p)JP\right)p\right)\right)\right) \qquad (5.7) \\
&= p(1-p)(1-JP)\frac{1-Q_2{}^m}{1-Q_2} + pQ_2{}^m \\
&Q_2 = pp + p(1-p)JP
\end{aligned}
$$

For the case of more than two receivers, the probability of jamming caused by NACKs from other non-leader receivers (denoted as S) should also be included. Similar to formula (5.7), formula (5.8) shows the final calculation, where Q_R denotes the probability that both of the considered non-leader receiver and the leader receiver lose the packet or the NACK from the considered non-leader receiver destroys the ACK from the leader successfully, or the NACKs from other non-leader receivers destroys the ACK, any of which leads to the retransmission from the sender. We note that $PLR_{LBP}^{nl}(i.i.d, p, m, JP) = p^{m+1}$ when $JP = 1$.

$$
PLR_{LBP}^{nl}\left(i.i.d, p, R, m, JP\right) = p(1-p)(1-JP)\left(1-S\right)\frac{1-Q_R{}^m}{1-Q_R} + pQ_R{}^m \qquad (5.8)
$$

$$= p(1-p)(1-JP)\left(1-S\right)\frac{1-Q_R{}^m}{1-Q_R} + pQ_R{}^m$$

$$Q_R = pp + p(1-p)JP + p(1-p)(1-JP)S$$

$$S = \sum_{r=1}^{R-2}\left(\binom{R-2}{r} p^r (1-p)^{R-2-r}\left(1-(1-JP)^r\right)\right)$$

Then we calculate the expected number of transmissions per packet in LBP. The calculation, shown in formula (5.9), is just a geometric progression and the common ratio is the probability (denoted as Q) that the leader loses the packet or the NACKs from non-leader receivers destroys the ACK frame successfully. Please note that there are a lot of unnecessary retransmissions due to unnecessary NACK frames as there are no sequence announcements other than the data frames themselves. Then we get the average redundant transmission, shown in formula (5.10), which equals the expected number of transmissions minus one.

$$Expt_{LBP}\left(i.i.d, p, R, m, JP\right) = \frac{1-Q^{m+1}}{1-Q} \tag{5.9}$$

$$Q = p + (1-p)\sum_{r=1}^{R-1}\left(\binom{R-1}{r} p^r (1-p)^{R-1-r}\left(1-(1-JP)^r\right)\right)$$

$$RI_{LBP}\left(i.i.d, p, R, m, JP\right) = Expt_{LBP}\left(i.i.d, p, R, m, JP\right) - 1 \tag{5.10}$$

Finally, the expected channel holding time ET_{LBP} for a packet with a length of L bytes in LBP can be calculated in formula (5.11),

$$ET_{LBP} = T_{LBP}(L)Expt_{LBP} \tag{5.11}$$

where $T_{LBP}(L)$ = T_{RTS} + T_{CTS} + $T_{DATA}(L)$ + T_{ACK} + T_{DIFS} + $3T_{SIFS}$ + $4T_{PLCP}$ is the channel holding time of one transmission in LBP (following IEEE 802.11 DCF specification [IEEE07]). T_{RTS}, T_{CTS}, $T_{DATA}(L)$ and T_{ACK} are the transmission time of frames RTS, CTS, DATA (L bytes) and ACK respectively. T_{DIFS} denotes the Distributed Inter Frame Space time and T_{SIFS} is the Short Inter Frame Space time. T_{PLCP} denotes the transmission time of the PLCP (Physical Layer Convergence Protocol) preamble and header.

5.2.2. Over the SGE Model

Now we consider the performance of LBP over the SGE model under a similar assumption that the channel model is identical for each receiver, error events at different receiver are independent and the ACK/NACK frames are reliable. Given the jamming probability JP of an ACK frame against a single NACK frame, receiver number R, retry limit m and SGE model parameters (p , α , β), we first calculate the residual error rate at the leader receiver, as shown in formula (5.12), which is a sequence probability of the SGE model as introduced in Chapter 3.

$$PLR_{LBP}^{l}(SGE, p, \alpha, \beta, m) = p\alpha^{m} \tag{5.12}$$

As shown in formula (5.13), the calculation of the residual error rate at non-leader receivers is similar to the one over the i.i.d channel model. Note that the packet loss rates in the first transmission and retransmissions are different due to the feature of the SGE model. Moreover, as an approximation, the stable probability of SGE is used for the jamming caused by NACKs from the other non-leader receivers. In other words, the probability of jamming caused by NACKs from the other non-leader receivers (denoted as S) is the same as the one used in formula (5.8) based on the i.i.d channel model. Simulation results (will be presented in the next chapter) confirm that this approximation is quite close to the real results. Please note that $PLR_{LBP}^{nl}(SGE, p, \alpha, \beta, m, JP) = p\alpha^{m}$ when $JP = 1$.

$$PLR_{LBP}^{nl}(SGE, p, \alpha, \beta, R, m, JP) = p(1-p)(1-JP)\left(1-S\right)$$

$$+\left(pp+p(1-p)JP+p(1-p)(1-JP)S\right)\left(\left(\alpha(1-p)(1-JP)\right)\frac{1-Q^{m-1}}{1-Q}\left(1-S\right)+Q^{m-1}\alpha\right) \quad (5.13)$$

$$Q=\alpha p+\alpha(1-p)Pj+\alpha(1-p)(1-Pj)S$$

$$S=\sum_{r=1}^{R-2}\left(\binom{R-2}{r}p^r(1-p)^{R-2-r}\left(1-(1-JP)^r\right)\right)$$

The calculation of the expected number of transmissions turns to be very complicated and we just give an approximation formula here, shown in formula (5.14). The calculation is the sum of triggering probability of each transmission round. The term for the second round is the sum of probabilities that the leader receiver loses the packet or the NACKs from non-leader receivers destroy the ACK successfully in the first round. The third round triggering probability is denoted by S, whose first sum is the probability that the leader loses the packet in the second round while the second sum denotes the jamming probability between NACKs from non-leaders and the ACK from the leader receiver in the second round. Note that the feedback jamming probability in the second round is calculated approximately based on that half of the receivers lose the packet on average. The calculation of the following transmission rounds are just approximations based on the third round using the same geometric progression as in the calculation of LBP over the i.i.d channel model. Simulation results confirm that this approximation is quite close to the real results.

$$Expt_{LBP}\left(SGE,p,\alpha,\beta,R,m,JP\right)\cong$$

$$1+\sum_{r=0}^{R-1}\left(\left(p+(1-p)\left(1-(1-JP)^r\right)\right)\left(\binom{R-1}{r}p^r(1-p)^{R-1-r}\right)\right)+S\,\frac{1-Q^{m-1}}{1-Q} \quad (5.14)$$

$$S=\sum_{r=0}^{R-1}\left(\left(\binom{R-1}{r}p^r(1-p)^{R-1-r}\right)\left(p\alpha+(1-p)\left(1-(1-JP)^r\right)(1-\beta)\right)\right)$$

$$+\sum_{r=0}^{R-1}\left(\begin{array}{l}\left(\binom{R-1}{r}p^{r}(1-p)^{R-1-r}\right)\\ *\Big(\big(p(1-\alpha)+(1-p)\big(1-(1-JP)^{r}\big)\beta\big)\big(1-(1-\alpha)^{r}\beta^{R-1-r}\big)\big(1-(1-JP)^{\lceil r/2\rceil}\big)\Big)\end{array}\right)$$

$$Q=p+(1-p)\sum_{r=1}^{R-1}\left(\binom{R-1}{r}p^{r}(1-p)^{R-1-r}\left(1-(1-JP)^{r}\right)\right)$$

Consequently we get the average redundant transmission, shown in formula (5.15), which equals the expected number of transmissions minus one. And the expected channel holding time ET_{LBP} for a packet with a length of L bytes in LBP can be calculated in formula (5.11).

$$RI_{LBP}\left(SGE,p,\alpha,\beta,R,m,JP\right)=Expt_{LBP}\left(SGE,p,\alpha,\beta,R,m,JP\right)-1 \tag{5.15}$$

At last, we discuss about the delay bound of the LBP protocol. The maximum delay can be calculated as in formula (5.16):

$$D_{LBP}\le(m+1)\left(T_{cc}+T_{LBP}(L)\right) \tag{5.16}$$

where T_{cc} denotes the channel contention time which can be obtained by measurements or calculated theoretically following [Bia00]. Please note that the choice of the retry limit m can follow this formula in practice especially for applications requiring strict delay constraints.

5.3. SEQ-LBP Performance

5.3.1. Over the i.i.d Channel Model

In this section, we analyze the performance of SEQ-LBP under the same assumption as for LBP: the channel model is identical for each receiver, error events at different receivers are independent and the ACK/NACK frames are reliable. The SEQ frame is assumed to be reliable also as ACK/NACK frames. Please note that here our calculation is also applicable to the cases that the SEQ frame experiences losses because the SEQ losses can be handled as a failure of ACK/NACK jamming as they have the same impact on non-leader receivers: the NACK information fails to reach the sender. We first consider the performance of SEQ-LBP over the i.i.d channel model. Similar to the calculation for LBP, the residual error rate at the leader receiver in SEQ-LBP is shown in formula (5.17).

$$PLR^{l}_{SEQ-LBP}(i.i.d, p, m) = p^{m+1} \tag{5.17}$$

The final PLR of non-leader receivers can be calculated in formula (5.18). The first term is the probability that this non-leader receiver loses all the packets in all the $m+1$ rounds while the leader receiver loses all the packets in the first m rounds. The rest sum denotes the residual error rates when the leader receiver gets the packet correctly in each round. The three terms of the sum are the residual error rates in the first (round i), middle and last rounds respectively. Here the jamming probability (by all NACKs) from round i to m is calculated approximately by a product whose terms (T_i) are the average jamming probabilities of each round. Simulation results confirm that this approximation is quite close to the real results, which will be shown in the next chapter. Please also note that $PLR^{nl}_{SEQ-LBP}(i.i.d, p, R, m, JP) = p^{m+1}$ when $JP = 1$.

$$PLR^{nl}_{SEQ-LBP}(i.i.d, p, R, m, JP) \cong p^{2m+1} +$$

$$\sum_{i=1}^{m} p^{2i-1}(1-p)\left((1-JP)(1-S_i) + \sum_{j=i+1}^{m}\left((1-JP)(1-S_j)\prod_{l=i}^{j-1} T_l\right) + \prod_{l=i}^{m} T_l\right) \tag{5.18}$$

$$T_i = p\left(JP + (1-JP)S_i\right)$$

$$S_i = \sum_{j=1}^{R-2}\left(\binom{R-2}{j} p^{ij}(1-p^i)^{R-2-j}\left(1-(1-JP)^j\right)\right)$$

The expected number of transmissions for one packet in SEQ-LBP over the i.i.d channel model is calculated in formula (5.19). The calculation is the sum of the probability that each transmission round is triggered. In detail, the first term of the sum is the probability that the leader receiver loses the packet in this round and triggers a retransmission from the sender definitely. The second term is the probability that the leader receiver gets the packet correctly but the NACKs from the non-leaders successfully destroy the ACK from the leader in the following rounds, where the jamming probability is calculated approximately based on the current round only. Simulation results confirm that this approximation is quite close to the real results.

$$Expt_{SEQ-LBP}(i.i.d, p, R, m, JP) \cong 1+\sum_{i=1}^{m}\left(p^{i} + p^{i-1}(1-p)\sum_{j=i}^{m} S_{j} \right) \quad (5.19)$$

$$S_i = \sum_{j=1}^{R-1}\left(\binom{R-1}{j} p^{ij}(1-p^{i})^{R-1-j}\left(1-(1-Pj)^{j}\right)\right)$$

Then we get the average redundant transmission, shown in formula (5.20), which equals the expected number of transmissions minus one. Please also note that, when $JP=1$, the average redundant transmission per data packet in SEQ-LBP can be calculated as $RI_{SEQ-LBP} = \sum_{i=1}^{m}\left(1-\left(1-p^{i}\right)^{R}\right)$ (can also be derived from formula 5.19) which is the limit of ARQ (with $m \to \infty$) or can be called as ARQ Shannon limit, e.g. RI=0.1111 when p =0.1 and R =1. This means SEQ-LBP can reach the ARQ limit as there is no unnecessary feedback and retransmissions due to the feedback jamming schemes. Smaller feedback jamming probabilities do not increase the redundant transmissions but cause high residual error rates.

$$RI_{SEQ-LBP}\left(i.i.d, p, R, m, JP\right) = Expt_{SEQ-LBP}\left(i.i.d, p, R, m, JP\right) - 1 \quad (5.20)$$

Finally, the expected channel holding time $ET_{SEQ-LBP}$ for a packet with a length of L bytes in SEQ-LBP can be calculated in formula (5.21),

$$ET_{SEQ-LBP} = T_{SEQ-LBP}(L)Expt_{SEQ-LBP} \tag{5.21}$$

where $T_{SEQ-LBP}(L) = T_{RTS} + T_{CTS} + T_{SEQ} + T_{DATA}(L) + T_{ACK} + T_{DIFS} + 4T_{SIFS} + 5T_{PLCP}$ is the channel holding time of one transmission in SEQ-LBP, and T_{SEQ} is the transmission time of the SEQ frame.

Similar to the calculation for LBP, the maximum delay in SEQ-LBP can be calculated as in formula (5.22), which can also be used to select the retry limit m.

$$D_{SEQ-LBP} \leq (m+1)\left(T_{cc} + T_{SEQ-LBP}(L)\right) \tag{5.22}$$

5.3.2. Over the SGE Model

Now we consider the performance of SEQ-LBP over the SGE model under the assumption that the channel model is identical for each receiver, error events at different receiver are independent and the control frames (SEQ, ACK and NACK) are reliable, given the jamming probability JP of an ACK frame against a single NACK frame, receiver number R, retry limit m and SGE model parameters (p , α , β). We first calculate the residual error rate at the leader receiver, as shown in formula (5.23).

$$PLR^{l}_{SEQ-LBP}(SGE, p, \alpha, \beta, m) = p\alpha^{m} \tag{5.23}$$

As shown in formula (5.24), the calculation of the residual error rate at non-leader receivers is similar to the one over the i.i.d channel model. The first term is the probability that this non-leader receiver loses all the packets in all the $m+1$ rounds while the leader receiver loses all the packets in the first m rounds. The rest sum denotes the residual error rates when the leader receiver gets the packet correctly in each round. The three terms of the sum are the residual error rates in the first (round i), middle and last rounds respectively. Here the jamming probability (by all NACKs) from round i to m is calculated approximately by a product whose terms (T_i) are the average jamming probabilities of each round. Simulation results in NS-2 (will be presented in the next chapter) confirm that this approximation is quite close to the real results. Please also note that $PLR^{nl}_{SEQ-LBP}(SGE, p, \alpha, \beta, R, m, JP) = p\alpha^m$ when $JP = 1$.

$$PLR^{nl}_{SEQ-LBP}(SGE, p, \alpha, \beta, R, m, JP) \cong p^2\alpha^{2m-1} + \sum_{i=1}^{m} \lambda_i \left((1-JP)(1-S_i) + \sum_{j=i+1}^{m} \left((1-JP)(1-S_j) \prod_{l=i}^{j-1} T_l \right) + \prod_{l=i}^{m} T_l \right) \quad (5.24)$$

$$\lambda_i = \begin{cases} p(1-p) & \text{i=1} \\ p^2\alpha^{2i-3}(1-\alpha) & \text{i} \geq 2 \end{cases}$$

$$T_i = \alpha\left(Pj + (1-JP)S_i\right)$$

$$S_i = \sum_{j=1}^{R-2} \left(\binom{R-2}{j} (p\alpha^{i-1})^j (1-p\alpha^{i-1})^{R-2-j} \left(1-(1-JP)^j\right) \right)$$

The average redundancy transmission for one data packet in SEQ-LBP over the simplified GE model is calculated in formula (5.25). The calculation is the sum of the probability that each redundancy transmission round is triggered. In detail, the first term of the sum is the probability that the leader receiver loses the packet in this round and triggers a retransmission from the sender definitely. The second term is the probability that the leader receiver gets the packet correctly but the NACKs from the non-leaders successfully destroy the ACK from the leader in the following rounds, where the jamming probability is calculated approximately based on the current round only. Simulation results confirm that this approximation is quite close to the real results.

$$Expt_{SEQ-LBP}(SGE, p, \alpha, \beta, R, m, JP) \cong 1 + \sum_{i=1}^{m} \left(p\alpha^{i-1} + \lambda_i \sum_{j=i}^{m} S_j \right) \quad (5.25)$$

$$\lambda_i = \begin{cases} 1-p & \text{i=1} \\ pa^{i-2}(1-\alpha) & \text{i} \geq 2 \end{cases}$$

$$S_i = \sum_{j=1}^{R-1}\left(\binom{R-1}{j}(p\alpha^{i-1})^j(1-p\alpha^{i-1})^{R-1-j}\left(1-(1-JP)^j\right)\right)$$

Then we get the average redundant transmission, shown in formula (5.26), which equals the expected number of transmissions minus one.

$$RI_{SEQ-LBP}\left(SGE, p, \alpha, \beta, R, m, JP\right) = Expt_{SEQ-LBP}\left(SGE, p, \alpha, \beta, R, m, JP\right) - 1 \qquad (5.26)$$

Finally, the expected channel holding time for a packet and maximum multicast delay in SEQ-LBP can be calculated as for the i.i.d channel model, shown in formula (5.21) and (5.22) respectively.

5.4. HLBP Performance

5.4.1. Over the i.i.d Channel Model

Now we analyze the performance of HLBP under a similar assumption to the one for LBP and SEQ-LBP: the channel model is identical for each receiver, error events at different receiver are independent and the control frames (SEQ, ACK and NACK) are reliable. The given parameters include the average error rate p of the i.i.d channel, the jamming probability JP of an ACK frame against a single NACK frame, FEC code (n,k), number of receivers R and the block retry limit m for a block of k packets. The residual error rate at the leader receiver can be calculated as formula (5.27) which is the probability that a data packet is lost and cannot be corrected.

$$PLR_{HLBP}^{l}\left(i.i.d, p, k, m\right) = \sum_{i=m+1}^{k+m}\left(\frac{i}{k+m}\binom{k+m}{i}p^i(1-p)^{k+m-i}\right) \qquad (5.27)$$

The residual error rate at non-leader receivers can be calculated as formula (5.28) which is an approximation but confirmed to be very close to the realistic result. The first term is the data packet loss probability at this considered non-leader receiver after m rounds retry and meanwhile the leader receiver have not received k packets for the current block until the $m-1$ retry rounds. The function $DataLR(k,x)$ computes the data packet loss rate after x retry rounds with a data block size k while $BlockLR(k,x)$ denotes the block loss probability after x retry rounds. The following sum in formula (5.28) is the residual data packet loss rate of each retransmission round in the case that the leader receiver gets the whole block correctly (receiving k correct packets) just at this retry round (denoted as λ_i). In the term λ_i, function $FECPLR(k,x,y)$ computes the probability that y packets are still missing to decode the current block after $k+x$ packets have been transmitted. The three terms in the sum denote the residual data packet loss rate in the retry round i, middle rounds and the last round respectively in this case. Similar to the calculation for SEQ-LBP, here we use a product of average jamming probability of each round to approximate the jamming probability from all other non-leader receivers. Simulation results in NS-2 confirm that this approximation is quite close to the realistic results, which will be presented in the next chapter.

$$PLR_{HLBP}^{nl}\left(i.i.d, p, k, R, m, JP\right) \cong DataLR(k,m)BlockLR(k,m-1) \tag{5.28}$$

$$+\sum_{i=1}^{m}\lambda_i\left(\begin{array}{l} DataLR(k,i-1)(1-JP)(1-S_{i-1})+ \\ \sum_{j=i+1}^{m}\left(DataLR(k,j-1)(1-JP)(1-S_{j-1})\prod_{l=i-1}^{j-2}T_l\right)+DataLR(k,m)\prod_{l=i-1}^{m-1}T_l \end{array}\right)$$

$$\lambda_i=\begin{cases}1-BlockLR(k,0) & \text{i=1}\\ FECPLR(k,i-2,1)(1-p) & \text{i} \geq 2\end{cases}$$

$$T_i = JP + (1-JP)S_i$$

$$S_i=\sum_{j=1}^{R-2}\left(\binom{R-2}{j}\left(BlockLR(k,i)\right)^j(1-BlockLR(k,i))^{R-2-j}\left(1-(1-JP)^j\right)\right)$$

$$FECPLR(k,x,y) = \binom{k+x}{x+y} p^{x+y}(1-p)^{k-y}$$

$$BlockLR(k,x) = \sum_{i=x+1}^{k+x} \left(\binom{k+x}{i} p^{i}(1-p)^{k+x-i} \right)$$

$$DataLR(k,x) = \sum_{i=x+1}^{k+x} \left(\frac{i}{k+x} \binom{k+x}{i} p^{i}(1-p)^{k+x-i} \right)$$

The calculation of the expected number of transmissions for a data packet is the sum of the probability that each transmission round is triggered, shown in formula (5.29). In detail, the first term of the sum is the probability that the leader receiver has not received enough packets for the current block in this round and triggers a retransmission from the sender definitely. The second term is the probability that the leader receiver gets the block correctly but the NACK frames from the non-leader receivers successfully destroyed the ACK frame from the leader in the following rounds and prompts a retransmission, where the jamming probability is calculated by an approximated sum only considering the loss probability of the current round with accumulated jamming in half of all related rounds. Simulation results in NS-2 confirm that this approximation is quite close to the realistic result. Please also note that the functions $BlockLR(k,x)$ and $FECPLR(k,x,y)$ are defined in formula (5.28).

$$Expt_{HLBP}\left(i.i.d,\, p,k,R,m,JP\right) = \tag{5.29}$$

$$1 + \frac{1}{k} \sum_{i=1}^{m} \left(BlockLR(k,i-1) + \lambda_i \sum_{j=i}^{m} JamSum(j-1,\, j-i+1) \right)$$

$$\lambda_i = \begin{cases} 1 - BlockLR(k,0) & \text{i=1} \\ FECPLR(k,i-2,1)(1-p) & \text{i} \geq 2 \end{cases}$$

$$JamSum(i,x) \cong \sum_{j=1}^{R-1} \left(\binom{R-1}{j} \left(BlockLR(k,i)\right)^{j} (1 - BlockLR(k,i))^{R-1-j} \left(\left(1-(1-JP)^{j}\right)^{\lceil x/2 \rceil} \right) \right)$$

Then we get the average redundant transmission, shown in formula (5.30), which equals the expected number of transmissions minus one. Similar to SEQ-LBP, thanks to the feedback jamming scheme, there is no unnecessary feedback and retransmissions in HLBP. When the feedback jamming probability is 100 percents, the average redundant transmissions per data packet of HLBP can reach a HEC limit with a certain block size: $RI_{\lim} = \frac{1}{k} \lim_{m\to\infty} \sum_{i=1}^{m} \left(1 - \left(\sum_{j=k}^{k+1-1} \left(\binom{k+i-1}{j} (1-p)^j \, p^{k+i-1} \right) \right)^R \right)$ (can also be derived from formula 5.29 with *JP*=1). Smaller feedback jamming probabilities do not increase the redundant transmissions but cause high residual error rates.

$$RI_{HLBP}\left(i.i.d,\, p, k, R, m, JP\right) = Expt_{HLBP}\left(i.i.d,\, p, k, R, m, JP\right) - 1 \tag{5.30}$$

Finally, the expected channel holding time ET_{HLBP} for a packet with a length of L bytes in HLBP (described in Figure 4.10) can be calculated in formula (5.31),

$$ET_{HLBP}(k,m,L) = \frac{1}{k}\begin{pmatrix} T_{DIFS} + (k-1)\left(T_{DATA}(L) + T_{PLCP}\right) + (k-2)T_{SIFS} \\ + \left(Expt_{HLBP}\left(i.i.d,\, p, k, R, m, JP\right) - k + 1\right)T_{Ex}(L) \end{pmatrix} \tag{5.31}$$

$$T_{Ex}(L) = T_{DATA}(L) + T_{SEQ} + T_{ACK} + 3T_{SIFS} + 3T_{PLCP}$$

where $T_{Ex}(L)$ denotes the channel holding time of the exchange phase in HLBP.

Similar to the calculation for LBP and SEQ-LBP, the maximum delay in HLBP can be calculated as in formula (5.32), which can also be used to select the FEC code (k,n) and retry limit m. The common choice of n also yields $n \geq k+m$. Please also note that in practice n can be less than $k+m$ as the parity packets or even the data packets can be reused for retransmissions. Moreover, here we only consider the multicast transmission delay. Other related delays in the MAC layer or higher layers, such as buffering delays, are not considered here.

$$D_{HLBP} \leq T_{cc} + T_{DIFS} + (k-1)\left(T_{DATA}(L) + T_{PLCP}\right) + (k-2)T_{SIFS} + (m+1)T_{Ex}(L) \tag{5.32}$$

where $T_{Ex}(L)$ is the channel holding time of the exchange phase in HLBP, as defined in formula (5.31).

5.4.2. Over the SGE Model

Now we consider the performance of HLBP over the SGE channel model under similar assumption as the last section. The residual error rate at the leader receiver can be calculated as formula (5.33) which is the probability that a data packet is lost cannot be corrected. Here the function $P_{SGE}[a,b]$ denotes the probability of a errors in a sequence of b packets over the SGE model, as shown in Chapter 3.

$$PLR^{l}_{HLBP}\left(SGE, p, \alpha, \beta, k, m\right) = \sum_{i=m+1}^{k+m}\left(\frac{i}{k+m} P_{SGE}[i, k+m]\right) \tag{5.33}$$

The calculation of the final PLR at non-leader receivers in HLBP over the SGE model is also an approximation formula, shown in formula (5.34), similar to the case over the i.i.d channel model. In detail, the first term is the data packet loss probability at this considered non-leader receiver after m rounds retries and meanwhile the leader receiver have not received k packets for the current block until the $m-1$ retry rounds. The function $DataLR(k,x)$ computes the data packet loss rate after x retry rounds with a data block size k while $BlockLR(k,x)$ denotes the block loss probability after x retry rounds. The following sum of formula (5.34) is the residual data packet loss rate of each retransmission round in the case that the leader receiver gets the whole block correctly (receiving k correct packets) just at this retry round (denoted as λ_i). The three terms in the sum denote the residual data packet loss rate in the retry round i, middle rounds and the last round respectively in this case. Here we use a product of average jamming probability of each round to approximate the jamming probability from all other non-leader receivers.

Simulation results in NS-2 (will be presented in the next chapter) confirm that this approximation is quite close to the realistic results.

$$PLR_{HLBP}^{nl}\left(SGE, p, \alpha, \beta, k, R, m, JP\right) \cong DataLR(k,m)BlockLR(k,m-1) \quad (5.34)$$

$$+\sum_{i=1}^{m}\lambda_i\left(\begin{array}{l} DataLR(k,i-1)(1-JP)(1-S_{i-1})+ \\ \sum_{j=i+1}^{m}\left(DataLR(k,j-1)(1-JP)(1-S_{j-1})\prod_{l=i-1}^{j-2}T_l\right)+DataLR(k,m)\prod_{l=i-1}^{m-1}T_l \end{array}\right)$$

$$\lambda_i = \begin{cases} 1-BlockLR(k,0) & \text{i=1} \\ P_{SGE}[i-1,k+i-2]\left(1-p\right) & \text{i} \geq 2 \end{cases}$$

$$T_i = JP + (1-JP)S_i$$

$$S_i = \sum_{j=1}^{R-2}\left(\binom{R-2}{j}\left(BlockLR(k,i)\right)^j(1-BlockLR(k,i))^{R-2-j}\left(1-(1-JP)^j\right)\right)$$

$$BlockLR(k,x) = \sum_{i=x+1}^{k+x}P_{SGE}[i,k+x]$$

$$DataLR(k,x) = \sum_{i=x+1}^{k+x}\left(\frac{i}{k+x}P_{SGE}[i,k+x]\right)$$

Similar to the calculation for HLBP over the i.i.d channel model, the calculation of the expected number of transmissions for a data packet is the sum of the probability that each transmission round is triggered, shown in formula (5.35). In detail, the first term of the sum is the probability that the leader receiver has not received enough packets for the current block in this round and triggers a retransmission from the sender definitely. The second term is the probability that the leader receiver gets the block correctly but the NACK frames from the non-leader receivers successfully destroyed the ACK frame from the leader in the following rounds and prompts a retransmission, where the jamming probability is calculated by an approximated sum only considering the loss probability of the current round with accumulated jamming in half of all related rounds. Simulation results in NS-2 confirm that this approximation is quite close to the realistic result. Please also note that the function $BlockLR(k,x)$ is defined in formula (5.34).

$$Expt_{HLBP}\left(SGE,p,\alpha,\beta,k,R,m,JP\right)= \tag{5.35}$$

$$1+\frac{1}{k}\sum_{i=1}^{m}\left(BlockLR(k,i-1)+\lambda_i\sum_{j=i}^{m}JamSum(j-1,j-i+1)\right)$$

$$\lambda_i=\begin{cases}1-BlockLR(k,0) & \text{i=1}\\ P_{SGE}[i-1,k+i-2]\left(1-p\right) & \text{i}\geq 2\end{cases}$$

$$JamSum(i,u)=\sum_{j=1}^{R-1}\left(\binom{R-1}{j}\left(BlockLR(k,i)\right)^{j}(1-BlockLR(k,i))^{R-1-j}\left(\left(1-(1-JP)^{j}\right)^{\lceil u/2\rceil}\right)\right)$$

Then we get the average redundant transmission, shown in formula (5.36), which equals the expected number of transmissions minus one.

$$RI_{HLBP}\left(SGE,p,\alpha,\beta,k,R,m,JP\right)=Expt_{HLBP}\left(SGE,p,\alpha,\beta,k,R,m,JP\right)-1 \tag{5.36}$$

Finally, the expected channel holding time for a packet and maximum multicast delay in HLBP can be calculated in formula (5.31) and (5.32) respectively, like for the i.i.d channel model.

5.5. Conclusion

In the Chapter, we first analyze the ACK/NACK jamming probability over the Rayleigh fading channel model. Based on the calculation, we will evaluate the feedback jamming through both theoretical analyses and NS-2 simulations with identical parameters in the next chapter. We also analyze the performance of LBP, SEQ-LBP and HLBP over both the i.i.d channel model and the SGE channel model. These protocols will be evaluated through both analysis results and NS-2 simulation results in the next chapter.

Chapter 6
Performance Evaluation by Analysis and Simulation

In this chapter we evaluate the feedback jamming probability through both calculation results based on the analyses in the last chapter and NS-2 simulation results with identical parameters under various scenarios. Furthermore, the performances of LBP, SEQ-LBP and HLBP are evaluated and compared through both calculation results and NS-2 simulation results in various scenarios. Finally, the performances of those protocols are also compared with the block-ACK polling based protocol and application layer multicast error control protocols.

6.1. Feedback Jamming

6.1.1. NS-2 Environment

NS-2 version 34 is used to build our test environment for the feedback jamming. This version of NS-2 introduces two new modules: Mac802_11Ext and WirelessPhyExt, developed by a team from Mercedes-Benz Research & Development North America and from University of Karlsruhe [Che08]. The extensions are based on Mac802_11 and WirelessPhy, but did a major modification to the original code, in order to provide a significantly higher level of simulation accuracy [Che08]. The new model contains the following features:

- Structured design of MAC functionality modules: transmission, reception, transmission coordination, reception coordination, backoff manager, and channel state monitor;
- Cumulative SINR computation;
- MAC frame capture capabilities;
- Multiple modulation schemes support;
- Packet drop tracing at the PHY layer;
- Nakagami fading model [Stu02];

We have used these two modules in our simulation (shown in Table 6.1). The capture effect in 802.11Ext is described as follows. Typically, there are two capture sections in different phases: preamble capture and data capture. The preamble capture happens during the preamble time in the receiving of each packet. When the second packet comes during the preamble phase of the first packet, it is checked whether the ratio of the signal strength of the first packet over the noise and interfering signals including the second packet is higher than the preamble capture threshold. If it does, the first packet is captured. Otherwise, it is checked whether the ratio of the signal strength of the second packet over the noise and interfering signals including the first packet is higher than the preamble capture threshold. If it does now, the second packet is captured. Otherwise, both packets are not captured. The data capture is similar to the preamble capture but takes place during the receiving period of the data part. For the ACK/NACK jamming, as the time drift set in the simulation is within the preamble duration, only the preamble capture is used for all our simulations.

In our simulation, we also use the TCL script from NS-2.34 and [Che08], IEEE80211a.tcl, which is used to simulate 802.11a. This TCL script implements the changes in parameter values for 802.11a PHY and MAC. We hereby give a brief summary of the parameters defined in 802.11a which helped us get the most accurate simulation model, shown in Table 6.2 and Table 6.3. The parameters' names are self explanatory.

Table 6.1: Code fragment 1: use of PHY and MAC Ext modules

Parameters Set	Values
set opt(chan)	*Channel/WirelessChannel ;#channel type*
set opt(prop)	*Propagation/Nakagami ;#radiopropagation model*
set opt(netif)	*Phy/WirelessPhyExt ; #network interface type*
set opt(mac)	*Mac/802_11Ext ;#MAC model*
set opt(ifq)	*Queue/DropTail/PriQueue ; #queue type*
set opt(ll)	*LL ;#link layer type*
set opt(ant)	*Antenna/OmniAntenna ;#antenna model*
set opt(ifqlen)	*20 ;#max packet in ifq*
set opt(rtg)	*DumbAgent ;#routing agent type*

Table 6.2: Code fragment 2: MAC parameters of the test scripts

Parameters Set	Values
*Mac/802_11 set CWMin_*	*15*
*Mac/802_11 set CWMax_*	*1023*
*Mac/802_11 set SlotTime_*	*0.000009*
*Mac/802_11 set SIFS_*	*0.000016*
*Mac/802_11 set ShortRetryLimit_*	*7*
*Mac/802_11 set LongRetryLimit_*	*4*
*Mac/802_11 set PreambleLength_*	*60*
*Mac/802_11 set PLCPHeaderLength_*	*60*
*Mac/802_11 set PLCPDataRate_*	*6.0e6*
*Mac/802_11 set RTSThreshold_*	*2000*
*Mac/802_11 set basicRate_*	*6.0e6*
*Mac/802_11 set dataRate_*	*6.0e6*
*Mac/802_11Ext set CWMin_*	*15*
*Mac/802_11Ext set CWMax_*	*1023*
*Mac/802_11Ext set SlotTime_*	*0.000009*
*Mac/802_11Ext set SIFS_*	*0.000016*
*Mac/802_11Ext set ShortRetryLimit_*	*0*
*Mac/802_11Ext set LongRetryLimit_*	*0*
*Mac/802_11Ext set HeaderDuration_*	*0.000020*
*Mac/802_11Ext set SymbolDuration_*	*0.000004*
*Mac/802_11Ext set BasicModulationScheme_*	*0*
*Mac/802_11Ext set use_802_11a_flag_*	*true*
*Mac/802_11Ext set RTSThreshold_*	*2000*
Mac/802_11Ext set MAC_DBG	*0*

Table 6.3: Code fragment 3: PHY parameters of the test scripts

Parameters Set	**Values**
*Phy/WirelessPhy set CSThresh_*	*6.30957e-12*
*Phy/WirelessPhy set Pt_*	*0.01; #0.01w, 10dbm*
*Phy/WirelessPhy set freq_*	*5.18e9*
*Phy/WirelessPhy set L_*	*1.0*
*Phy/WirelessPhy set RXThresh_*	*3.652e-10*
*Phy/WirelessPhy set bandwidth_*	*20e6*
*Phy/WirelessPhy set CPThresh_*	*10.0*
*Phy/WirelessPhyExt set CSThresh_*	*6.31e-12 ;#-82 dBm*
*Phy/WirelessPhyExt set Pt_*	*0.1; #20dbm*
*Phy/WirelessPhyExt set freq_*	*5.18e9*
*Phy/WirelessPhyExt set noise_floor_*	*1.0e-12; #-90dbm*
*Phy/WirelessPhyExt set L_*	*1.0*
*Phy/WirelessPhyExt set PowerMonitorThresh_*	*0;*
*Phy/WirelessPhyExt set HeaderDuration_*	*0.000020*
*Phy/WirelessPhyExt set BasicModulationScheme_*	*0; #*
*Phy/WirelessPhyExt set PreambleCaptureSwitch_*	*1*
*Phy/WirelessPhyExt set DataCaptureSwitch_*	*0*
*Phy/WirelessPhyExt set SINR_PreambleCapture_*	*3.1623;*
*Phy/WirelessPhyExt set SINR_DataCapture_*	*100.0; #w, 20db*
*Phy/WirelessPhyExt set trace_dist_*	*1e6*
*Phy/WirelessPhyExt set PHY_DBG_*	*0*

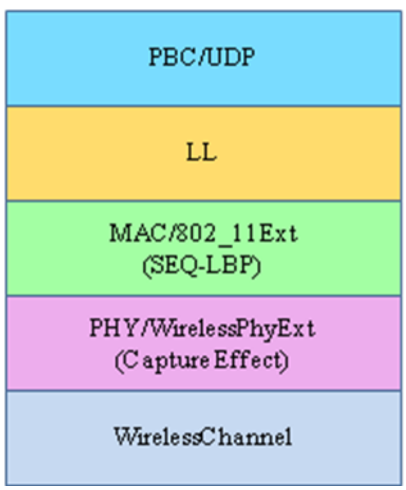

Figure 6.1: Simulation architecture for feedback jamming

Please note that the Rayleigh fading model is used for our simulation, which is a special case of the Nakagami channel model [Stu02]. We implement SEQ-LBP in the MAC layer based on the 802.11Ext package. The Periodic Broadcast (PBC) agent from [Che08] is used as the transmission load to test the feedback jamming. The simulation architecture is shown in Figure 6.1. The results of the ACK/NACK jamming for each data packet in SEQ-LBP are recorded. Simulation results in various scenarios are shown in the next section.

6.1.2. Evaluation

According to the analysis in the last chapter, we now calculate the jamming probability of one ACK frame and different number of NACK frames under different cases with variable distances from the leader to the AP but with a fixed distance (5m) from non-leaders to the AP. The main parameters are set as in Table 6.4 (some of them have already been shown in Table 6.2 and Table 6.3). We also run the simulation in NS-2.34 (shown in section above) with identical parameters. Figure 6.2 shows the results. For a better layout, the jamming failure probability is used in the figure, which equals to one minus the jamming probability.

Table 6.4: Experiment Parameters for feedback jamming

Parameters	Values
Path loss exponent	3
Transmitting power	20dBm
Bandwidth	20e6Hz
Carrier Frequency	5.1e9Hz
AWGN noise floor	1.0e-12W
Data rate	6Mbps
SINR preample capture threshold	5dB

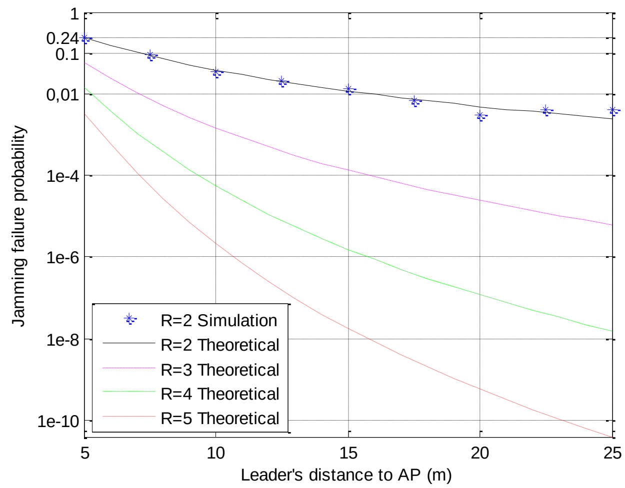

Figure 6.2: Feedback jamming failure probability (non-leaders' distance to the AP: 5m)

From Figure 6.2, we can see the feedback jamming probability is about 0.76 for two receivers over the Rayleigh fading channel in the worse case that both the leader and non-leader have the same distance from the AP. For normal cases that one receiver is further away from the AP than the other and can be selected as the leader, the feedback jamming probability can be 0.90+. The results also show that the feedback jamming probability becomes more than 0.90 for more than two receivers and even as high as 0.99 for 5 receivers. We explore further for the worst case that all receivers located at the same distance to the AP. Figure 6.3 shows the results from both the theoretical calculation and NS-2 simulation.

From Figure 6.3, we can see the simulation results and the analysis results match very well. The feedback jamming probability is about 0.76 for two receivers over the Rayleigh fading channel in the worse cases that both the leader and non-leader have the same distance from the AP. The results also show that the feedback jamming probability becomes 0.90+ for more than two receivers and as high as 0.99+ for 5 receivers even for the worst case. Please also note that we are considering the Rayleigh fading channel where the channel conditions change frequently. So the feedback jamming probability will be much higher if the leader always experiences the worst channel condition for sure by dynamic leader selections. Moreover, if the transmitting power of the feedback can be controlled dynamically and a higher transmitting power can be used for NACK frames, the ACK/NACK jamming probabilities will be higher.

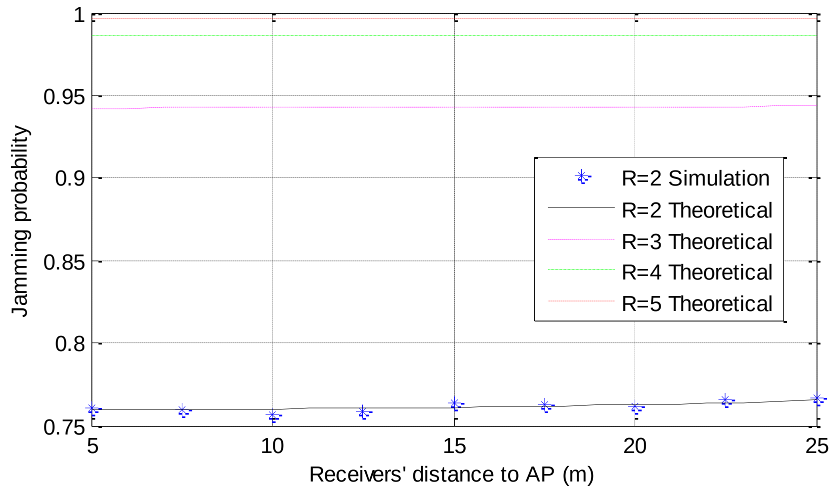

Figure 6.3: Feedback jamming probability in worst cases

6.2. LBP, SEQ-LBP and HLBP

6.2.1. NS-2 Environment

In this section, to evaluate the performance of LBP, SEQ-LBP and HLBP, we present our calculation results and NS-2 simulation results using identical parameters setting according to IEEE 802.11a. As described in Chapter 3, the i.i.d channel model is actually a special case of the SGE model when the temporal correlation coefficient equals to zero. So here we only consider the SGE model but using various parameters.

We conduct our simulation study using NS-2 (version 2.26) and implement LBP, SEQ-LBP and HLBP based on the IEEE 802.11e simulation model from [Wie06a] [Wie06b]. All client nodes are one hop to the AP and at most two hops to each other. We use IEEE 802.11a parameters to model the physical layer. The data rate we choose is 24Mbps. The first receiver that joins the multicast group acts as the leader. The load date rate is about 4.5Mbps with a packet interval 2.5ms. The total payload length in the MAC layer is 1356 bytes, and there is no fragmentation in the MAC layer or the network layer. The application layer multicast error control scheme ARQ (AL-ARQ or HEC-PR [Tan09]) and HARQ Type I are implemented based on the real-time transport protocol (RTP) [Sch96], [Ott04]. And RTP runs on the Multicast Ad-hoc On-demand Distance Vector (MAODV) routing protocol [Roy00], [Zhu04], which is simplified a little to suit wireless LANs. Normally we do not perform the application layer approaches if not otherwise stated.

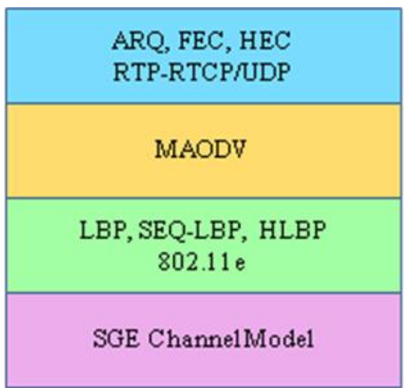

Figure 6.4: The simulation architecture for the proposed protocols

The simplified GE channel model is implemented in the physical layer, but it is used only for data frames. The MAC control frames (RTS, CTS, SEQ and ACK) are error free from the error model. (The control frames also may be lost because they might collide with the background traffic.) The average packet error rates at all receivers are the same and the error events at different receivers are independent. The temporal correlation coefficient of the simplified GE model is set to $\rho = 0.1$ which is proper for common wireless LANs [Li09] [Tan09]. The parameters of the SGE error model and the feedback jamming probability can be adjusted through simulation scripts in TCL. In other words, the PHY layer is simulated as a SGE channel model. The architecture is shown in Figure 6.4.

We hereby give a brief summary of the parameters used in our simulation scripts, shown in Table 6.5. The parameters' names are self explanatory.

Table 6.5: Code fragment 4: Test scripts for protocol simulations

Parameters Set	**Values**

*#RTP, RTCP
set session_bw 24000000
set payload_size 1316
set ldf_l 0.05
set ldf_k 0.05
set dither_max 0.02
set rtt_sim 1
set interval 0.0025
set session_time 100000
 	
#PHY and MAC
Mac/802_11em set bandwidth_ 24Mb
Mac/802_11em set dataRate_   24Mb
Mac/802_11em set basicRate_ 6Mb
 	
#MAODV and RTP-RTCP
set agent_s [new Agent/RTP_STD]
$agent_s set dst_addr_ 0xE000000
$ns at 0.0100000000 "$node_r(0) aodv-join-group 0xE000000"
set agent_r($i) [new Agent/RTP_STD]
$agent_r($i) set dst_addr_ 0xE000000
$ns at 0.0100000000 "$node_r($i) aodv-join-group 0xE000000"
 	
#SGE model
for {set i 1} {$i <= $rev_num} {incr i} {
        $node_r($i) SetGeErrorRate $err_1
        $node_r($i) SetGeErrorCorr $corr_1
}*	

The simulation results of LBP, SEQ-LBP and HLBP are shown in the next section. Further evaluation on a real test-bed built with consumer wireless LAN hardware will be presented in the next chapter.

6.2.2. Performance Evaluation

We first evaluate the final PLR at non-leader receivers in LBP, SEQ-LBP and HLBP under various scenarios. Figure 6.5 – Figure 6.10 present both the calculation results and simulation results. We refer to the residual error rate as the one at non-leader receivers if not otherwise stated. From the figures, we can see first that the theoretical calculation results and simulation results match very well and hence our analyses are verified. LBP, SEQ-LBP and HLBP correct multicast losses roughly at an amount larger than the feedback jamming probability (Figure 6.5, Figure 6.7 and Figure 6.9). The feedback jamming probability has a great influence on the final residual error rates in these protocols. We will explore the feedback jamming probability further on a test-bed built using consumer IEEE wireless LAN cards.

In detail, from Figure 6.5 and Figure 6.6 we can see that LBP can correct multicast losses roughly at an amount larger than the feedback jamming probability when the retry limit is no less than 4. For example, as shown in Figure 6.5, about 99% losses can be corrected with a feedback jamming probability 0.90. Moreover, the residual error rates in LBP keep stable when the retry limit is more than 4 for normal size (1~7 receivers) wireless LANs (Figure 6.6). The residual error rates have a good convergence property, which is necessary for the implementation of these protocols in practice. As described in the last chapter, please also note that in all the analyses and simulations for LBP it is assumed that the data frames are only partially damaged and non-leader receivers can reply NACKs based on the data frames (feedback time and destination). As a result the performances of LBP are just upper bounds and might not always hold in practice. However, this assumption has no impact on the performance comparison with other protocols as LBP has the worst performance.

Similar to LBP, from Figure 6.7 and Figure 6.8 we can see that SEQ-LBP can correct multicast losses at an amount roughly larger than the feedback jamming probability when the retry limit is no less than 4. For example, as shown in Figure 6.7, about 99% losses can be corrected with a feedback jamming probability 0.90. Moreover, the residual error rates in SEQ-LBP do not increase with the retry limit when the retry limit is more than 4 for normal size (1~7 receivers) wireless LANs (Figure 6.8). The residual error rates in SEQ-LBP have a good convergence property as in LBP. From the simulation results, it seems LBP has the same effectiveness to correct multicast losses as SEQ-LBP. However, as we just discussed in the last paragraph, the performance of LBP in this simulation is just an upper bound for comparison simplicity.

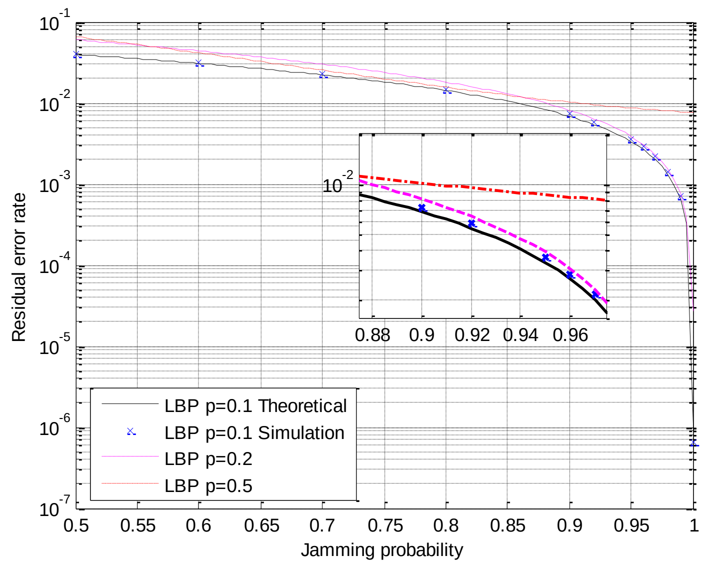

Figure 6.5: LBP residual error rate vs. Feedback jamming probability (Retry limit 7)

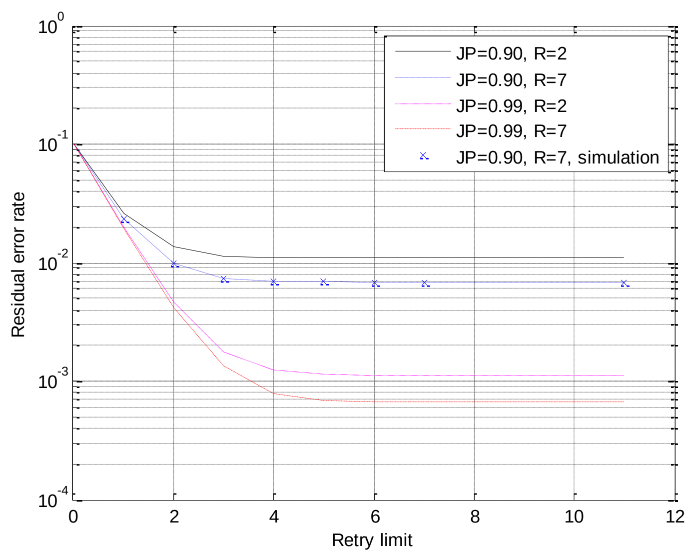

Figure 6.6: LBP residual error rate vs. Retry limit (error rate 0.10)

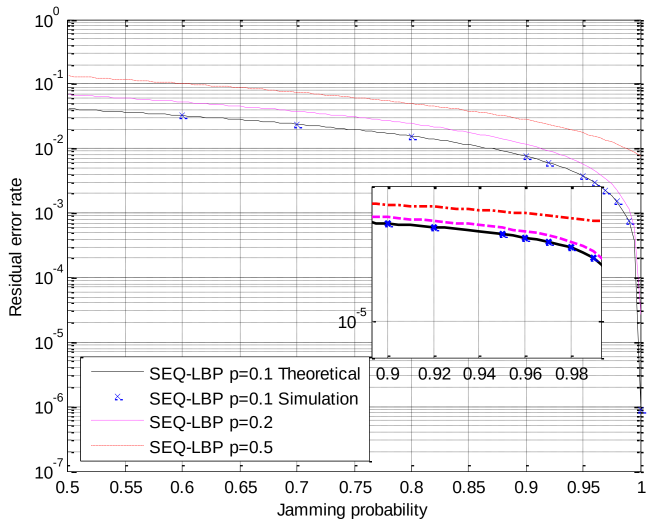

Figure 6.7: SEQ-LBP residual error rate vs. Feedback jamming probability (Retry limit 7)

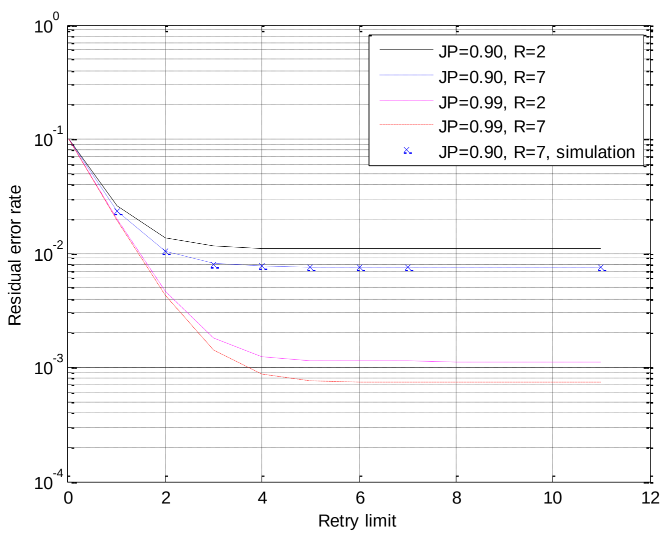

Figure 6.8: SEQ-LBP residual error rate vs. Retry limit (error rate 0.10)

Figure 6.9 and Figure 6.10 present the performances of HLBP on residual error rates with different feedback jamming probabilities and retry limits respectively. Similar to LBP and SEQ-LBP, it is observed that HLBP can correct multicast losses roughly at an amount a little larger than the feedback jamming probability. For example, as shown in Figure 6.9, about 99% losses can be corrected with a feedback jamming probability 0.90. HLBP and SEQ-LBP have the same effectiveness to correct multicast losses as they use the same feedback jamming scheme. Moreover, for HLBP with $k = 20$, the residual error rates do not increase with the block retry limit when the block retry limit is more than 12 (Figure 6.10). The residual error rates have a good convergence property as well. From the simulation results, we can also see that the required average retry limit per data packets (e.g. 12/20 in this simulation case) in HLBP is much smaller than the one (e.g. 4) in SEQ-LBP and LBP. This is due to the block coding and block feedback in the MAC layer.

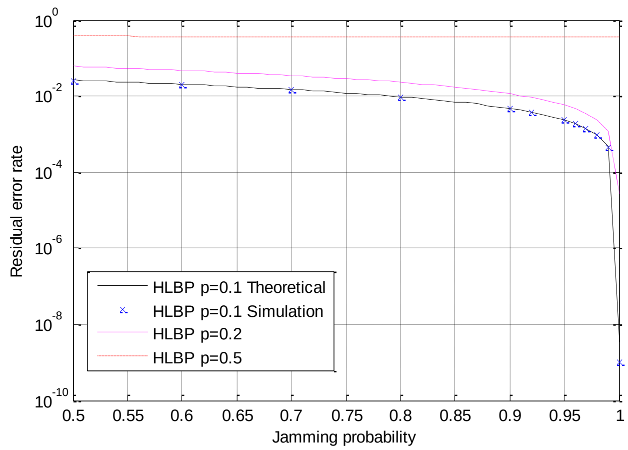

Figure 6.9: HLBP residual error rate vs. Feedback jamming probability (*k*=20; *p*=0.10; *m*=20)

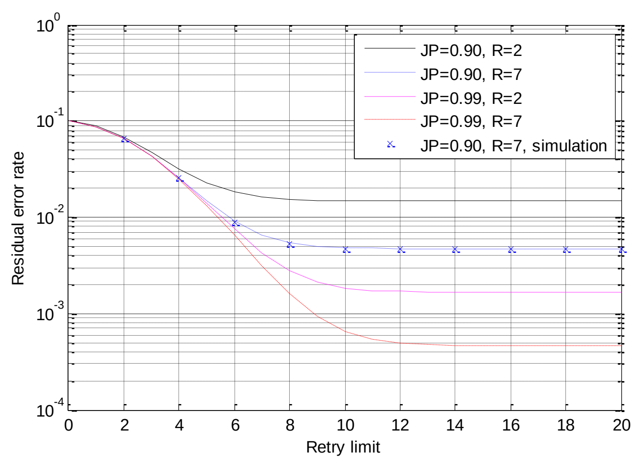

Figure 6.10: HLBP residual error rate vs. Block retry limit (*k*=20; *p*=0.10)

Now we consider the efficiency of these protocols and compare them in a figure. Figure 6.11 and Figure 6.12 show their average redundancy transmissions while Figure 6.13 to

Figure 6.15 show their average channel holding time in various scenarios. From Figure 6.11, we can see that the theoretical analyses of SEQ-LBP and HLBP match the simulation results very well, especially for high feedback jamming probabilities. For LBP, the analysis results are a little lower than simulation results. However, as LBP is typically less efficient than both SEQ-LBP and HLBP, it is a good approximation for LBP to use the analysis results for comparison with other protocols. From now on, we just use the analysis calculation to compare the performance of LBP, SEQ-LBP and HLBP if not otherwise stated. A feedback jamming probability of 0.99 is used for calculation, which is a normal value according to the experiments in real wireless LANs.

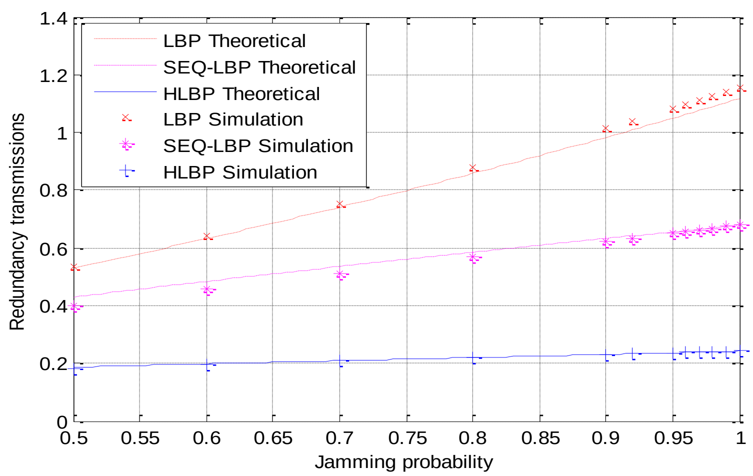

Figure 6.11: Redundant transmission vs. Feedback jamming probability (p=0.10; R=7; LBP and SEQ-LBP: m=7; HLBP: k=20; m=20)

Figure 6.12 presents the redundancy transmission with different number of receivers for LBP, SEQ-LBP and HLBP. For LBP, the redundancy transmission increases sharply with the number of receivers. Among these protocols, LBP is the least efficient and has the lowest scalability. As expected, SEQ-LBP improves LBP very much due to the SEQ frame and the ACK/NACK jamming mechanism. SEQ-LBP is more efficient and has a higher scalability than LBP. We also can see that the redundancy transmission of SEQ-LBP is quite close to the ARQ limit which is calculated ideally with perfect feedback (no unnecessary ones) over the i.i.d channel model, e.g. RI=0.1111 when p=0.10 and R=1. As we discussed in the last chapter, the redundancy transmission of SEQ-LBP just reaches the ARQ limit if the feedback jamming probability is 100 percents. Meanwhile HLBP has the highest efficiency and scalability due to the packet level FEC coding and block feedback. Similarly, due to the feedback jamming scheme and the MAC layer FEC coding, the performance of HLBP is also quite close to the HEC limit with the same block size (e.g. *k*=20). SEQ-LBP is a good choice for small multicast groups while HLBP is a better choice for large multicast groups.

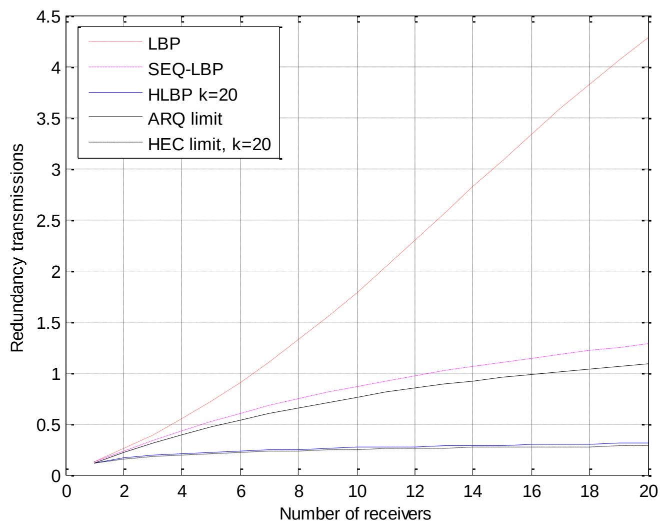

Figure 6.12: Redundant transmission vs. Number of receivers (*JP*=0.99; *p*=0.10; LBP and SEQ-LBP: *m*=7; HLBP: *k*=20; *m*=20)

Figure 6.13 to Figure 6.15 compare the average channel holding time of LBP, SEQ-LBP and HLBP under various scenarios. The average channel holding time is a natural criterion because the reciprocal of it provides a measure of throughput. Figure 6.13 shows the average channel holding time with different number of receivers. For LBP, we can see the average channel holding time increases sharply with the number of receivers. SEQ-LBP is more efficient than LBP, especially for large multicast groups. We also note that LBP is more efficient than SEQ-LBP at low error rates and with small numbers of receivers. This is because in these cases the overhead of the SEQ frame counteracts the benefit that the SEQ frame creates. Due to the limitation of scalability, both LBP and SEQ-LBP are not efficient for large multicast groups. As expected, due to FEC coding and block feedback, HLBP is much more efficient than LBP and SEQ-LBP, and the average channel holding time even keeps stable as the number of receivers increases. For HLBP, a larger FEC code *k* leads to a higher efficiency but a longer multicast delay as well. The choice of FEC code can follow formula (5.32) under the requirements of real applications.

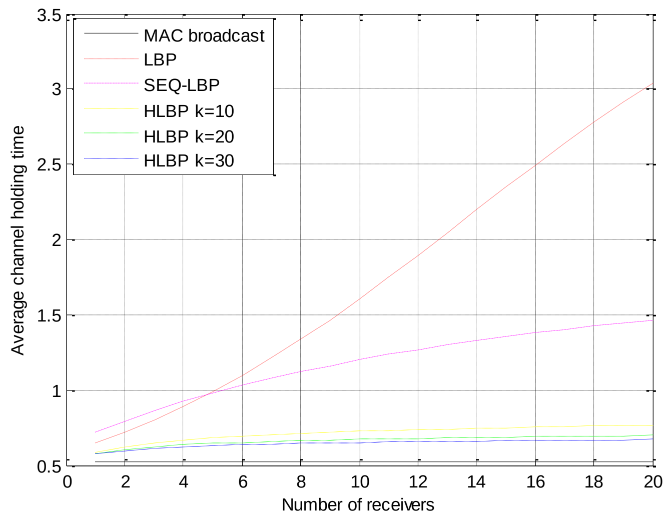

Figure 6.13: Channel holding time vs. Number of receivers (*JP*=0.99; *p*=0.10; LBP and SEQ-LBP: *m*=7;)

Figure 6.14 presents the average channel holding time with different channel error rates. The results show that SEQ-LBP is more efficient than LBP for high error rates due to the sequence check and feedback jamming. Meanwhile, LBP is more efficient than SEQ-LBP for good channel conditions (low error rates) because of the overhead of SEQ frames. Moreover, as both LBP and SEQ-LBP are pure ARQ schemes, the average channel holding time increases sharply with the error rate. As expected, due to FEC coding and block feedback, HLBP is more efficient than both LBP and SEQ-LBP, and the average channel holding time increases slowly with the error rate. Please also note that HLBP is even more efficient than MAC broadcast when the error rates are extremely low. This is because the packets in HLBP are sent based on blocks (e.g. reserve the channel once per block) and hence the average channel management overhead per packet is lower than the single MAC broadcast.

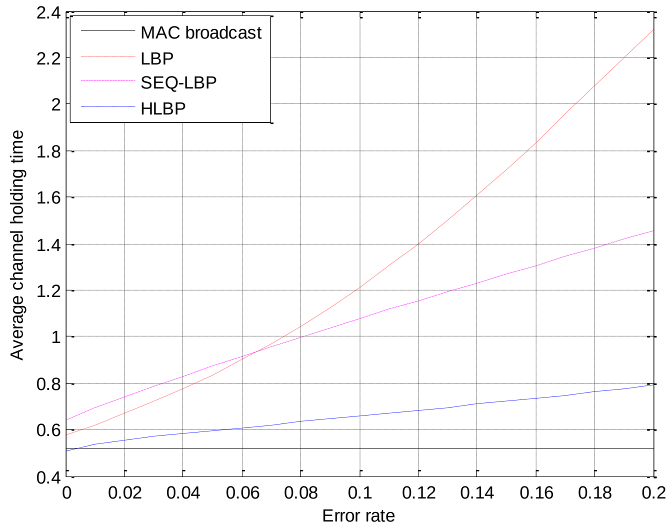

Figure 6.14: Channel holding time vs. Error rates (*JP*=0.99; *R*=7; LBP and SEQ-LBP: *m*=7; HLBP: *k*=20; *m*=20)

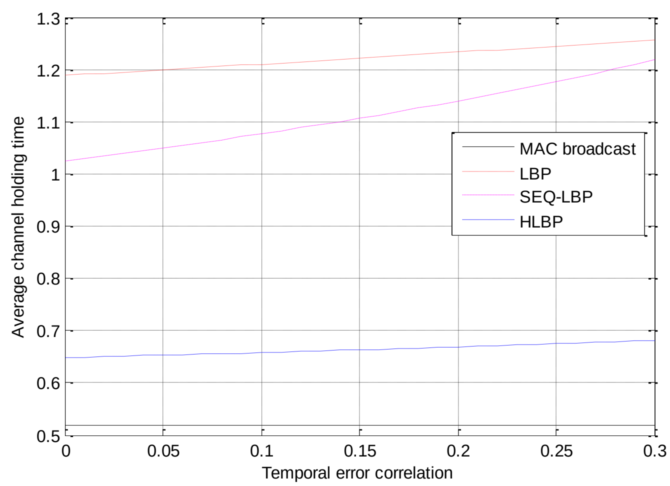

Figure 6.15: Channel holding time vs. Temporal error correlation (*JP*=0.99; *R*=7; *p*=0.10; LBP and SEQ-LBP: *m*=7; HLBP: *k*=20; *m*=20)

At last, Figure 6.15 shows the influence of the temporal error correlation of the SGE model. For all LBP, SEQ-LBP and HLBP, the average channel holding time increases very slowly with the temporal error correlation. Compared with SEQ-LBP, LBP suffers less influence because there are many unnecessary retransmissions in LBP. Meanwhile, HLBP suffers less influence from the temporal error correlation as well. This is because HLBP use block transmission and feedback. The larger the block is, the less influence HLBP will suffer from.

6.2.3. Compared with other Protocols

In this section, we compare the proposed protocols SEQ-LBP and HLBP with block-ACK polling and application layer HARQ. The block-ACK polling is described in Chapter 2. Sequence numbers are used to number frames. A block of data frames are transmitted at a time once the channel is granted. Then the sender polls each receiver and each receiver replies a block ACK which indicates the transmission result of each data frame of the block based on bitmap. Any loss of ACK leads the sender to retransmit the data frame until all ACKs are received or the retry limit is reached. A more efficient polling scheme is that the sender sends an ACK-Request frame to arrange for each receiver to reply ACK at a scheduled time. The latter one is used for comparison as it is more efficient.

Figure 6.16 presents the theoretical calculation curves of the average channel holding time with different number of multicast receivers for HLBP and block-ACK polling. Here, the jamming probability is set to 100 percent as the numbers of receivers are large. The i.i.d channel model is used for analysis simplicity. Under these conditions, as discussed in chapter 5, here the average redundant transmissions of SEQ-LBP and HLBP can reach the ARQ limit and HEC limit (versions of Shannon limit) respectively. Please note that SEQ-LBP is a special case of HLBP when k=1. The results show that SEQ-LBP is more efficient than the single packet polling scheme (block-ACK polling size 1), especially for large multicast groups. This is because the ACK/NACK jamming based multicast feedback is much more efficient than polling based feedback and the first one has a higher scalability. From the results, we also can see that HLBP is more efficient than block-ACK polling with the same block size. The performance of the block-ACK polling is far away from the Shannon limit. For large multicast groups, the error correction costs of the block-ACK polling are even twice higher than the one of HLBP. This is because HLBP uses FEC coding and ACK/NACK jamming based multicast block feedback which is efficient and has a higher scalability.

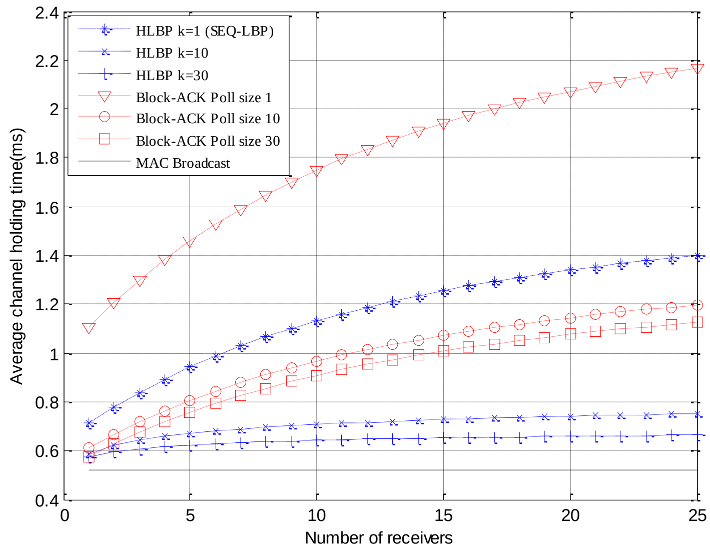

Figure 6.16: HLBP vs. Block-ACK Polling (*JP*=1.0; i.i.d channel model *p*=0.10;)

We than compare HLBP with an application layer multicast error correction approach: HARQ Type I. The FEC codes of HLBP and HARQ Type I both are obtained by performance optimization under delay constraints (e.g. from formula 5.22 for HLBP). Figure 6.17 shows the simulation results. The results show that HLBP is always more efficient than HARQ Type I, especially under short delay constraints. This is because the MAC layer ACK/NACK jamming based block feedback and retransmission of HLBP are much faster than the application layer feedback and retransmission in HARQ Type I and so HLBP can use a larger FEC code k, hence it is more efficient. Moreover, the performance of HARQ Type I decreases sharply when the delay constraints are very short. This is due to the fact that the application layer feedback and retransmission in HARQ Type I always take a long time and HARQ Type I has to switch to a pure FEC scheme (no ARQ) when the delay constraint are very short, hence it is not efficient. However, due to the FEC coding and the fast ACK/NACK jamming based feedback in the MAC layer, HLBP is always very efficient even when the delay constraints are very short. Please also note that here we only consider the one-hop 802.11 wireless networks, wireless LANs. For multi-hop multicast in wireless networks such as wireless Sensor, Mesh and Ad-Hoc Networks, HLBP will be much more efficient than HARQ Type I as HLBP can correct multicast errors at local multicast branch nodes and meanwhile HARQ Type I corrects multicast losses at remote sender which causes unnecessary retransmissions over multicast members with good channel conditions.

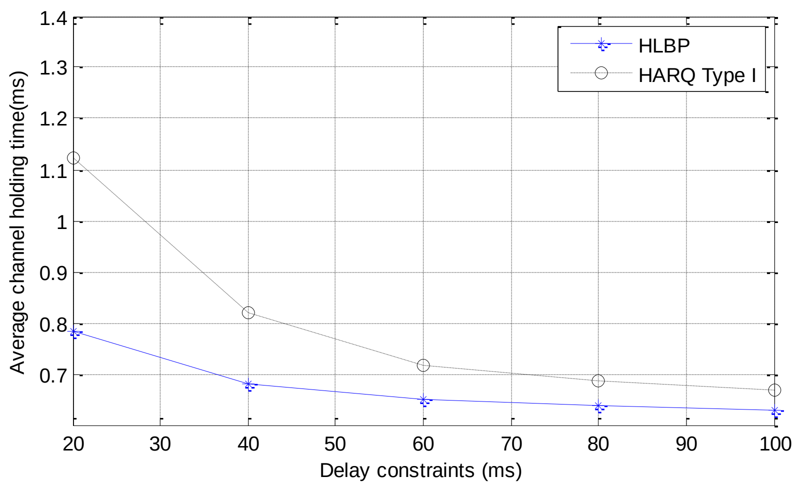

Figure 6.17: Channel holding time vs. Delay constraints (*JP*=1.0; *R*=7; SGE model *p*=0.10;)

6.2.4. Evaluation in Multi-hop Scenarios

In this section, we roughly evaluate the performance of SEQ-LBP and HLBP for 802.11 based wireless multi-hop networks. A typical kind of topology (two level multiway tree) is used, where the sender has r one-hop receivers or called relay nodes and each relay node has r one-hop receivers as well. Each sub-group has its own transmission range. The topology with two level triple-groups (triple tree) is shown in Figure 6.18. First, for MAC broadcast without any error correction, let ET_{BST} denote the average channel holding time for one hop broadcast. Then, for a two level r-way tree multicast, the total average channel holding time can be calculated as $ET_{BST}^{r}() = ET_{BST} * (r+1)$. For SEQ-LBP, let $ET_{SEQ-LBP}(R)$ denote the average channel holding time for one-hop multicast with R receivers. Then, for a two level

r-way tree multicast, the total average channel holding time can be calculated as $ET^{r}_{SEQ-LBP}() = ET_{SEQ-LBP}(r)*(r+1)$. Similarly, for HLBP, $ET^{r}_{HLBP}() = ET_{HLBP}(r)*(r+1)$. For application layer end-to-end ARQ (AL-ARQ), let $ET_{AL-ARQ}(R)$ denote the average channel holding time and $FB_{AL-ARQ}(p)$ denotes the feedback overhead for one receiver in one-hop multicast. Then, for a two level r-way tree multicast, the total average channel holding time can be calculated as $ET^{r}_{AL-ARQ} = ET_{AL-ARQ}\left(r*(r+1)\right)*(r+1) + FB_{AL-ARQ}(p)*r*r*2$. Similarly, for HARQ Type I, $ET^{r}_{HARQ-I} = ET_{HARQ-I}\left(r*(r+1)\right)*(r+1) + FB_{HARQ-I}(p)*r*r*2$.

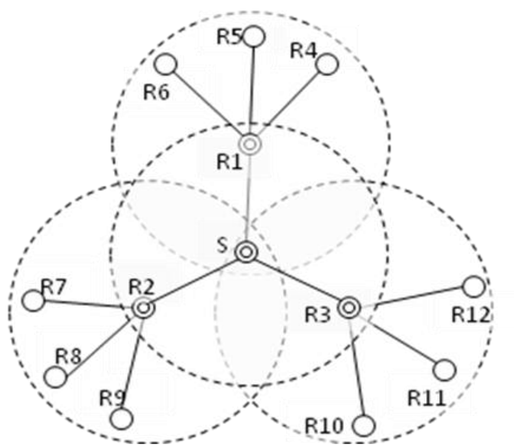

Figure 6.18: A Multi-hop Topology with Triple-groups (triple tree)

We first compare the performances of SEQ-LBP, HLBP, AL-ARQ and HARQ Type I for a two level triple tree shown in Figure 6.18. For calculation simplicity, the i.i.d channel model is used and the feedback jamming probability is set to 1. Figure 6.19 shows the calculation results. As expected, AL-ARQ is the least efficient one. SEQ-LBP improves AL-ARQ a lot due to local error correction and feedback jamming. However, as the overhead of the SEQ frame, SEQ-LBP is less efficient than AL-ARQ under very good channel condition (low error rates). Please note that SEQ-LBP has a very short multicast delay due to the local MAC layer multicast error correction. Figure 6.19 also shows that both HARQ Type I and HLBP are more efficient than AL-ARQ and SEQ-LBP thanks to packet level FEC coding. HLBP is much more efficient due to local multicast error correction. Please also note, here

HLBP uses the same large *k* as HARQ Type I does. In practice, due to the short delay of local multicast error correction, HLBP can use a much larger *k* than HARQ Type I under the same end-to-end delay requirements, and thus HLBP could be more efficient. Please also note that the block based protocols are even more efficient than MAC broadcast when the error rates are extremely low as the block based channel management overhead (per packet) is lower than the single MAC broadcast, e.g. they only reserve the channel once per block.

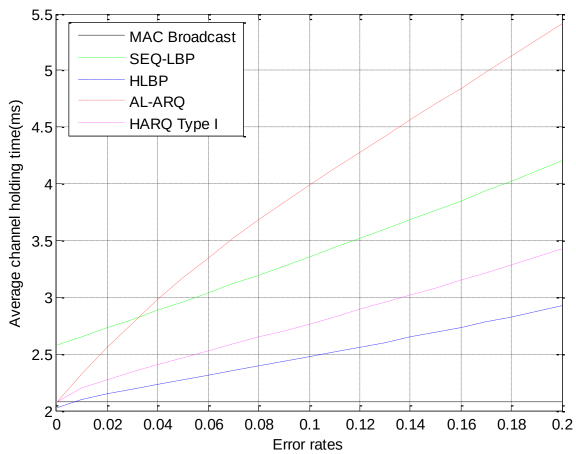

Figure 6.19: Two level triple tree multicast (k=20 for both HLBP and HARQ Type I)

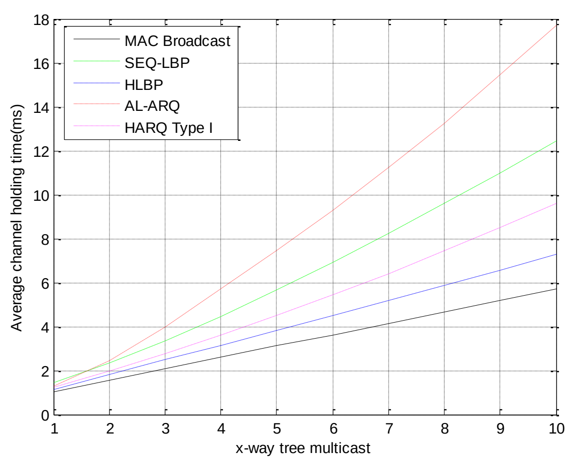

Figure 6.20: Two level x-way tree multicast (k=20 for both HLBP and HARQ Type I)

Next, we compare those protocols in networks with different sizes. Figure 6.20 shows their performances for different two level x-way tree multicast which are similar to the topology of Figure 6.18. As expected, the MAC layer protocols SEQ-LBP and HLBP is much more efficient than end-to-end approaches AL-ARQ and HARQ Type I respectively, especially for large size multicasts. Moreover, here we consider only two-hop multicast. For larger hop multicast, local approaches could be more efficient and have much shorter multicast delays relatively.

6.3. Conclusion

In the Chapter, we evaluate the ACK/NACK jamming through both analysis results and NS-2 simulations under various scenarios. Both analysis results and simulation results confirm that the feedback jamming probability could be as high as 0.99 when the receiver with the worst channel condition is chosen as the leader for normal scenarios and about 0.76 for the worst case with only two receivers which experience nearly the same channel condition. We will explore the feedback jamming probability further on a real test-bed built with consumer wireless LAN cards in the next Chapter.

We then compare the protocols SEQ-LBP and HLBP with LBP through both theoretical calculation results and NS-2 simulation results under various scenarios. We also compare SEQ-LBP and HLBP with block-ACK polling, application layer ARQ and HARQ Type I, even in a multi-hop multicast scenario. Both analysis results and simulation results verify that the SEQ-LBP protocol is a good choice for small multicast groups due to its simplicity, efficiency and short delay thanks to the feedback jamming mechanism. For large multicast groups, HLBP is a better choice for its high efficiency and scalability thanks to the FEC coding and ACK/NACK jamming based block feedback.

Chapter 7

Experimental Evaluation on a Real Test-bed

In the last Chapter, the feedback jamming was evaluated through both theoretical calculations and NS-2 simulations. Given a feedback jamming probability, we also analyzed the performance of LBP, SEQ-LBP and HLBP over both the i.i.d channel model and the SGE channel model, evaluated them on NS-2 and compared them with block-ACK polling and application layer approaches.

To further explore the feedback jamming scheme and the proposed protocols, in this chapter, we build a test-bed using consumer wireless LAN cards and test them in a real environment. Using *Atheros* chipset along with the *Madwifi* driver, a flexible software platform that runs in real-time Linux is designed. The test-bed supports microsecond precision and packet transmission at a configurable time and frame format by not triggering hardware level CSMA contention or backoff schemes. We also develop and implement a Dynamic Leader Selection (DLS) algorithm and a multicast management mechanism. To evaluate the feasibility of feedback jamming, the hardware ACK/NACKs jamming probabilities are measured in various scenarios. Moreover, a driver level SEQ-LBP is also implemented and evaluated.

7.1. Test Environment

7.1.1. Related Work

The low-cost of 802.11 devices and its wide availability has made it the de-facto choice for developing and evaluating new wireless systems and applications. Many researchers have successfully built wireless 802.11 test-beds. Using both experimental simulations and test-bed measurements, many studies [Neu05] [Lu08a] [Sha06] [Dju09] have identified the limitations of the conventional CSMA MAC protocols used by 802.11 devices, and proposed modifications and alternatives. In addition to incorporating these new designs in future wireless devices, several efforts modify commodity 802.11 devices for immediate benefits.

SoftMAC [Neu05] is a software system developed at the University of Colorado built to provide a flexible environment for experimenting with MAC protocols. The ability to cheaply create, modify and conduct system level experimentation with hardware is often a goal of many research projects. However, many of these projects ultimately fail due to the cost, time, and effort involved in deploying a large scale experimental platform. The SoftMAC platform fills this need. SoftMAC uses a commodity 802.11b/g/a networking card with a chipset manufactured by the *Atheros* Corporation to build a software radio with predefined physical layers but a flexible MAC layer. Internally, the *Atheros* chipset provides considerable flexibility over the format of the transmitted packets, though this flexibility is not generally exposed by network drivers. By reverse-engineering many of those controls, SoftMAC provides a driver that allows extensive control over the MAC layer while still allowing use of the waveforms defined by the underlying 802.11b/g/a physical layers. SoftMAC also includes a software control system that allows its users to address many of the "systems level" issues facing researchers.

In terms of host-based platforms, FlexMAC [Lu08a] [Lu08b], SoftMAC and MadMAC [Sha06] are the most similar work in the literature. While MadMAC and SoftMAC seek to broaden the range of supported MAC protocols, FlexMAC focus on 802.11-style protocols, i.e. various variants of CSMA. Specifically, FlexMAC allow flexible host-based implementations of retransmission, backoff, and timing of transmissions. Because FlexMAC leverages a standard 802.11 card, it is relatively easy to have the protocols coexist or even interoperate with 802.11. If interoperability can be sacrificed, it is likely to make even more radical changes, e.g. allowing the host to generate acknowledgements.

Many literatures have also successfully shown that, in addition to CSMA protocol, 802.11 hardware can be used to build non-CSMA protocols such as TDMA. Soft-TDMA [Dju09] is a similar TDMA protocol test-bed whose dependence on the *Madwifi* driver is weaker. The entire MAC is implemented in Linux user space, without the use of any special features of the hardware, e.g. *Atheros* hardware timers. Soft-TDMAC only relies on the 802.11 QoS features provided by the driver, which are also available in other wireless drivers. With precise clock synchronization, soft-TDMA achieves a higher throughput than legacy CSMA protocols.

Based on our ample survey of the related literature, there is still no evaluation of MAC layer reliable multicast protocol on real test-beds, let alone ACK/NACK jamming based ones.

7.1.2. Test Environment

We test the ACK/NACK jamming on IBM X31 laptops with *Atheros* Communications AR5212 chipsets using the *Madwifi* driver through the Linux networking sub-system. The laptops run Linux kernel 2.6.26 with the real-time extensions[8] (kernel 2.6.26-rt16). The linux real-time extension streamlines the kernel to remove unnecessary software locks and provides preemptive priority-based thread scheduling, which is necessary for precise software timers.

[8] http://www.kernel.org/pub/linux/kernel/projects/rt/

To test the ACK/NACK jamming in the driver level and to implement SEQ-LBP, our platform needs precise control over the timing of wireless transmissions. Because our platform system is implemented by overriding an implementation of the 802.11 MAC layer provided by a commercial family of networking cards, it is important to understand the key attributes of the 802.11 MAC and PHY layers and how they can help and hinder this overall goal: a) The PHY and MAC layers have checksums, and any failure in those checksums causes the message to be ignored; b) The MAC protocol is controlled by a series of precise timing intervals; c) Contention is handled by a combination of carrier sensing and collision avoidance using specified transmission durations contained in message headers.

Commodity 802.11 hardware typically divides up the functionality of the 802.11 MAC between the hardware/firmware on the card and the driver running on the host system. This means that the flexibility of such systems varies greatly among manufacturers. Based on the features of the *Atheros* (AR5212) chipsets, the open-source *Madwifi* driver uses the Hardware Abstraction Layer (HAL) to take control over the radio hardware.

Overall, there are six primary tasks we need to perform in order to implement our platform:

- Control the timing of transmission;
- Control retransmission;
- Control backoff procedure;
- Add a new SEQ-LBP header;
- Transaction of control messages;
- Microsecond timing precision;

To control the timing of transmission, we need to eliminate the RTS/CTS exchange, virtual carrier sense, automatic ACK and retransmission from the legacy driver and hardware, which are done by direct register setting. This is to let receivers reply ACK/NACK frames immediately upon receiving a SEQ frame. We still operate the card in normal mode, neither monitor mode nor promiscuous mode. This is to implement a usable driver level SEQ-LBP for real applications. We modify the *madwifi* driver and implement a user-defined retransmission and backoff procedure. Furthermore, we will in the same measurement setup determine loss probabilities for identical frames being transmitted at the same time. Without direct hardware access, this is possible only when the cards are modified to reply with an ACK frame at the same time. Thus obtained results will reveal the feasibility of feedback jamming.

7.2. Multicast Management and Dynamic Leader Selection

7.2.1. Design and Implementation

To maintain a multicast group (we assume there is only one multicast group in the test, and it is straightforward to extend this to multiple groups.), we need to handle dynamic member joining and member leaving. Moreover, a dynamic member status collection is also needed for leader selection. For wireless LANs, where the AP is de facto the center of all members, a centralized algorithm is appropriate and efficient. The multicast management and leader selection for SEQ-LBP are shown in Figure 7.1 and are explained as follows.

The AP broadcasts beacon frames (LBP_BN) periodically. Each receiver replies a LBP_JOIN frame to join the multicast group. When receiving a LBP_JOIN frame, the AP records its channel status (RSSI value), and chooses the one with the lowest average RSSI value as the leader. Then the AP broadcasts a LBP_LA or LBP_NLA to announce the new member as a leader or non-leader respectively. To enhance reliability, the AP sends multiple of these frames (e.g. double in tests). All receivers update their roles based on the announcements.

The AP broadcasts a LBP_LP frame to probe members' status if there are some members which have been inactive for a certain duration (e.g. 5s in tests). When receiving a LBP_LP frame, each receiver replies multiple (e.g. double in tests) LBP_LR frames at a random time. The AP updates the receiver's channel status and chooses the one with the lowest average RSSI value as the leader. If the new leader is different from the old one, the AP broadcasts a few LBP_LA frames to announce the new leader.

Receivers leave the multicast group by sending a LBP_QUIT frame to the AP. The AP eliminates a receiver when receiving a LBP_QUIT frame from it or it has been inactive for a certain time (e.g. 20s in tests). The AP selects a new leader if the old leader quits.

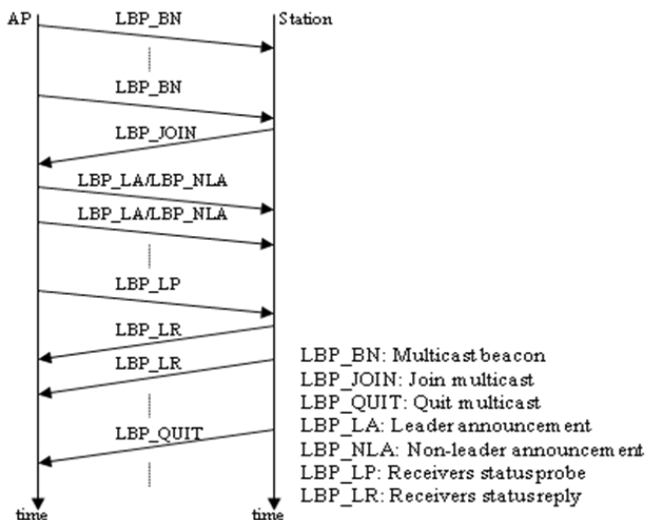

Figure 7.1: Multicast Management and Dynamic Leader Selection

7.2.2. Test Results

Now we evaluate the performance of our dynamic leader selection algorithm. The curves of the dynamic average RSSI per second for two members and four members groups are shown in Figure 7.2 and Figure 7.3 respectively. The worst scenario is considered where all receivers are the same distance away from the AP. The leader could be either receiver dynamically. From the curve, we can see that the dynamic leader selection algorithm works fine, and the receiver whose ACK/NACK frame has the lowest average RSSI is selected to be the leader dynamically for almost all the time.

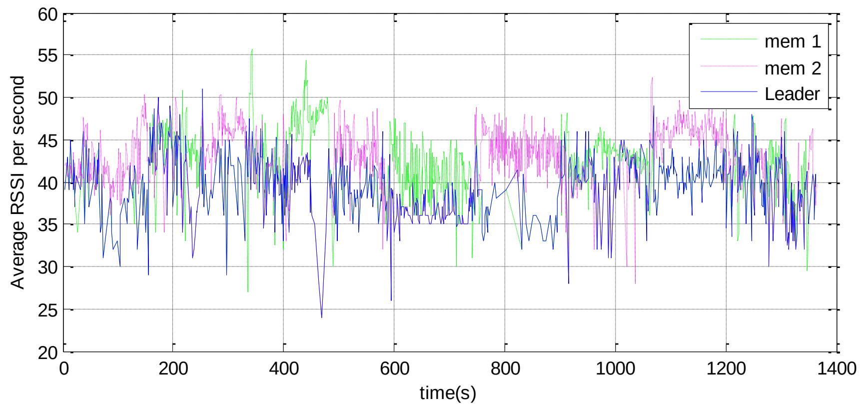

Figure 7.2: Average RSSI with the dynamic leader selection result I (the worst case, 2 members)

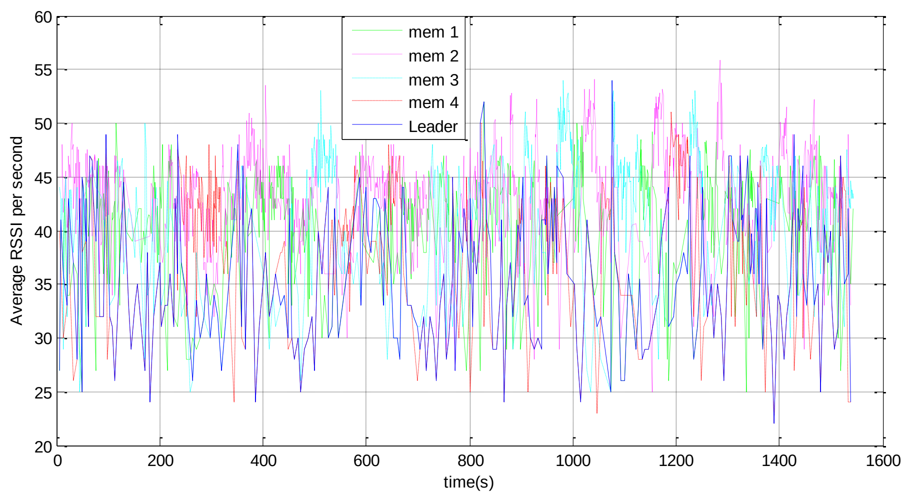

Figure 7.3: Average RSSI with the dynamic leader selection result II (the worst case, 4 members)

There are also other kinds of leader selection/election mechanisms. One simple scheme is to change the leader uniformly among all receivers, which can be based on the SEQ frame from the AP for example. This scheme is simple and works fine for more than two receivers. However, it is not the most efficient because the leader may e.g. be very close to the AP. Another kind of leader selection is a distributed algorithm. Each receiver randomly calls to be the leader by sending an announcement packet with its RSSI value. Other receivers (who may have a lower RSSI value) send announcement packet to compete for the leader. The station with the worst channel condition is agreed upon as the leader. This scheme works fine with the dynamic joining and leaving of group members. However, the direct station-to-station communication is not supported for all wireless networks, such as wireless LANs. For those, all traffic flows via the AP and a centralized algorithm is simpler and more efficient.

7.3. Evaluation of the ACK/NACK Jamming[9]

7.3.1. Test Topology

We evaluate the ACK/NACK jamming in both driver level and hardware level. We refer to driver level LBP frames as LBP-(N)ACK or software SW-(N)ACK as opposed to hardware (HW) ACK. Furthermore, we will in the same measurement setup determine loss probabilities for identical frames being transmitted at the (with consumer hardware most accurately possible) exact same time. Without direct access to the wireless LAN card physical layer firmware, we have found this is possible only when all stations are forced to reply with a HW-ACK after a SIFS for the same preceding data frame. In our test-bed this is done by setting the stations' MAC addresses to the same value (MAC address deceit). In order to evaluate the worst-case HW feedback jamming probability by transmitting most robust, maximally short yet non-identical (leader vs. non-leader) frame, we use another property of 802.11: a station can reply an ACK at a physical layer modulation and code rate that is less than or equal to the immediately preceding data frame rate[10]. Consumer wireless

[9] This section is a joint work with Jochen Miroll, Telecommunications Lab, Saarland University, Germany.

[10] For the exact definition, refer to 9.6 (multirate support) in [IEEE07]

LAN cards manufactured by *Atheros* can be set to make use of this feature or instead transmit the ACK at the lowest PHY rate[11].

Both the hardware feedback and driver level feedback (longer, less robust and less timing accurate) are evaluated. These relationships regarding our test setup are depicted in Figure 7.4. The AP transmits LBP SEQ frames at 12Mbps periodically to a unicast destination MAC address. The leader replies both a HW-ACK and a driver level LBP ACK with the same data rate of 6Mbps, where the former is supported by wireless LAN cards while the later is supported by our test-bed as describe above. The other receivers, deceived to use the same MAC address as the leader, reply HW-ACKs at 12Mbps and also reply a driver level LBP NACK frame at 6Mbps. The LBP NACK frames from different stations are slightly different in content (the last 3 bytes are different). Due to this, their increased length and the fact that the driver level response time jitter is far above the HW level timing accuracy, we assume that LBP NACK frames will never contribute to each other's received signal strength.

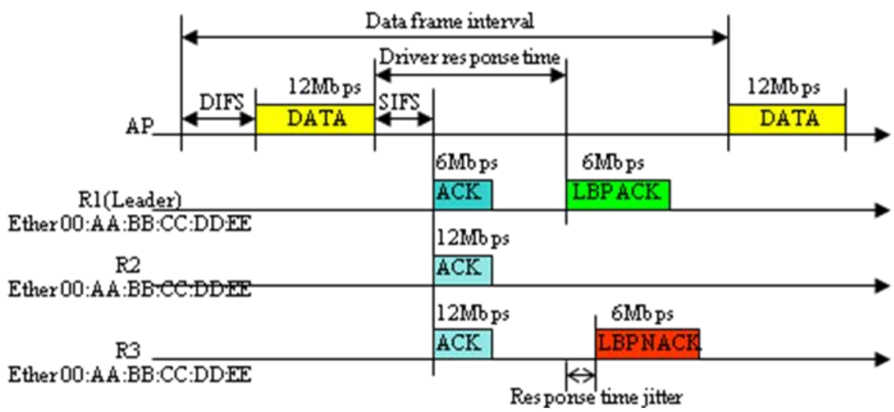

Figure 7.4: Test Diagram of Feedback Jamming

Most of our tests are under the worst scenarios that all receivers experience nearly the same channel condition and the receiving powers of ACK/NACK frames are roughly the same. The test topology is shown in Figure 7.5. All receivers are positioned on a circle while

[11] The lowest PHY rate in our tests always is BPSK, FEC ½, i.e. 6 Mbps

the AP is located at the center of the circle. The diameter of the circle is about 8 meters. During the test, all receivers change their location uniformly on the circle. The role of each receiver (leader, or non-leader) also changes periodically (per 100 seconds). This is to make sure all transmitted frames experience nearly the same channel condition on average over the complete measurement run.

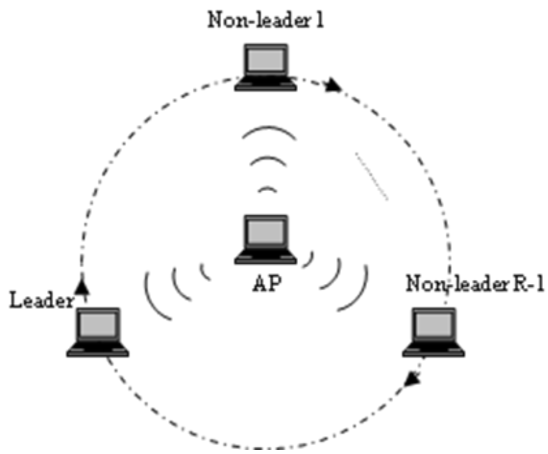

Figure 7.5: Test Topology for identical channels: receivers' locations are permuted with approx. constant distance to the AP

7.3.2. Experimental Results

We measure the feedback jamming probabilities with and without the dynamic leader selection algorithm in different scenarios: over an independent channel, over a shared channel, in an anechoic chamber, over a loud shared channel with people walking around, etc. Each simulation case runs for at least one hour. The results are presented and explained as follows.

7.3.2.1. Over an Independent Channel

We first test the feedback jamming probabilities over a 5GHZ channel, 802.11a, which is an independent channel in our test environment. Table 7.1 shows the results for normal scenarios with two receivers which have different distances to the AP. The results show that, with dynamic leader selection, the ACK/NACK jamming probabilities can be as high as 0.999+ in normal scenarios. The feedback jamming probability is about 0.90 in the worst case that both receivers experience nearly the same channel conditions and the receiving powers of ACK/NACK frames are roughly the same. As expected and observed, the receiving powers of the ACK/NACK frames determine the feedback jamming probability.

Table 7.1: ACK/NACK jamming probabilities in different scenarios (AP-mem1 10m, 802.11a 5 GHZ, with DLS)

Measurement parameter	**Distance of mem2 from the AP**			
	1m	**4m**	**7m**	**10m**
LBP ACK loss	0.999951	0.996433	0.999045	0.846425
LBP NACK loss	0.004044	0.419478	0.601006	0.509594
HW-ACK (6mbps) loss	0.999854	0.999604	0.999547	0.913178
HW-ACK (12mbps) loss	0.003069	0.020855	0.029115	0.428555
LBP SEQ loss at mem1	0.000000 RSSI 30	0.000000 RSSI 37	0.000000 RSSI 38	0.000266 RSSI 42
LBP SEQ loss at mem2	0.000000 RSSI 52	0.000000 RSSI 46	0.000000 RSSI 41	0.000151 RSSI 46

We test further for the worst case that all receivers experience nearly the same channel conditions and the receiving powers of ACK/NACK frames are roughly the same, shown in Figure 7.5. Table 7.2 and Table 7.3 show the experiment results with two and four members respectively. We can see that the dynamic leader selection algorithm increases the jamming probability dramatically: from 0.69 to 0.86 for two members group and from 0.88 to 0.97 for four members group.

Table 7.2: Test results for the worst case I (802.11a 5GHz, 2 members, with or without DLS)

Measurement parameter	**Role Change**	**DLS**
LBP ACK loss	0.711514	0.886956
LBP NACK loss	0.714842	0.517801
HW-ACK (6mbps) loss	0.685121	0.858483
HW-ACK (12mbps) loss	0.614805	0.402463

Measurement parameter	Role Change	DLS
LBP SEQ loss at mem1	0.000121	0.000331
LBP SEQ loss at mem2	0.000000	0.000056

Table 7.3: Test results for the worst case II (802.11a 5GHz, 4 members, with or without DLS)

Measurement parameter	Role Change	DLS
LBP ACK loss	0.894134	0.992322
LBP NACK loss	0.753818	0.683027
Hardware ACK-6 loss	0.883892	0.967472
Hardware ACK-12 loss	0.864081	0.780548
LBP SEQ loss at station 1	0.000137	0.000000
LBP SEQ loss at station 2	0.000168	0.000063
LBP SEQ loss at station 3	0.000246	0.000058
LBP SEQ loss at station 4	0.000138	0.000176

7.3.2.2. Over a Shared Channel

In this subsection, we present the test results over a 802.11g channel, 2.4GHz, which is shared with the campus wireless LAN in our test environment. The test topology in Figure 7.5 is used. Moreover, the role of leader is changed frequently (per 2s) and uniformly among all four receivers. From the test results shown in Table 7.4, as expected, we observe that the feedback jamming probability is higher than the one over an independent channel. The feedback jamming probability is as high as 0.95 without dynamic leader selection. This is because the SNR of the ACK frame is lower in a loud environment with interference from other networks, and hence it is easier to be destroyed.

Table 7.4: Test results for the worst case III (802.11a 5GHz and 802.11g 2.4GHz, 4 members, role change)

Measurement parameter	802.11a 5GHz	802.11g 2.4GHz
LBP ACK loss	0.894134	0.974267
LBP NACK loss	0.753818	0.899021

Measurement parameter	802.11a 5GHz	802.11g 2.4GHz
Hardware ACK-6 loss	0.883892	0.951087
Hardware ACK-12 loss	0.864081	0.910189
LBP SEQ loss at station 1	0.000137	0.006039
LBP SEQ loss at station 2	0.000168	0.003828
LBP SEQ loss at station 3	0.000246	0.005725
LBP SEQ loss at station 4	0.000138	0.004993

7.3.2.3. In an Anechoic Chamber

We also test the feedback jamming probability in an anechoic chamber, where there is no signal reflection from walls, floor or ceiling. Two receivers are located with the same distance to the AP and the role of leader changes frequently (per 2s) and uniformly between two receivers. The result is shown in Table 7.5. Contrast to the shared channel, we can see that the feedback jamming probability is quite low (about 0.55) in an anechoic chamber. This is due to high SNR of the ACK frame with low interference.

Table 7.5: Test results for the worst case IV (Anechoic chamber, 802.11a 5GHz, 2 members, role change)

Measurement parameter	Rate
Hardware ACK-6 loss	0.542427
Hardware ACK-12 loss	0.542293
LBP SEQ loss at station 1	0.000026
LBP SEQ loss at station 2	0.000132

7.3.2.4. Over a Loud Shared Channel

We then measure the feedback jamming probabilities in an extreme scenario with a shared 802.11g 2.4GHz channel and people walking around (in a small party). One test case is shown in Table 7.6 with two receivers, using the topology shown in Figure 7.5, and the role of leader is changed frequently (per 2s) and uniformly between two receivers. From the results shown in Table 7.6, it is observed that the feedback jamming rate is quite high (0.90) even without the dynamic leader selection algorithm. This is because the SNR of the ACK frame is lower in a loud environment with high interferences, and hence it is easier to be destroyed.

Table 7.6: Test results for the worst case V (802.11g 2.4GHz, 2 members, role change)

Measurement parameter	Rate
Hardware ACK-6 loss	0.903905
Hardware ACK-12 loss	0.906685
LBP SEQ loss at station 1	0.004366
LBP SEQ loss at station 2	0.004776

In the same test environment, the other case contains four receivers, using the same test topology but with the dynamic leader selection algorithm. The result is shown in Table 7.7. As expected, we can see that the feedback jamming probability is as high as 0.99 because of the dynamic leader selection and the larger group size.

Table 7.7: Test results for the worst case VI (802.11g 2.4GHz, 4 members, DLS)

Measurement parameter	Rate
Hardware ACK-6 loss	0.988859
Hardware ACK-12 loss	0.989130
LBP SEQ loss at station 1	0.004306
LBP SEQ loss at station 2	0.000206
LBP SEQ loss at station 3	0.000302
LBP SEQ loss at station 4	0.000292

From all the test results, we can see that the feedback jamming probability depends on the receiving power of ACK/NACK frames and the channel conditions. As a result, smart control of the transmitting power can achieve high feedback jamming probability, e.g. low transmitting power for the ACK frame and high transmitting power for the NACK frames. The dynamic leader selection can increase the feedback jamming probability greatly. Moreover, the multicast management (including DLS) overhead is about 1% for two members groups and about 2% for four members groups in the worst case that all members move at 1mps.

In summary, we conclude that, for the first time, the hardware ACK/NACK jamming works well and can be used for the MAC layer multicast in IEEE 802.11 wireless LANs. With similar hardware supports, the feedback jamming could also work in other wireless networks, which are out the scope of this paper.

7.4. Evaluation of SEQ-LBP

7.4.1. Architecture and Implementation

Our implementation here aims to replace the legacy 802.11 MAC broadcast with SEQ-LBP in the driver level. Figure 7.6 shows the architecture of the driver level SEQ-LBP (for both AP and Stations). Please note that the wireless LAN card still runs in normal mode. This is not like other similar platforms which have to run in monitor mode (softMAC) or promiscuous mode (FlexMAC). We only need to configure the hardware to send SEQ-LBP control frames immediately at multicast receivers.

In both the AP and receivers, three modules are added on top of the original *Madwifi* driver. A driver level queue is kept on the host to buffer incoming packets from the kernel. The *transmission controller* prepares the next packet to be transmitted, which can be a SEQ-LBP data packet, a SEQ-LBP retransmission data packet, a SEQ-LBP control frame, or a head-of-line unicast packet in the packet pool, depending on the logic of the protocol. The transmission controller posts the transmission request to the frame scheduler and waits for the completion of the transmission. The transmission controller allows at most one packet in the hardware queue and instead keeps a queue on the host to buffer incoming packets from the kernel. The *frame scheduler* is responsible for delivering data packet to the hardware based on the schedule of transmissions specified by the transmission controller. At the AP, both SEQ-LBP packets and original unicast packets use the original hardware functions. At the receivers, the hardware is turned to transmit driver level packets immediately, in other words the RTS/CTS exchange, virtual carrier sense, automatic ACK and retransmission are disabled. The broadcast packets are sent following SEQ-LBP while the unicast packets are sent using the original functions. All SEQ-LBP frames are handled in the driver layer. The *frame dispatcher* sends packets that are destined for the station itself to the kernel, as in the original *Madwifi* driver. It passes the control messages to the transmission controller where they can be handled properly.

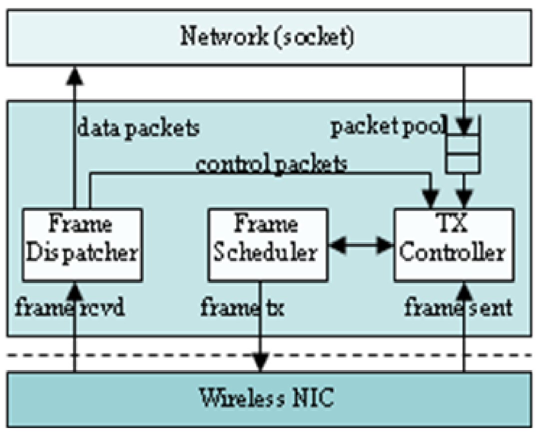

Figure 7.6: Architecture of the driver level SEQ-LBP

2	2	4
Type	Mem id	SEQ

Figure 7.7: SEQ-LBP header format

In our implementation, a header is added to all SEQ-LBP frames in the drivel level, which includes three fields: frame type (2 octets), member id (Mem id, 2 octets) and sequence number (SEQ, 4 octets), shown in Figure 7.7. The type field denotes different packet types in SEQ-LBP, shown in Table 7.8. The member id is used to identify packets from each station which is fundamental for multicast membership management. Each receiver is allocated a unique member id manually or based on its own MAC address. The sequence number is a 4-octets field, which is used for indicating the data packet and also for multicast loss detection.

Table 7.8: SEQ-LBP frame types

Nr.	Type name	Comments
1	LBP_SEQ	SEQ packets
2	LBP_DATA	Data packets
3	LBP_ACK	ACK packets
4	LBP_NACK	NACK packets
5	LBP_BN	Multicast beacon packets
6	LBP_JOIN	Multicast join request packets
7	LBP_QUIT	Multicast quit packets
8	LBP_LA	Multicast leader announcement packets
9	LBP_NLA	Multicast non leader announcement packets
10	LBP_LP	Status probe packets for leader selection
11	LBP_LR	Status reply packets for leader selection

The driver level SEQ-LBP is according to the protocol shown in Figure 4.6 but is a little different from the hardware version, such as no RTS-CTS exchange, different header format etc. The flow diagrams in both AP mode and station mode are shown in Figure 7.8 and Figure 7.9 respectively. A driver level queue is kept on the host to buffer incoming packets from the kernel. The protocol works as follows.

a. [AP] Keep the incoming network packets in a driver queue;
b. [AP → Receivers] Check the driver queue: If the head packet of the queue is a unicast packet, send it down to hardware level for transmitting, and go to Step F. If the head packet is a broadcast packet, go to Step C.
c. [AP → Receivers] Send a LBP_SEQ frame down to the hardware level for transmitting. The LBP_SEQ frame carries the sequence number of the following data packets and reserves the channel for it. The AP encapsulates the data packet (adding a LBP Header in the tail) and sends it out following the SEQ frame after a SIFS. The AP starts a timer to wait for ACK/NACK frames when the transmitting of the data frame is over.
d. [Receivers → AP] Both leader and non-leader receivers trigger a timer upon receiving a LBP_SEQ frame. When the timer expires, the leader replies an ACK

frame if the data is correct or has been received correctly before based on sequence check. When the timer expires, non-leader receivers reply an NACK frame if the data has never been received correctly before based on sequence check.

e. [AP] If an ACK frame is received, or the retry limit is reached, the transmission is complete, go to Step B for the next packet; If no ACK frame has been received and the retry limit has not expired, go to Step C for a retry.

f. [AP] When a unicast transmission is complete (informed from the hardware level), go to Step B for the next packet.

g. [Receivers] When a transmission is complete (informed from the hardware level), check its driver level queue to send the next packet.

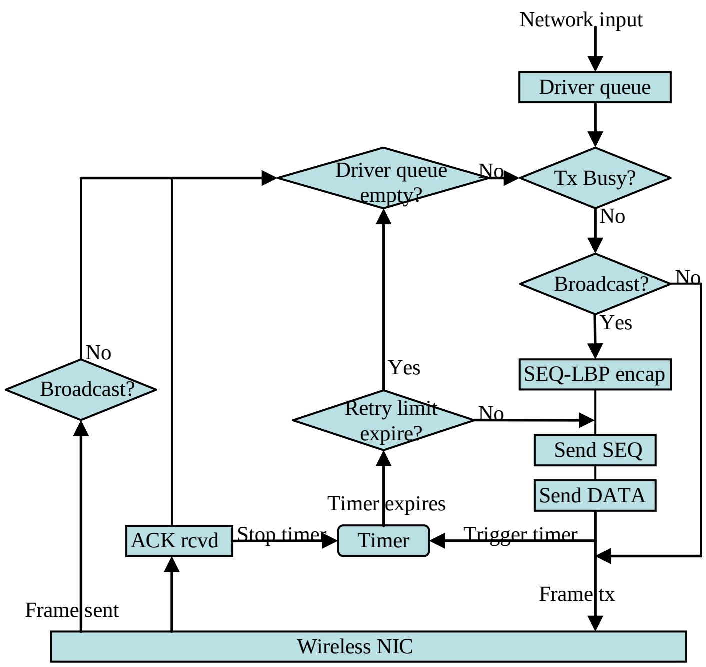

Figure 7.8: SEQ-LBP diagram on AP mode

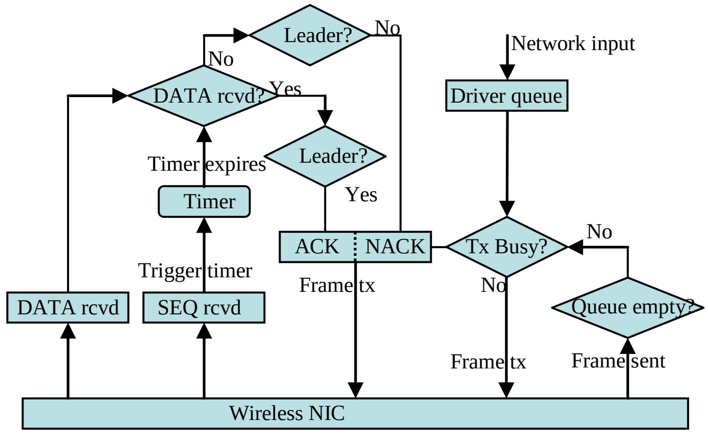

Figure 7.9: SEQ-LBP diagram on station mode

7.4.2. Experimental Results

The performance of the driver level SEQ-LBP is evaluated in this subsection. To compare the performance of SEQ-LBP with the legacy MAC broadcast, both SEQ-LBP and MAC broadcast are tested at the same time as follows. When the sequence number is even, the LBP SEQ and LBP DATA are transmitted using the legacy MAC broadcast. Both the leader and non-leader receivers set a timer and reply an ACK or NACK respectively when the timer expires no matter it does receive a LBP DATA or not. This case is used to measure the ACK/NACK jamming probability and the path loss (raw loss without retransmissions) rate of LBP SEQ and LBP DATA frames. When the sequence number is odd, the driver level SEQ-LBP is performed and the final PLR of LBP SEQ and LBP DATA frames are measured. The parameters used in the tests are shown in Table 7.9.

Table 7.9: Parameters of SEQ-LBP experiment

Parameters	**Values**	**Parameters**	**Values**
LBP SEQ length	44Bytes	Receiver TXPOWER	8dBm
LBP ACK/NACK length	200Bytes	Data load interval	0.1s
LBP DATA length	1028Bytes	Number of data per test	20000
Date rate of LBP Data	24Mbps	Data rate of LBP control and mng. frames	6Mbps
Retry limit	7		

The test results for the multicast group with four members under 802.11a and 802.11g mode are shown in Table 7.10 and Table 7.11 respectively. As described before, in our test environment, 802.11a is an independent channel while 802.11g is a shared channel with the campus wireless LANs. From the results, we can see that SEQ-LBP recovers the multicast packet losses dramatically: about 70% - 90% errors have been corrected. However, the errors cannot be corrected completely, especially at non-leader receivers. There are two main reasons. One is that the ACK/NACK jamming probability is not 100% and sometimes the NACK fails to destroy the ACK as a result of which no retransmission is prompted. The other reason is that the driver level SEQ-LBP has a relatively rougher time scale and channel management than the hardware version. Hence the LBP SEQ is not reliable and the LBP NACK cannot be triggered when the LBP SEQ is lost. Please note that the packet loss rates are very high when the test runs in 802.11g mode. This is due to the inferences from other wireless networks (campus wireless LAN) on the same channel. As a result, some of the LBP DATA frame cannot be recovered due to the high loss of LBP SEQ. The hardware version of SEQ-LBP with a higher time precision will have a better performance. Moreover, RTS-CTS exchange could relieve this SEQ loss problem which is considered as our future work.

Table 7.10: 802.11a 5GHz SEQ-LBP test result

	SEQ Path Loss	**Data Path Loss**	**SEQ Loss Final**	**Data Loss Final**
Mem1	0.000000	0.001271	0.000000	0.000282
Mem2	0.000000	0.000808	0.000000	0.000303
Mem3	0.000000	0.000853	0.000000	0.000426
Mem4	0.000000	0.000878	0.000000	0.000000
AP	ACK/NACK Jamming rate: 0.998753		Redundancy: 0.061346	

Table 7.11: 802.11g 2.4 GHz SEQ-LBP test result

	SEQ Path Loss	**Data Path Loss**	**SEQ Loss Final**	**Data Loss Final**
Mem1	0.008252	0.066149	0.005995	0.003916
Mem2	0.003008	0.028968	0.002658	0.002044

	SEQ Path Loss	Data Path Loss	SEQ Loss Final	Data Loss Final
Mem3	0.008210	0.065577	0.007852	0.005471
Mem4	0.013417	0.059629	0.011006	0.002592
AP	ACK/NACK Jamming rate: 0.999720		Redundancy: 0.311648	

7.5. Conclusion

In this chapter, we evaluate the feedback jamming on a real test-bed built using commodity wireless LANs hardware. By long-time tests (each one lasts for several hours) under various scenarios, we found that the hardware ACK/NACK jamming probability can be as high as 0.99+ for normal scenarios (about 0.90+ for the worst case with only two receivers which even experience nearly the same channel conditions) when a simple dynamic leader selection algorithm is used. These results confirm the feasibility of ACK/NACK jamming.

We also implement a driver level SEQ-LBP with dynamic leader selection & multicast management on the test-bed and confirm its performance for recovering multicast packet losses. Based on the results we assume that a hardware SEQ-LBP implementation, if incorporated directly into the wireless modem and thus with more precise timing, is an effective and efficient MAC layer multicast ARQ mechanism. Our driver level SEQ-LBP can replace the normal MAC broadcast in the *Madwifi* driver and provide a MAC layer multicast ARQ for real applications.

Chapter 8

Accessory Techniques

In this chapter, we will talk about some accessory techniques for the proposed feedback jamming based protocols. We first present the experiment results of NACKs aggregation through a pure NACK jamming based MAC layer multicast approach. Then a MAC layer FEC coding with a fine granularity is discussed. Moreover, we will also talk about the potential data rate adaptation mechanisms for the feedback jamming based protocols. Finally, the potential cross layer mechanisms will be discussed as well.

8.1. Pure NACK Jamming

Based on the implementation of the driver level SEQ-LBP, we evaluate a pure NACK jamming based ARQ scheme (shown in Figure 8.1) and feedback collision detection in this section. There are two differences from the driver level SEQ-LBP: 1) All receivers reply driver level NACK at the same time when the data packet is lost (to answer the SEQ packet). 2) The AP retransmits the data packet when a collision is detected or an NACK is received in the duration of NACK time slot. All packets are handled in the driver level. Retransmission is allowed only when the sequence number is odd while it is used to test the path losses when the sequence number is even. Two receivers are positioned the same distance away from the AP, about 5meters. The AP and receivers all are kept in stationary. We use the same parameters as for the driver level SEQ-LBP, shown in Table 7.9.

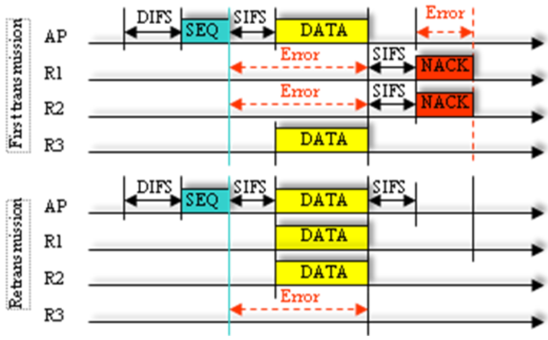

Figure 8.1: Pure-NACK ARQ

The feedback collision detection is through hardware-register-reading and it is based on threshold check in hardware. The performances of pure-NACK ARQ in 802.11a and 802.11g mode are shown in Table 8.1 and Table 8.2 respectively. From the results, we first observe that the residual error rates at each receiver are still high. As discussed in Chapter 4, this is because of the fake positive feedback detection, in particular receivers cannot reply feedback when both the SEQ and DATA frames are completely destroyed (e.g. due to interferences), in which situation the sender will detect a clean channel in the feedback time slot and treat it as a successful transmission. The feedback collision detection works well in 802.11a mode which has been a clean channel (no sharers) in the test environment. However, there are a lot of fake detections in 802.11g mode which is a busy channel shared with the campus wireless LAN. Fake collision detection causes unnecessary retransmissions and hence the transmission redundancy is very high.

In summary, two limitations of the pure-NACK based multicast have been confirmed: 1) The data errors cannot be recovered when both the SEQ and DATA are lost at the same time, e.g. due to severe interferences. As a result, the residual error rates are still very high (much higher than the ones in the ACK/NACK jamming based schemes). 2) The channel status of clean or collision cannot be perfectly detected and distinguished. Smart threshold choice may relieve this problem, which needs specially hardware supports and is considered as our future work. The fake detection leads to high residual error rates and unnecessary retransmissions. The busy-tone based and physical layer subcarrier based schemes (see

Chapter 2) suffer from similar limitations as well. These problems should be considered for the design of MAC layer multicast and cross MAC and Physical layer multicast for wireless networks.

Table 8.1: 802.11a 5GHz pure-NACK ARQ test result

	SEQ Path Loss	Data Path Loss	SEQ Loss Final	Data Loss Final
Mem1	0.008532	0.008776	0.009282	0.008518
Mem2	0.014546	0.014546	0.014296	0.014050
AP	NACK loss rate: 0.184101		Redundancy: 0.001245	

Table 8.2: 802.11g 2.4 GHz pure-NACK ARQ test result

	SEQ Path Loss	Data Path Loss	SEQ Loss Final	Data Loss Final
Mem1	0.025883	0.078622	0.021046	0.020246
Mem2	0.020849	0.031373	0.019150	0.018512
AP	NACK loss rate: 0.389393		Redundancy: 1.243237	

8.2. Fine Granularity MAC HEC

To further improve the performance of multicast error control in the MAC layer, especially to further shorten the multicast delay, FEC coding can perform in a flexible granularity instead of just in the packet level (e.g. HLBP). As described in previous chapters, current wireless MAC protocols (e.g. IEEE 802.11) are designed for reliable data transmissions and all these error control schemes are based on the packet level, which means that even one bit error (residual errors which have not been corrected by the PHY codes) in a packet could result in the whole packet being dropped in the MAC layer, which is a huge waste of wireless channel resources.

One possible approach to reduce the error recovery cost is to pass the damaged packets up to upper layers, e.g. the application layer, which of course must be capable of detecting errors and utilizing the partially damaged data (using coding techniques). Many researchers proposed cross-layer error control schemes [Wel05], [Kor07] for both unicast and multicast based on UDP-Lite [Lar04], which uses a partial checksum to cover only the packet header and the critical data of the payload located in the beginning of the packet. If a bit error is detected in the protected part, the whole packet is discarded. Otherwise, it is passed further up to the application layer. To support protocols like UDP-Lite, the link layer has to shut down its own error recovery schemes (e.g. FEC and ARQ) and pass the damaged packets to upper layers. However, with respect to the independence of each layer, passing the damaged packets to upper layers and forwarding them among wireless stations/clients is not a perfect approach. We prefer to handle this in the MAC layer.

An intuitive MAC layer approach to reduce the error recovery overhead for residual bit errors is the bytes level FEC such as the one in [Cho06], where the MAC header is encoded by a (40, 24) RS code [non08b] and the MAC payload is split into multiple blocks, which are encoded using a (255, 239) RS code. And when the errors cannot be recovered by the FEC, MAC layer retransmission is also used. Although it is effective to correct the bit errors, MAC layer bytes-level FEC causes fixed overhead even under good channel conditions. Furthermore, the existing MAC layer FEC schemes (with retransmissions) are only for unicast.

In this section, we discuss flexible block erasure codes in the MAC layer, called fine granularity HEC. A MAC Protocol Data Unit (MPDU) is packetized into k segments. A block erasure code (n,k) is used to generate $n-k$ parity segments from the original k data segments. The header is encoded with a bytes-level FEC. The k data segments and a certain amount of parity segments are remerged and transmitted in the first transmission. Different frames consisting of parity segments are transmitted in each retransmission if necessary. Combined with the ACK/NACK feedback jamming scheme, MAC layer erasure code can also be used for MAC layer multicast.

The frame formats are shown in Figure 8.2. A L bytes data frame in the MAC layer is divided into k segments with a length s where $k = \lceil L/s \rceil$. Each segment also includes a 4-octets CRC field. A (n,k) block erasure code is used to convert the original k data segments into a block of n encoded segments: k original data segments and $n-k$ parity segments. The new header has three new fields: segment length, number of segments and segment start index. A byte level (46, 30) RS code, which is a shortened RS code, is used for the header. Note that the encoded header can correct up to 8-bytes errors. This is to enhance the reliability of the header, which is more important than the following data field. So the whole data frame includes an encoded header, $k+c$ segments (k data segments and c parity segments) and a 4-octets CRC field. Moreover, different $k+c$ parity segments are used in each retransmission. Note that the outer FCS allows the receivers to skip the segment FCS checks if the outer FCS is correct.

Frame Control	Duration	RA	TA	BSSID	Sequence Control	Frame Body	FCS
2	2	6	6	6	2	0-2308	4

(a)

MAC Header		Data Segments			Parity Segments			FCS
Header	FEC	Seg 1	...	Seg k	Seg k+1	...	Seg k+c	FCS
30	16	s+4	...	s+4	s+4	...	s+4	4

Old Header	Seg num	Seg Len	Start Index
24	2	2	2

Seg Data	FCS
s	4

(b)

00 01	02 03	04 07	08	09	10	11	12	13	14	15
Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgt	More Data	WEP	Order

(c)

Figure 8.2: IEEE 802.11 MPDU format without and with erasure code: (a) Original MPDU, (b) MPDU with erasure code, and (c) Frame Control field in MAC header

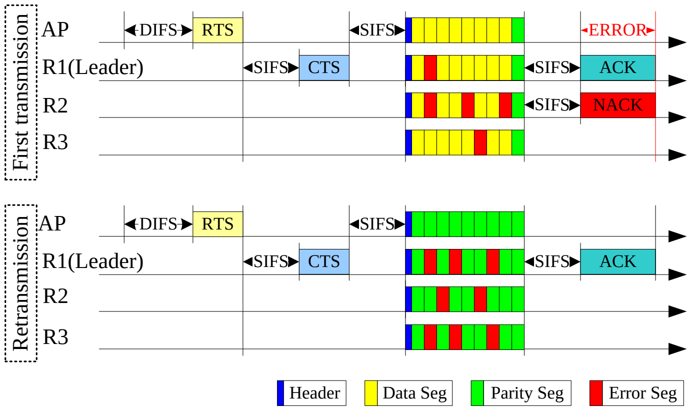

Figure 8.3: Fine Granularity MAC HEC

The protocol is shown in Figure 8.3. The k data segments and c parity segments are transmitted in the first transmission. The leader receiver replies an ACK if at least k segments are received correctly. Each non-leader receiver replies a NACK if the total correct segments is less than k . $k+c$ different parity segments are transmitted in each retransmission if necessary. Similar to HARQ Type III [Sol3], each retransmission packet in the fine granularity MAC HEC is also self-decodable which is good for the case that the transmitted data is lost at the first transmission or seriously damaged by noise. Note that as the encoded header is very reliable, it is not needed to use the SEQ frame like in SEQ-LBP anymore. Please also note that here unicast is just a special case of multicast with a single receiver. Intuitively, the proposed protocol can correct the errors for all receivers due to the feedback jamming scheme and retransmissions in the MAC layer. It achieves complete feedback suppression thanks to the feedback jamming scheme and has a high scalability with respect to the size of multicast group due to the FEC coding.

In practice, the parity segments can be generated in advance before the data transmission. As the reply of NACK is only based on CRC check and the number of correct segments, the FEC decoding of a block can be started after any k correct segments have been received. As a result, the segment level FEC in the MAC layer can meet the time-critical requirements of the MAC layer protocols.

Compared with packet level error correction schemes in the MAC layer, this approach exploits erroneous packets, takes a much shorter time and is better to be combined with application layer error control schemes, especially for multicast transmissions. This approach fully utilizes the one-hop/local wireless transmissions for both unicast and multicast and is a good complement to Network Coding [Ahl00] which deals with end-to-end transmissions using coding. Moreover, for 802.11n or future wireless LAN protocols with higher data rates, it is more efficient to use large data frames and to merge small data packets into a large one, in which case the legacy DCF protocols will be less efficient while the MAC layer coding based ones could be good candidates. Please also note, different from SEQ-LBP and HLBP, this protocol is not compatible to legacy IEEE 802.11 stations as the frame formats have been changed significantly. As a result, in this thesis, we do not explore this MAC layer coding with a fine granularity any further and leave it as a future work.

8.3. Multicast Rate Adaptation

8.3.1. Background

As described in Chapter 1, it is a big challenge to support high rate real-time multimedia applications in wireless LANs. A component which is critical to the performance of a wireless link is the transmission rate adaptation mechanism. Rate adaptation mechanisms are not standardized in IEEE 802.11 wireless LANs, and each manufacturer chooses its own. To improve the throughput in wireless LANs, an AP needs to estimate the channel state so that it can select an appropriate transmission rate. Several rate adaptation mechanisms have been proposed and deployed for unicast transmissions. In a unicast scenario, an AP can determine the channel state of the receiving stations through a feedback mechanism such as RTS/CTS or the MAC layer ACK frame, and then adapt its transmission rate appropriately [Kam97][Hol01][Sad04]. In [Kam97], Kamerman and Monteban present the ARF protocol, Auto Rate Fallback, for IEEE 802.11, used in Lucent's WaveLAN II devices. The ARF protocol is the most known commercial implementation of rate adaptation for the IEEE

802.11 MAC. Under the ARF protocol, after the reception of ten consecutive ACKs, the next higher mode is selected for future data frames. If the delivery of the eleventh frame is unsuccessful, it immediately falls back to the previously supported mode. During other cycles with less than ten consecutive ACKs, it switches to a lower rate mode after two successive ACK failures. In [Hol01], Holland *et al.* proposed a RBAR (Receiver Based Auto Rate) protocol, which lets the receiver measure the channel quality and decide the transmission rate, and then inform it to the sender before the data packet transmission. Sadeghi *et al.* proposed an OAR (Opportunistic Auto Rate) protocol in [Sad04]. The major difference between OAR and RBAR is that OAR lets the sender send more packets when the channel quality is high.

However, the legacy broadcast/multicast has no feedback mechanisms because it is usually an unreliable one-to-many communication scenario. Therefore, unicast transmission rate adaptation mechanisms, such as ARF and RBAR, cannot be directly applied to multicast since they are based on the estimation of individual channel states. Most commercial APs use a fixed and relatively very low transmission rate for multicast, although more recent APs have included a manual configuration facility which enables an administrator to select a transmission rate. Even through IEEE 802.11a/b/g supports transmission rates up to 11Mbps or 54Mbps, multicast packets are often transmitted at a configurable basic rate (e.g., 1 or 2Mbps in 802.11b, 6Mbps in 802.11a/g). Such transmissions are a significant waste of wireless channel resources with a negative impact on the whole network [Ber03].

A few rate adaptation approaches have been proposed for multicast scenarios [Par06] [Che06]. Park *et. al* [Par06] proposed a rate adaptation scheme that improves throughput by utilizing periodic link level SNR feedbacks from clients. Using this feedback, the AP can collect SNR values for all stations participating in multicast groups and then determine the transmission rate for each group. Chen *et. al* [Che06] use unary channel feedbacks (UCF) and unary negative feedback (UNF) to estimate channel quality information. For each packet, the sender broadcasts a RTS and each receiver replies a UCF (or UNF if not wanting) at the same time slot but with different length. The sender then estimates the highest tolerable data rate supported by all receivers and uses it to send out (or forward) the data packet.

8.3.2. Proposals for SEQ-LBP and HLBP

We now consider rate adaption mechanisms for the feedback jamming based protocols: LBP, SEQ-LBP and HLBP. The ACK/NACK jamming based multicast feedback provides a unary feedback for the entire multicast group, similar to the feedback mechanism in unicast. So some rate adaptation mechanisms for unicast can be used for the feedback jamming based multicast protocols. For example, the ARF schemes can be used for LBP, SEQ-LBP and HLBP. We describe ARF for SEQ-LBP as follows. After the reception of ten (or a smaller threshold) consecutive ACKs, the next higher mode is selected for future data frames. If the delivery of the eleventh frame is unsuccessful (No ACK or receiving a NACK), it immediately falls back to the previously supported mode. During other cycles with less than ten consecutive ACKs, it switches to a lower rate mode after two successive ACK failures. Although, this ARF mechanism for multicast is very similar to the one for unicast, some new issues come forth, such as fairness issues. For example, the worst channel condition among receivers has a great impact on the rate selection at the sender. So the total multicast performance depends on the “worst” receiver which is the bottleneck of the multicast group. Smart QoS control or admission control can be performed at the sender or receivers to ignore or remove the worst receiver if necessary.

Apparently, the existing rate adaptation mechanisms for multicast [Par06] [Che06] can also be used for LBP, SEQ-LBP and HLBP. As the MAC layer multicast always requires multicast management in the MAC layer, the periodical exchange between the sender and each receiver is required definitely, and hence the sender knows the current channel status of each receiver and can choose an appropriate rate for data transmissions. Please note that for this kind of information collection, the more frequently the sender requests each receiver, the more accurate information the sender will get but with a larger overhead. Based on the experiments on multicast in wireless LANs, we believe that, different from unicast, the sender should access more fresh information about each receiver in the MAC layer to obtain a better multicast performance, especially for high rate real-time multimedia applications.

8.4. Cross-Layer Cooperation

As in a relatively smaller time range compared to the application layer approaches, the MAC layer multicast error correction cannot guarantee the final reliability because the wireless channel always experiences burst errors. So application layer multicast error control protocols are always needed to control the final QoS for multicast applications. Moreover, the MAC layer multicast can cooperate with upper layer approaches in order to obtain a better performance. The main potential cooperation approaches are described as follows.

One possible cooperation is the retry limit adaptation for MAC layer multicast protocols. The retry limit can be adjusted based on both the application requirements and the dynamic channel conditions. For example, the MAC layer retry limit can be smaller for applications with very strict delay constraints, in which situation, application layer FEC is a more appropriate approach. Meanwhile, if the channel error burst length is as large as the retry limit, the MAC layer multicast could use a smaller retry limit and leave the residual error correction to the application layer. Similarly, if the application with long delay constraints, the MAC layer multicast could use a larger retry limit and obtain a better overall performance (more efficient).

Another possible cooperation is about rate adaptation. As the application layer has the more detailed information about each receiver, especially the channel status, the MAC layer could share those information and choose an appropriate rate for data transmissions. Furthermore, the application layer can even measure or predict the channel condition of each receiver and share it with the MAC layer.

Moreover, for multi-hop multicast in wireless networks, the MAC layer multicast can share the multicast topology and other information with upper layers where the multicast is managed. Using this information, the MAC layer multicast can supply different services to stations depending on their roles in the multicast. For example, the MAC layer should provide more reliable delivery (e.g. using larger retry limit) for the branch/relay stations.

Furthermore, the cooperation could be based on the protocol itself. For example, in the HLBP approach, the packet level FEC coding could be performed in upper layers, in which way the burden of the MAC layer is relieved and the strict timing requirement is met. Moreover, the parameters of the FEC coding could be optimized dynamically based on application layer PLR measurements or prediction.

Based on the experiments on multicast in wireless LANs, we believe that, compared with legacy unicast/broadcast, the MAC layer should access more information about the application and more information about each receiver to obtain a better performance, especially for high rate real-time multimedia applications.

8.5. Conclusion

In this chapter, we talked about some accessory techniques for the proposed feedback jamming based protocols. We first presented the experiment results of a pure NACK jamming based MAC layer multicast approach and confirmed the problems of pure NACK aggregation. Then we discussed a MAC layer FEC coding with a fine granularity which could be a good candidate for the future wireless LANs with high data rates. Moreover, we also talked about the data rate adaptation mechanisms and cross-layer issues which could enhance the proposed feedback jamming based protocols.

Chapter 9

Conclusions and Future Work

This thesis focused on MAC layer multicast which outperforms the application layer ones with both shorter delays and higher efficiencies for high rate real-time traffic in IEEE 802.11 based wireless LANs. We first explored the potential feedback mechanisms for MAC layer multicast and confirmed the advantages of the feedback jamming scheme where both ACK from the leader receiver and NACKs from non-leader receivers are aggregated in the same time slot. Then we proposed two MAC layer multicast protocols, SEQ-LBP and HLBP, based on the feedback jamming scheme. We also analyzed the feedback jamming probabilities over the Rayleigh channel model and the theoretical performance of LBP, SEQ-LBP and HLBP over two channel models: the i.i.d channel model and the GE channel model. These performances were also confirmed by NS-2 simulations. Furthermore, we tested the ACK/NACK jamming probabilities in a real test-bed built with consumer wireless LAN hardware and confirmed its feasibility. A madwifi driver level SEQ-LBP was also implemented and tested in our test-bed. At last, we discussed some accessory techniques for the proposed feedback jamming based protocols: a pure NACK jamming based approach, a MAC layer FEC coding with a fine granularity, the potential data rate adaptation mechanisms and the related crossing layer issues.

The primary contributions of this thesis are depicted in the following section in details.

9.1. Contributions

- We explored the potential feedback mechanisms for MAC layer multicast in IEEE 802.11 wireless LANs. Besides the polling scheme which is very time-consuming, feedback (ACK or NACK) aggregation is a potential candidate. However, pure NACK aggregation has fake detection problems which cause high residual error rates or severe unnecessary retransmissions. The feedback jamming, which is the aggregation of an ACK and NACKs in the same time slot, avoids the fake detection problems and has an outstanding performance. We confirmed the feasibility of feedback jamming by theoretical analysis over a Rayleigh channel model and measurements in a real test-bed, which will be concluded later on in this section.
- We proposed a feedback jamming based MAC layer multicast protocol SEQ-LBP, which enhances LBP with a MAC control frame carrying the Sequence number. Initially, LBP is not reliable for the non-leader receivers and has poor performance at high error rates due to no request frame or sequence check. SEQ-LBP solves the problems of LBP well. All the non-leader receivers can send feedbacks according to the timers set based on the SEQ frame. Both the leader receiver and non-leader receivers reply ACK and NACK respectively based on sequence check, hence it avoids the unnecessary transmissions in LBP. SEQ-LBP needs the minimum number of redundancy transmissions among all pure ARQ based schemes.
- To overcome the scalability limitation of pure ARQ schemes, we combined SEQ-LBP and packet level FEC and proposed HLBP. Using a RS block code, parity packets are generated from a block of original data packets. HLBP transmits a block of original data packets using the raw broadcast and retransmits parity packets if necessary using an improved SEQ-LBP which is based on block feedback. HLBP is much more efficient than both LBP and SEQ-LBP especially for large multicast groups. HLBP needs the near-minimum number of redundancy transmissions among all packet level schemes. LBP, SEQ-LBP and HLBP are all back compatible to legacy 802.11 stations.

- We analyzed the performances of LBP, SEQ-LBP and HLBP over two channel models: the i.i.d channel model and the simplified GE channel model. The used metrics include the final PLR at receivers, the expected number of transmission per data packet, the average channel holding time per data packet, the maximum multicast delay in the MAC layer, etc. We also evaluate their performances on NS-2. The simulation results verify the theoretical analyses and show the advantages of the proposed protocols. Due to the SEQ frame, SEQ-LBP avoids the problems of LBP and is more efficient under various scenarios, especially for large multicast groups. Due to the block coding and block feedback, HLBP is much more efficient than both LBP and SEQ-LBP and has a superior scalability with respect to the number of receivers per multicast group. Moreover, simulation results confirm that SEQ-LBP outperforms the application layer ARQ schemes with both a shorter multicast delay and a higher efficiency. Meanwhile, under the same delay constraints, HLBP is more efficient than the application layer HEC schemes. Furthermore, confirmed by a rough calculation, SEQ-LBP and HLBP outperform the application layer ARQ and HEC respectively further for multi-hop multicast, e.g. in wireless Mesh, Sensor or Ad Hoc networks. In conclusion, SEQ-LBP is a good approach for small multicast group while HLBP is better for large multicast groups.
- We confirmed the feasibility of ACK/NACK jamming through theoretical analyses, NS-2 simulations, as well as measurements on a real test-bed. Using *Atheros* chipset along with the *Madwifi* driver, we designed a flexible software platform running in real-time Linux. Our platform supports microsecond precision and packet transmissions at a configurable time and frame format by not triggering hardware level CSMA contention or backoff schemes. Based on the platform, we implemented a driver level dynamic leader selection algorithm and a multicast management approach. By hundred hours of tests (each one lasts for several hours) under various scenarios, we found that the hardware ACK/NACK jamming probability can be as high as 0.99+ for normal scenarios (about 0.90+ for the worst case with only two receivers which have nearly the same channel condition) when a simple dynamic leader selection

algorithm is used. As a result, we confirmed that, for the first time, the hardware ACK/NACK jamming can be applied as a multicast feedback in the design of MAC layer reliable Multicast.

- We implemented a driver level SEQ-LBP with dynamic leader selection and multicast management on the test-bed and confirmed its performance for recovering multicast packet losses. Based on this we assume that a SEQ-LBP implementation, if incorporated into the wireless modem and thus with more precise timing, is an effective and efficient MAC layer multicast ARQ mechanism. Our driver level SEQ-LBP can replace the normal MAC broadcast in the *Madwifi* driver and provide a MAC layer multicast ARQ for real applications.
- We evaluated a pure NACK jamming based ARQ scheme on the test-bed and confirmed the fake detection problems of pure NACK aggregation which cause high residual error rates or unnecessary retransmissions. Busy tone and physical subcarrier based multicast schemes suffer from the same or similar limitations as well. These limitations should be considered for the design of MAC layer multicast and cross MAC and Physical layer multicast for wireless networks. Moreover, we also discussed about the fine granularity MAC layer coding, data rate adaptation mechanisms and crossing layer issues which could enhance the proposed feedback jamming based protocols.

9.2. Future Work

In this section, we discuss some directions of future work related to the work described in this thesis.

- Due to the limitations of our test-bed, we tested the feedback jamming probabilities using MAC address deceit and the implementation of SEQ-LBP is in the driver level. Further tests for the feedback jamming scheme and related protocols on a specified IEEE 802.11 test-bed are still needed.

- To realize a MAC layer multicast, no matter polling based ones or feedback jamming based ones, the MAC layer multicast management, e.g. member joining, member leaving, etc, is fundamental. This thesis just implements a basic scheme with only one multicast group. Further exploration about MAC layer multicast management is still necessary, in particular multicast group management, cooperation with upper layer multicast managements, admission control, etc.
- Cross layer optimization is a potential direction to further improve the total performance of multicast in wireless LANs. The more information the MAC layer knows about upper layer applications, the better it can support them. For example, the MAC layer can share the multicast management in upper layers. Moreover, as the MAC layer multicast cannot guarantee the final PLR due to long error bursts, application layer multicast correction mechanisms are still needed to control the final PLR. The potential cooperation issues also include: MAC layer retry limit adaptation, channel prediction, data rate adaptation, parameters optimization, admission control, content aware optimization, etc.
- As described in this thesis previously, MAC layer multicast have better performances (shorter delays and higher efficiencies) for multi-hop multicast, e.g. in wireless Mesh, Sensor and Ad Hoc networks. Moreover, the MAC layer multicast, in particular the proposed feedback jamming based approaches, could be a potential candidate for multicast in the Cyber Physical Systems[12] and Machine-to-Machine networks[13] which are emerging and even driving the next industrial revolution. The related topics for multi-hop multicast include: multicast architecture, protocol design, performance optimization, multicast management, routing, reliable relay, cross layer optimization, etc.
- To support high rate real-time multimedia traffic (unicast and multicast) in wireless LANs, a potential architecture is to apply a smart gateway to control and to optimize the multimedia transmissions in the local networks. One related application is the

[12] http://en.wikipedia.org/wiki/Cyber-physical_system and http://www.cps-vo.org/.

[13] http://en.wikipedia.org/wiki/Machine-to-Machine

smart home, where wireless LAN is definitely a candidate to support multimedia traffic delivery. Moreover, a smart gateway is also a potential solution for energy monitoring and management, which is a more and more important issue for wireless networks and devices. The other related functions include admission control, channel status measurement and prediction, local error correction, QoS control and optimization, load balance, etc.

- As described in Chapter 8, the future wireless LANs will have much higher data rates, where the legacy MAC protocols will be less efficient and the optimal packet sizes will be larger (e.g. through data merging). In this situation, the ACK/NACKs feedback jamming will shows a bigger advantage for MAC layer multicast. Moreover, with the development of hardware, coding in the MAC layer with various granularities could become reality because it is very suitable for large data packets. The other related topics include: protocol design, architecture, data merging, performance optimization, cooperation with Network Coding, parameters adaptation, etc.
- Another direction of future work is to apply feedback jamming based multicast error correction to other types of networks, e.g. mobile networks, LET-Advanced. As described in chapter 4, the feedback jamming scheme does not need strict time synchronization, in particular the distance difference among receivers could be as far as 10 kilometers, which is suitable for mobile communications. As the multicast group sizes are always very large, the feedback jamming scheme will show outstanding performance with both a short delay and a high efficiency. Moreover, with the smarter control of the transmitting power and large multicast groups, the feedback jamming probabilities could be very close to 1. Furthermore, the feedback jamming could be cooperated with PHY layer and inspires new multicast mechanisms for broadcast and multicast delivery which is an important service in mobile networks.

9.3. Publications

The publications related to this thesis are as follows:

- Miroll, J.; Li, Zhao; Herfet, Th.: “Wireless Feedback Cancellation for Leader-Based MAC Layer Multicast Protocols”, The 14th IEEE International Symposium on Consumer Electronics (ISCE2010), June 2010.
- Li, Zhao; Miroll, J.; Herfet, Th.: “Video Transmission in IEEE 802.11aa”, NEM Summit 2009 "Towards Future Media Internet", Saint Malo, September 2009.
- Li, Zhao; Herfet, Th.: “MAC Layer Multicast Error Control for IPTV in Wireless LANs”, IEEE Transactions on Broadcasting, June 2009, Volume 55, Number 2/II, p. 353.
- Li, Zhao; Herfet, Th.: “HLBP: A Hybrid Leader Based Protocol for MAC Layer Multicast Error Control in Wireless LANs”, IEEE Global Communication Conference 2008 (GlobeCom2008), New Orleans, LA, USA, Nov. 30th – Dec. 4th, 2008.
- Li, Zhao; Herfet, Th.: “Beacon-driven Leader Based Protocol over a GE Channel for MAC Layer Multicast Error Control”, International Journal of Communications, Network and System Science (IJCNS), 2008.
- Li, Zhao; Herfet, Th.: “BLBP: A Beacon-driven Leader Based Protocol for MAC Layer Multicast Error Control in Wireless LANs”, 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008), Dalian, China, October 12th-14th, 2008.

Bibliography

[11n09] IEEE 802.11n-2009, **“**Amendment 5: Medium Access Control (MAC) and Physical Layer (PHY) specifications, enhancement for higher throughput”, 2009.

[3gp00] 3GPPTR25.835(v1.0.0). Report on Hybrid ARQ Type II/ III (Release 2000). Sep., 2000.

[3gp05a] 3GPP TS 26.346 V6.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; *Protocols and Codecs*, June 2005.

[3gp05b] 3GPP TS 23.246 V6.9.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Architecture and functional description, Dec. 2005.

[3gp06] 3GPP2 Std. C.S0054-A v1.0, CDMA2000 high rate broadcast-multicast packet data air interface specification, Feb. 2006.

[Aci99] O. F. Acikel and W. E. Ryan, "Punctured Turbo Codes for BPSK/QPSK Channels," *IEEE Trans. Communications*, vol. 47, No. 9, pp. 1315-1323, Sept. 1999.

[Ada04] B. Adamson, C. Bormann, M. Handley and J. Macker, “Negative-Acknowledgment (NACK) – Oriented Reliable Multicast (NORM) Protocol,” RFC 3940, Nov. 2004.

[Ahl00] R. Ahlswede, N. Cai, S. R. Li, and R.W. Yeung, “Network information flow”, In Proc. of IEEE Transactions on Information Theory, July 2000.

[Ake04] R. Akester: "A Resilient Multicast Protocol for Digital TV over 802.11 Wireless Networks", In Transactions on Information Science and Applications, Issue 3, Vol 1, 2004.

[Ana07] B. Anas and S. Takuro, "Adaptive FEC Reliable Multicast MAC Protocol for WLAN," *IEEE 66th Vehicular Technology Conference, VTC-2007 Fall*, pp. 244 – 248, Sept. 30 2007-Oct. 3 2007.

[Arm92] S.Armstrong, A.Feirer, and K. Marzullo, "Multicast transport protocol," RFC, 1301, Feb. 1992.

[Bas06] A. Basalamah, H. Sugimoto, and T. Sato, "A Rate Adaptive Multicast Protocol for Providing MAC Layer Reliability in WLANs," *IEICE Transactions on Communications*, no. 10, pp. 2733-2740, 2006.

[Beh07] K.C.Beh, A.Doufexi, S.Armour, "Performance Evaluation of Hybrid ARQ. Schemes of 3GPP LTE OFDMA System", in *Proceedings of Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, 2007.

[Ber03] G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda, "Performance anomaly of 802.11b," In Proc. IEEE INFOCOM 2003, San Francisco, USA, 2003.

[Bia00] Giuseppe Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal on Selected Areas in Communications*, Vol. 18, Issue 3, Mar 2000.

[Cai01] G. Caire and D. Tuninetti, "ARQ protocols for the Gaussian collision channel," *IEEE Trans. Inform. Theory*, Vol. 47, pp. 1971-1988, July 2001.

[Cai02] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan., "Internet Group Management Protocol, Version 3", RFC 3376, Oct. 2002.

[Car97] G. Carle and E.W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," *IEEE Network*, Vol.11, Issue 6, Nov.-Dec. 1997, pp.24 – 36.

[Che06] A. Chen, G. Chandrasekaran, D. Lee, and P. Sinha, "HIMAC: High throughput mac layer multicasting in wireless networks," in *IEEE MASS*, 2006.

[Che08] Q. Chen, F. Schmidt-Eisenlohr, and D. Jiang, “Overhaul of IEEE 802.11 Modeling and Simulation in NS-2 (802.11Ext)”, http://dsn.tm.uni-karlsruhe.de/medien/downloads_old/Documentation-NS-2-80211Ext-2008-02-22.pdf, 2008.

[Cho06] Sunghyun Choi, Youngkyu Choi, and Inkyu Lee, "IEEE 802.11 MAC-Level FEC with Retransmission Combining", *IEEE Trans. on Wireless Communications*, vol. 5, no. 1, pp. 203-211, January 2006.

[Cos98] D. J. Costello, Jr., J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of Error-Control Coding", *IEEE Trans. Inform. Theory*, Vol. 44, pp. 2531-2560, Oct. 1998.

[Dem06] Murat Demirbas, Muzammil Hussain, "A MAC Layer Protocol for Priority-based Reliable Multicast in Wireless Ad Hoc Networks," *International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06)*, pp.447-449, 2006.

[Den95] H. R. Deng and M. L. Lin, ‘‘A Type I Hybrid ARQ system with Adaptive Code Rates’’, *IEEE Transactions on Communications, COM-*43(2/3/4):733-737, February/March/April 1995.

[Dju09] P. Djukic and P. Mohapatra, "Soft-TDMAC: A software TDMA-based MAC over commodity 802.11 hardware", INFOCOM 2009, IEEE, pp. 1836-1844, April 2009.

[Dut09] A. Dutta, D. Saha, D. Grunwald and D. Sicker, "SMACK - a SMart ACKnowledgment Scheme for Broadcast Messages in Wireless Networks", *In Proceedings of SIGCOMM 2009, ACM*, 2009.

[Ell63] E.O. Elliott, “Estimates of error rate for codes on burst-noise channels,” *Bell Syst. Tech. J.*, vol.42, pp.1977-1997, Sept. 1963.

[Ets07] ETSI TS 102 034 V1.3.1, Transport of MPEG 2 Transport Stream (TS) based DVB Services over IP Based Networks, DVB Blue Book A086rev5, Oct. 2007.

[Fai02] G. Fairhurst and L. Wood, "Advice to Link Designers on Link Automatic. Repeat Request (ARQ)," IETF RFC 3366, Aug. 2002.

[Flo97] Floyd, S., Jacobson, V., McCanne, S., Liu, C., and Zhang, L. “A Reliable

Multicast Framework for Light–weight Sessions and Application Level Framing", *In IEEE/ACM Transactions on Networking*, Vol.5, No.6, pages 784–803, December 1997.

[FMI09] Th. Herfet, "Future Media Internet: Video- & Audiotransport – A New Paradigm", *Lecture WS09/10, Telecommunications Lab, Saarland University*, 2009.

[FMI10] White paper on "Future Media Internet", *European Commission, Networked Media Unit, Information Society and Media*, April 2010.

[Fuj04] H. Fujisawa, K. Aoki, M. Yamamoto, and Y. Fujita, "Estimation of multicast packet loss characteristic due to collision and loss recovery using FEC on distributed infrastructure wireless LANs", in *Proc. IEEE WCNC* Vol.1, Page(s): 399 – 404, 21-25 March 2004.

[Gil60] E.N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol.39, pp.1253-1265, Sept. 1960.

[Gor07] Manuel Gorius, "Application Layer Hybrid Error Coding – (AL-HEC) for In-Home Multicast digital TV Distribution," *Master's thesis, Saarland University*, Germany, 2007.

[Gup03] S. K. S. Gupta, V.Shankar, and S. Lalwani, "Reliable Multicast MAC Protocol for Wireless LANs", *IEEE International Conference on Communications*, Vol. 1, pp. 93-97, 2003.

[Had02] Z. Hadzi-Velkov and B. Spasenovski, "Capture Effect in IEEE 802.11 Basic Service Area under Influence of Rayleigh Fading and Near/Far Effect", *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 1 (2002).

[Har08] B. Hart, L. Qian, N. Kakani, A. Ashley, L. Chu, G. Vlantis, H. Liu, I. Mandrekar, M. Wu, R. Perumanam, S. Mathur, "IEEE P802.11 Wireless LANs: Broadcast/Multicast Requirements", doc.: IEEE 802.11-08/1093r0, September 2008.

[Hol01] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multihop wireless networks," in Proceedings of ACM MOBICOM 2001, Rome, Italy, 2001.

[IEEE07] IEEE Std. 802.11™-2007, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, *IEEE Computer Society*, 2007.

[Joe97] Inwhee Joe, “An adaptive hybrid ARQ scheme with concatenated FEC codes for wireless ATM,” in *Proceedings of ACM Mobicom’97*, Budapest, Hungary, Sept. 1997.

[Kam97] A. Kamerman and L. Monteban, “WaveLAN II: A High-Performan Wireless LAN for the Unlisenced Band,” *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[Kar03] S. Karande, Syed A. Khayam and et al., “Analysis and Modeling of Errors at the 802.11b Link Layer,” *IEEE ICME*, July 2003.

[Kha03] S.A. Khayam, S.S. Karande, H. Radha, and D. Loguinov, "Performance Analysis and Modeling of Errors and Losses over 802.11b LANs for High-Bitrate Real-Time Multimedia," Signal Processing: Image Communication, vol. 18, no. 7, August 2003.

[Kim08] B.-S. Kim, S.W. Kim, and R. EH, "OFDMA-Based Reliable. Multicasting MAC Protocol for WLANs," *IEEE Trans. Veh. Technol*, vol. 57, no. 5, Sept. 2008.

[Kim99] J. H. Kim and J. K. Lee, “Capture Effects of Wireless CSMA/CA Protocols in Rayleigh and Shadow Fading Channels”, *IEEE Transactions on Vehicular Technology*, Vol. 48, Issue 4, July 1999.

[Kor07] J. Korhonen, and P. Frossard, “Bit-error Resilient Packetization for Streaming H.264/AVC Video”, Mobile Video Workshop (MV'07), Augsburg, Germany, pp. 25-30, September 2007.

[Kur99] J. Kuri, S. K. Kasera, “Reliable multicast in multi-access wireless LANs”, *in IEEE Infocom 1999*, 1999.

[Lar04] Larzon, L., Degermark, M., Pink, S., Jonsson, L., and Fairhurst, G., "The Leightweight User Datagram Protocol (UDP-Lite)", IETF RFC 3828 (2004).

[Li06] X. Li and Q. Zeng, "Capture Effect in the IEEE 802.11 WLANs with Rayleigh Fading, Shadowing, and Path Loss," *IEEE International Conference on*

Wireless and Mobile Computing, Networking and Communications, 2006. (WiMob'2006), 2006, pp. 110-115.

[Lin93] Lin, S. and D. Costello, "Error Control Coding: Fundamentals and Applications", *Prentice Hall*, 1993.

[Lot07] C. Lott, O. Milenkovic, and E. Soljanin, ``Hybrid ARQ: theory, state of the art and future directions," *in Proc. IEEE Inform. Theory Workshop*, pp. 1--5, July 2007.

[Lu08a] M. Lu, P. Steenkiste, and T. Chen, "Using commodity hardware platform to develop and evaluate CSMA protocols", *Proc. of ACM WiNTECH 2008*, Sept. 2008.

[Lu08b] M. Lu, P. Steenkiste, and T. Chen, "FlexMAC: a wireless protocol development and evaluation platform based on commodity hardware", *Proc. of ACM WiNTECH 2008*, Sept. 2008.

[Lub97] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, V. Stemann, "Practical Loss-Resilient Codes", *29th STOC'97*, 1997.

[Lub02] M. Luby, "LT codes," *In 43rd IEEE Symposium on Foundations in Computer Science*, Nov. 2002.

[Mor02] R. H. Morelos-Zaragoza, "The Art of Error Correcting Coding," *John Wiley*, 2002.

[Mus89] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channels," *IEEE Trans. on Inform. Theory*, vol. 35, pp. 1277-1290, Nov. 1989.

[Nar97] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Communication Letters*, vol. 1, pp. 49-51, Mar. 1997.

[Neu05] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald. "Softmac - flexible wireless research platform", *In HotNets IV*, Nov. 2005.

[Nil02] T.Nilsson, G.Wikstrand, J.Eriksson, "Early Multicast Collision detection in CSMA/CA Networks" *The Fourth IEEE Conference on Mobile and Wireless Communications Networks* (MWCN 2002), 2002.

[Non97] J. Nonnenmacher, E. W. Biersack, and D. Towsley, ‘‘Parity-Based Loss Recovery for Reliable Multicast Transmission’’, *In Proceedings of ACM SIGCOMM’97*, pp. 289-300, Cannes, France, September 1997.

[Non98a] J. Nonnenmacher and E. W. Biersack, "Optimal multicast feedback," *in IEEE Infocom 1998*, p. 964, March/April 1998.

[Non98b] J. Nonnenmacher, E.W. Biersack and et al., “Parity-Based Loss Recovery for Reliable Multicast Transmission,” *IEEE/ACM Trans. Networking*, Vol. 6, Aug. 1998.

[Ott04] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, “Extended RTP profile for RTCP-based feedback,” *IETF RFC*, draft-ietf-avt-rtcp-feedback-11.txt, August 2004.

[Par06] Y. Park, Y. Seok, N. Choi, Y. Choi, and J.-M. Bonnin, “Rate-adaptive multimedia multicasting over ieee 802.11 wireless lans,” in *IEEE CCNC*, 2006.

[Pej96] S. Pejhan, M. Schwartz, and D. Anastassiou, “Error control using retransmission schemes in multicast transport protocols for real-time media**”,** *IEEE/ACM transaction on networking*, Volume 4, Issue 3, Page(s):413 – 427, June 1996.

[Pen02] Peng Ge, P.K. McKinley, "Comparisons of error control techniques for wireless video multicasting", *The 21st IEEE International Conference on Performance, Computing, and Communications*, pp.93-102, 2002.

[Pet03] L. L. Peterson and B. S. Davie, “Computer Networks: A Systems Approach”, Third Edition, 2003.

[Pin94] S. Pingali, D. Towsley, and J. Kurose, “A comparison of sender-initiated and receiver-initiated reliable multicast protocols**”,** in *Proc ACM SIGMETRICS*, Nashville, TN, May 1994.

[Qia00] D. Qiao and K. G. Shin, “A two-step adaptive error recovery scheme for video transmission over wireless networks,” *in IEEE Infocom 2000*, March 2000.

[Rap96] T. S. Rappaport. “Wireless Communications: Principles and Practice”, Prentice Hall, 1996.

[Riz97a] L. Rizzo, “Effective erasure codes for reliable computer communication protocols,” *ACM Computer Communication Review*, April 1997.

[Riz97b] L. Rizzo, "On the Feasibility of Software FEC," DEIT Technical Report LR-970131, Available as http:l/www.iet.unipi.it/ luigi/softfec.ps, Jan. 1997.

[Row00] D. N. Rowitch, L. B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Transactions on Communications*, Vol. 48 Issue: 6, pp. 948 959, June 2000.

[Roy00] Royer, E. M. and Perkins, C. E., "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing", IETF, Intemet Draft: draft- ietf-manet-maodv-00.txt, 2000.

[Sad04] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “OAR: an opportunistic MAC for multirate ad hoc networks,” ACM Journal of Mobile Networks and Applications (MONET), 2004.

[Sch96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP – A Transport Protocol for Real – time Applications,” RFC 1889, January 1996.

[Ses04] S. Sesia, G. Caire, and G. Vivier, “Incremental redundancy hybrid ARQ schemes based on low-density parity check codes,” *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311-1321, 2004.

[Sha06] A. Sharma, M. Tiwari, and H. Zheng. "Madmac: Building a reconfigurable radio testbed using commodity 802.11 hardware", In Proceedings of IEEE SECON-WSDR, Sept. 2006.

[Sho06] A. Shokrollahi, “Raptor Codes,” *IEEE Trans. Information Theory*., vol. 52, no.6, pp.2551-2567, June 2006.

[Sin08] H. Singh, S. Electronics et al., “IPTV over Wireless LAN: Promises and Challenges,” *IEEE Consumer Communications and Networking Conference*, Jan. 2008.

[Sob96] JL Sobrinho, AS Krishnakumar. “Real-time traffic over the IEEE 802.11 medium access control layer”, *Bell Labs Technical Journal* 1996, 172-87.

[Sol03] Soljanin E., "Hybrid ARQ in Wireless Networks", presented at Wireless System

Lab Seminar, Texas A&M University, April 2003, and DIMACS Workshop on Network Information Theory, March 2003.

[Spe01] Speakman, T., Crowcroft, J., Gemmell, J., Farinacci, D. , Lin, S., Leshchiner, D., Luby, M., Montgomery, T. , Rizzo, L., Tweedly, A., Bhaskar, N., Edmonstone, R., Sumanasekera, R., Vicisano, L., PGM Reliable Transport Protocol Specification, RFC 3208, December 2001.

[Stu02] Gordon L. Stuber, "Principles of Mobile Communication", Edition II, 2002.

[Sun02] M. Sun, L. Huang, A. Arora, and T. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," *in Proc. IEEE ICPP '02*, July 2002.

[Tan07a] Guoping Tan and Th. Herfet, "Optimization of an RTP Level Hybrid Error Correction Scheme for DVB Services Over Wireless Home Networks Under Strict Delay Constraints," *IEEE Trans. Broadcasting*, Vol. 53, Issue 1, pp.297 – 307, Mar. 2007.

[Tan07b] G. Tan and Th. Herfet, "Application Layer Hybrid Error Correction with Reed-Solomon Code for DVB Services over wireless LANs," *IEEE WiCOM2007*, Shanghai, China, Sep. 2007.

[Tan09] G. Tan, "Optimum Hybrid Error Correction Scheme under Strict Delay Constraints", *PhD. Thesis, Saarland University*, 2009.

[TGaa10] IEEE TGaa, Draft P802.11aa D1.0, "Amendment 4: MAC enhancements for Robust Audio Video Streaming", July 2010.

[Tou98] J. Tourrilhes, "Robust broadcast: improving the reliability of broadcast transmission on CSMA/CA", in *Proceedings of the Ninth IEEE International Symposium on. Personal, Indoor and Mobile Radio Communications*, 1998.

[Tow97] D. F. Towsley, J. Kurose and S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No.3, April 1997.

[Tri82] S. K. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Applications", *Prentice-Hall*, Englewood Cliffs, NJ, 1982.

[Tun02] D. Tuninetti and G. Caire, "The optimal throughput of some wireless multiaccess systems," IEEE Trans. Inform. Theory, Oct. 2002.

[Vid04] R. Vida, L. Costa, Eds., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[Wai88] D. Waitzman, C. Partridge, and S. Deering, “Distance Vector Multicast Routing Protocol”, http://www.ietf.org/rfc/rfc1075.txt, RFC 1075, 1988.

[Wie06a] Sven Wiethölter, Christian Hoene, “An IEEE 802.11e EDCA and CFB Simulation Model for ns-2”, http://www.tkn.tu-berlin.de/research/802.11e_ns2/.

[Wie06b] Sven Wiethölter, M. Emmelmann, Christian Hoene, Adam Wolisz "TKN EDCA Model for ns-2", *Technical Report TKN-06-003, Telecommunication Networks Group, Technische Universität Berlin*, June 2006.

[Wei99] L. Wei, et. al, “Protocol Independent Multicast Version 2 Dense Mode Specification”, http://www.ietf.org, draft-ietf-pim-v2-dm-03.txt, 1999.

[Wel05] Michael Welzl, “Passing corrupt data across network layers: an overview of recent developments and issues”, EURASIP Journal on Applied Signal Processing, v.2005 n.1, p.242-247, 1 January 2005.

[Woz60] J. M. Wozencraft and M. Horstein, "Digitalized communication over two-way channels," *the Fourth London Symp. Inform. Theory*, London, UK, Sept. 1960.

[Woz61] J. M. Wozencraft and M. Horstein, “Coding for two-way channels,” *Tech. Rep. 383, Res. Lab. Electron*., MIT, Cambridge, MA, Jan. 1961.

[Xia07] Y. Xiao, X. Du et al., “Internet Protocol Television (IPTV): The Killer Application for the Next-Generation Internet,” *IEEE Commun. Mag.*, pp.126-134, Nov. 2007.

[Yee95] J.R. Yee and E.J. Weldon, Jr., “Evaluation of the performance of error-correcting codes on a Gilbert channel”**,** *IEEE Tran. On Communications*, Volume 43, Issue 8, Aug. 1995.

[Zhu04] Y. Zhu and Thomas Kunz, “MAODV implementation for NS-2.26”, *Carleton university, technical report SCE-04-01*, Jan 2004.