Reliable Multicast Mechanism in WLAN with Extended Implicit MAC Acknowledgment

Xiaoli Wang, Lan Wang, Yingjie Wang, Daqing Gu
Docomo Beijing Communication Labs Co., Ltd, China
E-mail: {wangxl, wangl, wangyj}@docomolabs-beijing.com.cn; daqing.gu@orange-ftgroup.com

Abstract—**In this paper, we study reliable multicast at the MAC layer for IEEE 802.11 Wireless LANs. In IEEE 802.11, multicast protocol is based on the basic access procedure of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This protocol does not provide any recovery mechanism for multicast frames. As a result, transmitted multicast frames may be lost due to collisions or errors. Recently, several reliable multicast protocols at MAC layer have been proposed for 802.11, and they can be classified into two categories: one is based on negative feedback (NFB-based) and the other is based on positive feedback (PFB-based). We first analyze the problems with existing reliable multicast MAC protocols and then propose a novel PFB-based scheme: "Extended Implicit MAC Acknowledgment (EIA)". With EIA, ACK packets are eliminated and collisions of CTS frames are avoided. Since ACK packets account for nearly one-third of the total control overhead, we argue that such a scheme can be beneficial. Simulation results, using the OPNET, confirm that the improvements are encouraging.**

I. INTRODUCTION

Multicast is an efficient way to transmit data to a group of receivers, which brings lower network costs and bandwidth consumption than unicast to individual group members, especially in wireless environment. In WLAN, multicast can efficiently support a variety of applications, such as multimedia conferencing, shared whiteboards, distance learning, multi-party games and distributed computing. However, in IEEE 802.11 specification, the multicast sender simply listens to the channel and then transmits its data frame when the channel becomes free for a period of time. There is no MAC-level recovery on multicast frame as in unicast. As a result, the reliability of multicast is reduced due to the increased probability of lost frames resulting from collisions or errors.

Various MAC multicast protocols [1-9] have been proposed recently to enhance the reliability and the efficiency of the 802.11 multicasting. They can be classified into two categories: one is based on negative feed back [1-2] [9] and the other is based on positive feed back [3-8]. However, these protocols have serious problems about reliability and/or efficiency. In this paper, we show the reliable and efficient problems in the NFB-based protocols (LBP, PBP and DBP [1-2]) and demonstrate that while the PFB-based protocols (BMW [3] and BMMM [4]) are logically reliable, they can not be very efficient. Further, towards redressing these reliability and efficiency issues, we design a novel PFB-based multicast MAC protocol, extended implicit MAC acknowledgment (EIA).

The proposed reliable multicast mechanism EIA has two advantages: (1) the control overhead is reduced by using an implicit acknowledgment scheme. The main idea is as follows. If there are at least two multicast packets to be transmitted to a group, then the sender requests the receivers not to transmit an explicit ACK packet. When the RTS/CTS handshake is initiated for the second packet, the receivers can acknowledge the receipt of the first packet by piggybacking the CTS with a SEQ field. Similarly, the CTS frame for the third packet carries the acknowledgment information of the second packet, and so on. When it is time for the MAC layer to multicast the last packet, RTS is used to explicitly notify the group members that the receivers must send an explicit ACK packet now. It is obviously that as long as there are packets in the queue, explicit acknowledgments can be eliminated, resulting in a potential benefits. (2) The collisions among control frame transmissions can be avoided. For a multicast RTS, if more than one intended receiver replies with a CTS frame, these CTS frames may collide with each other at the sender. In order to avoid this problem, EIA provides a simple coordination among the intended receivers. In EIA, each group member is assigned with priority. The CTS frame is sent one after another based on the priority so that the collision of CTS frames can be avoided.

The rest of the paper is organized as follows. In section Ⅱ the current multicast MAC protocols and their problems are described. Section Ⅲ introduces the proposed EIA protocol, followed by a priority setting method in section Ⅳ. Simulation analysis is showed in Section Ⅴ. Finally, Section Ⅵ draws the conclusion and discusses the future research directions.

II. PROBLEMS WITH EXISTING MULTICAST MAC PROTOCOLS

In IEEE 802.11, the RTS/CTS extension is not used for multicast; and the receivers are not required to return an ACK. As a result, the quality of multicast service is not as good as that of unicast.

The LBP (leader based protocol) [1] attempts to extend the IEEE 802.11 multicast protocol with handshaking mechanism and recovery mechanism. The protocol assumes that a receiver is selected as the leader for the multicast group. According to the protocol, when there is a multicast data packet to send, the access point (AP) first sends a multicast RTS (MRTS) frame to all receivers, and only the leader transmits a multicast CTS (MCTS) frame in reply to the AP. The AP is then assured that the channel is granted and starts the transmission of a multicast data frame. The leader sends an ACK in reply if the data is received correctly, or sends a NAK otherwise. If any other receiver detects a transmission error, a NAK is also sent. This NAK frame will collide with the ACK, if any, sent by the leader. This leads to the AP not hearing any ACK, and thus retransmitting the lost frame.

Applying LBP to IEEE 802.11 suffers from some problems: (1) Type-unknown for lost packet. This problem exists in all NFB-based protocols. When collisions or link error occurs, the group member can not receive the frame correctly and then it can not acquire the information contained in the MAC header, such as frame type, source address and destination address. So it is difficult for the receiver to decide what type (NCTS or NAK) the feedback should be and which node the feedback should be sent to. (2) Unnecessary retransmission. After the data multicast, if error occurs at the non-leader hosts, they will send NAK to collide with the leader's ACK. Such collision will lead to no ACK being heard at AP, and then the data is retransmitted. However, the non-leader hosts send a NAK, regardless of whether this erroneous frame has been received successfully before or not, which result in redundant retransmission. (3) Capture effect. When a node receives multiple frames at the same time, the frame with the strongest power can be captured as long as its Signal-to-Interference Ratio (SIR) is larger than 10dB [10]. For a MRTS frame, if the leader and a non-leader host reply with a MCTS and NCTS respectively at the same time, then it is possible for the MCTS with the strongest power to be captured by the AP. Thus, the AP will start to transmit data packet rather than back off.

The delayed feedback-based protocol, termed DBP [1], is different from LBP in two ways: (1) MCTS frame is sent by each receiver instead of only the leader. In order to avoid the collisions of MCTS, each receiver will not send MCTS frame until a random timer expires. (2) Each receiver sends NAK if a transmission error is detected. PBP (Probabilistic feedback-Based Protocol) [1] is similar to DBP with one important difference. In PBP, instead of waiting for a random number of time slots to send a MCTS, the group members send out a MCTS in the slot with a certain probability.

In comparison to LBP, it would take longer time in both DBP and PBP to complete a successful MRTS/MCTS exchange. This is because DBP and PBP have to deal with the possibility of MCTS collisions. They delay feedback or decrease the feedback probability to reduce the possibility of collision. However, they still have to go through several rounds of MRTS-MCTS exchange due to the MCTS collision. This failed exchanges reduce the channel utilization efficiency. Another problem with DBP and PBP is the choice of right parameters for waiting times and probability of sending feedback. This choice is based on the number of the group members.

In [3], the Broadcast Medium Window (BMW) is introduced to provide a reliable broadcast MAC. The basic idea of BMW is to treat each broadcast request as multiple unicast requests. Each unicast is processed using the reliable IEEE 802.11 DCF MAC protocol (i.e., RTS/CTS/DATA/ACK) with some minor modifications. BMW protocol is reliable because the AP will retransmit the data frame until it has received an ACK from every intended receiver. Unfortunately, it is not very efficient. In order to improve the efficiency, Batch Mode Multicast MAC Protocol (BMMM) is proposed in [4]. Fig. 1 illustrates the communication process of BMMM. Although BMMM is more efficient than BMW, the control traffic overhead is still very high. Table Ⅰ shows the number of time slots in physical layer that is required to multicast 512 byte data packet in comparison to the RTS, CTS, RAK and ACK control packets. 10 members in the multicast group are assumed. The table corresponds to a channel data rate of 54 Mbps. Observe from the table, we can see that the total percentage of bandwidth invested on control packets in BMMM accounts for 92%. Detailed explanations can be found in [11]

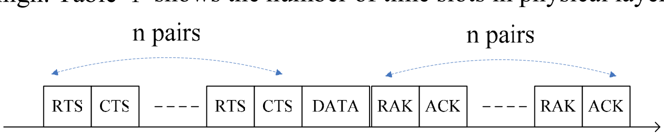

Fig. 1 Primary idea of BMMM

Table Ⅰ
TRANSMISSION TIMES FOR BMMM

RTS	CTS	DATA	RAK	ACK
50 slots	40 slots	15 slots	40 slots	40 slots

III. Extended Implicit ACK Multicast MAC Protocol

In this section, we discuss the main ideas of our proposal in detail. For reliable MAC-layer multicast, two important problems should be resolved: (1) How to decrease the control packet overhead to improve the efficiency. (2) How to avoid the collisions among control frame transmissions. In our proposed EIA protocol, these two problems are resolved by two ways: (1) the acknowledgement information is piggybacked in MCTS frame. In those multicast protocols we mentioned above, reliable delivery is supported by the MRTS/MCTS/DATA/ACK scheme. However, in EIA, the acknowledgement information can be piggybacked in MCTS, so the control traffic can be decreased highly and collision of multiple ACK frames can be eliminated. (2) The MCTS is sent in order. In EIA we allow the MCTS to be sent one after another by deliberately introducing a fixed amount of delay between successive transmissions. Thus, each receiver calculates the time it must wait before sending its MCTS frame. This time is based on the priority. We will discuss the priority setting separately in the last section.

A. *Frame structure*

In EIA, reliable delivery is supported by MRTS/MCTS/DATA scheme. The structure of MRTS frame is the same as RTS defined in IEEE 802.11, which is shown in Fig. 2. It is worth noting that the RA field in the MRTS is the group address instead of the broad address (-1) defined in 802.11. The group address can be derived from the IP-layer multicast address, which is a class D address. The mapping method ought to be the same as what is adopted in Ethernet.

The structure of MCTS frame is defined in Fig. 3, which is modified from the CTS frame by inserting a field SEQ. The SEQ is the sequence number of the latest successfully received DATA frame at a group member. Since the announcement from sequence number, the sender can judge clearly whether the latest transmitted packet have been received correctly or not. As a result, EIA can effectively eliminate the redundant retransmissions caused by LBP.

Frame Control	Duration	RA	TA	CRC

Fig. 2 MRTS Frame Format

Frame Control	Duration	RA	TA	SEQ	CRC

Fig. 3 MCTS Frame Format

B. Protocol description

Consider a simple scenario showed in Fig. 4, in which source node in cell 1 transmits multiple DATA packets to a multicast group, G, in cell 2. Packets are generated at some rate by source node, and then passed to AP1, which in turn passes them to the AP2. The AP2 transmits these multicast packets using the proposed reliable multicast mechanism, EIA. The notations used in this paper are summarized in Table Ⅱ.

When the MAC layer at AP2 receives a multicast DATA packet from the upper layer, it first translates the IP-layer multicast address into MAC-layer group address. Before initiating an MRTS transmission to group G, the MAC layer at node AP2 determines if there are other packets for group G in the outgoing queue. If at least one other packet for G is in the queue, then AP2 set the *subtype* value of frame control field in MRTS to 0000 (this value is reserved in IEEE 802.11). If there are no multicast packets for group G in the queue, then the *subtype* is set to 0001 (this value is reserved in IEEE 802.11). After performing the collision avoidance procedure and when the medium is idle, AP multicasts an MRTS frame to request access to the medium from all members.

On receiving the MRTS from AP2, each group member records the subtype value, and replies with the MCTS. Since multiple nodes exist in the group, the sender may expect multiple MCTS frames. If all the MCTS frames are sent simultaneously, they may not be correctly received. In order to avoid the collision, the MCTS frame is sent one after another by deliberately introducing a fixed amount of delay. Thus, each receiver calculates the time it must wait before sending its MCTS frame. The wait times are calculated as follows. The Mth receiver waits for a time equal to $M \times SIFS + (M-1) \times T_{MCTS}$, where M is the priority assigned to the receiver when it joined to the group. After successful reception of MCTS from all members, AP2 multicasts the DATA frame, and record the sequence number of this DATA to *Frametx_seq*. Later, on receiving the multicast DATA packet from AP2, group members transmit an ACK packet only if the *subtype* in the MRTS was set to 0001; otherwise they omit sending the ACK packet and only record the sequence number of the successfully received DATA to *Framerv_seq*.

Let us assume that AP2 had multiple multicast packets for group G in its queue, and therefore had set the *subtype* to 0000 within the MRTS. As the consequence, group members had not replied with an ACK packet at the end of the dialog. Observe that at the end of the first dialog (i.e., MRTS/MCTS/DATA), AP2 is unaware whether the DATA packet was successfully received by all nodes. Now, AP2 must initiate MRTS transmission for the next packet in the queue. On receiving the MRTS packet form AP2, each group member piggybacks the *Framerv_seq* recorded previously, on the MCTS. If the *Framerv_seq* feeding back from Mth node is less than *Frametx_seq*, it means Mth node received an erroneous DATA packet. After received MCTS frames from all group members, AP2 judges whether need to retransmit the DATA packet. If all *Framerv_seq* are equal to the *Frametx_seq*, then AP2 transmits the new DATA packet. Otherwise, AP2 retransmits the previous packet and temporarily stores the new DATA packet. The key ideas are illustrated in Fig. 5.

Table Ⅱ
Definitions of terms used in the paper

R	The number of multicast receivers in the group
Rcvd_MCTS_number	The number of MCTS frames received at AP when it is waiting for feedback
Frametx_seq	The sequence number of the latest transmitted DATA frame from AP
Framerv_seq[i]	The sequence number of the latest successfully received frame at group member with priority i
SIFS	Short Inter-Frame Spacing
T_{MCTS}	The time to transmit a MCTS frame

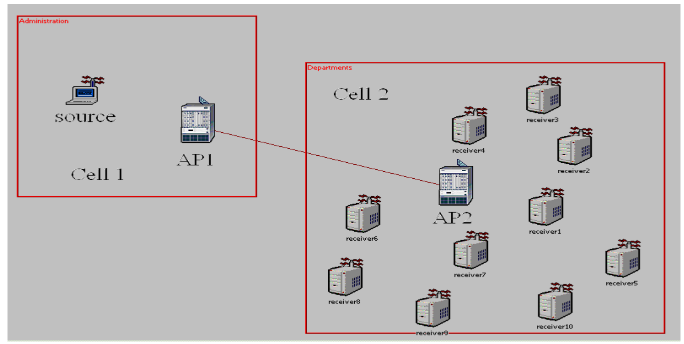

Fig. 4 Multicast Scenario

Next we are going to describe the clearer definition of EIA protocol.

[A] AP->Receivers

1. Send MRTS.
2. Start a timer (timeout period $T_1 = T_{MCTS} + SIFS$) expecting to hear the first MCTS frame before the timer expires.

[B] Receivers->AP

1. On hearing MRTS, each receiver calculates the time it must wait before sending MCTS frame, which is based upon its priority. The receiver with priority m should wait for a time equal to $T_2 = m \times SIFS + (m-1) \times T_{MCTS}$.
2. Start a timer (timeout period is T_2).
3. Piggyback the *Framerv_seq* on the MCTS (If no DATA packets have been received, the *Framerv_seq* is 0)
4. Send MCTS until timer expires.

[C] AP->Receivers

1.1. If a MCTS frame was heard within T_1, AP acquires the *Framerv_seq* contained in the MCTS frame, and then adds it into the array, *Framerv_seq[]*.

1.2. If no MCTS was heard before timer expires, back off and go to step A.

2. Start the timer (timeout period is T_1) expecting to hear the next MCTS before the timer expires.

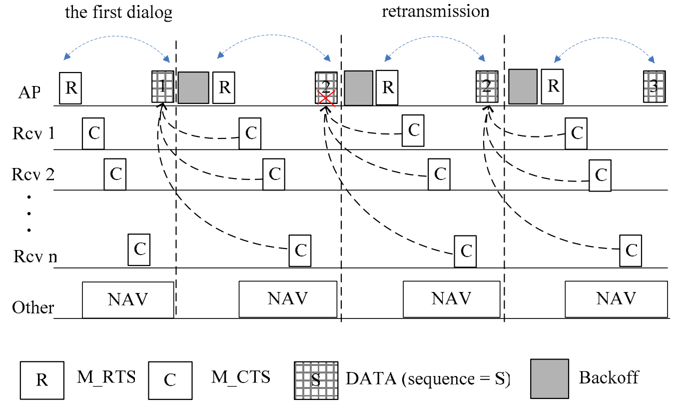

Fig. 5 Transmission between AP and group members

3. After received all MCTS frame (i.e., *Rcvd_MCTS_number* =*R*), AP get an *R*-sized array *Framerv_seq []*.

4.1. If min{*Framerv_seq[]*} = *Frametx_seq*, AP transmit a new DATA packet and then set *Frametx_seq* = *Frametx_seq*+1;

4.2. Otherwise, AP retransmits the previous packets.

[D] Receivers->AP

1.1. If *subtype* in MRTS was set to0000, omit sending the ACK packet and only record *Framerv_seq*;

1.2. Otherwise, send ACK packet according to its priority.

In summary, when the sending rate at the source node is high or bursty, the outgoing queue at the AP2 has always multiple packets to group G. This is an opportunity to eliminate the ACK packet for every DATA packet, because the MCTS of the next packet can be piggybacked with an acknowledgment for the previous packet. Clearly, EIA will decrease the overhead of conventional PFB-based multicast protocols greatly, and then lead to improvements in throughput and delay. In addition, there is always a non-zero probability that multiple ACK packets are lost or collided when using explicit Acknowledgement and then the MRTS/MCTS/DATA/ACK dialog is unnecessarily initiated once again. With our scheme such possibilities are eliminated, resulting in encouraging performance improvement.

IV. PRIORITY SETTING

In this section, we discuss the priority setting process. We assume that upon joining or leaving a group, a terminal sends explicit *join-group* or *leave-group* messages to its AP [1]. The AP maintains a table containing each group and the corresponding member with priority as in Table III.

When a terminal *T* sends a *join-group* message to join group G_i, the AP check the table to find out the existed maximum value of priority, P_{max}, in G_i. Then the AP replied with the message that the priority for T will be P_{max} +1.

When a terminal *T* sends a *leave-group* message to leave group G_i, the AP checks the table to see whether the priority of *T* is maximum or not. If it is, the AP does nothing. If it is not, the AP ought to send a change-priority message to notify other members to change their priority.

Table III

Group-priority table maintained at AP

Group Address	MAC address	Priority
G_1	A_1	1
G_1	A_2	2
G_1	A_3	3
……	…….	……
G_2	A_i	1
G_2	A_{i+1}	2

V. SIMULATION RESULT

We have used network simulator OPNET 11.5 to implement the multicast MAC protocols (EIA, LBP, DBP and IEEE 802.11). In this section we will describe the simulation results. The following metrics are used to compare the performance of different schemes: (1) Packet Delivery Ratio (PDR): Ratio of the number of data packets actually received by group members to the number of data packets which should have been received. (2) Average Packet Delay. (3) Delay jitter.

A. Simulation Setting

We have set up the simulations using a grid of size 300 × 300 with 30 nodes. The simulation scenario is shown in Fig. 4. We use the two ray ground propagation model in the physical layer in one set of experiments and fixed BER for another set of experiments. In the 2nd simulation, we simulate 7 cases. In each case, we set fixed BER, which is selected from $\{10^{-7} — 10^{-4}\}$. Some of the simulation parameters are shown in Table IV

Table IV

Simulation Parameters

Parameters	Value
Application Description	FTP Download
File size	1000byte
Inter-request time	Constant (1 second)
Simulation Time	1 hour
Bandwidth	11M
Node Placement	Random
MAC Protocol	IEEE 802.11, LBP, EIA, DBP
Transport Protocol	UDP

B. Simulation Result

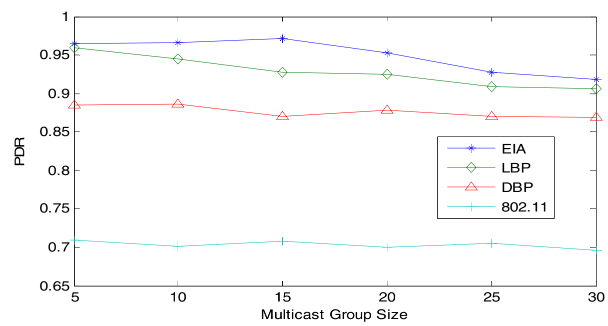

Fig. 6. Packet delivery ratio versus multicast group size with a two ray ground propagation model

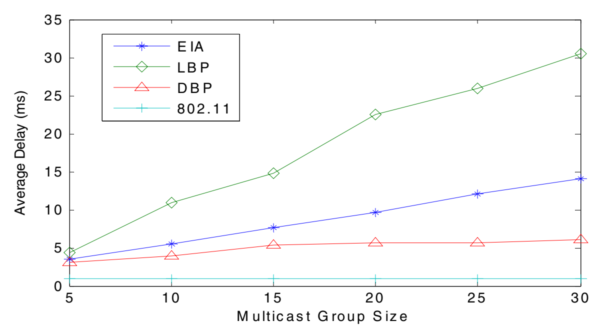

Fig. 7. Average delay versus multicast group size with a two ray ground propagation model

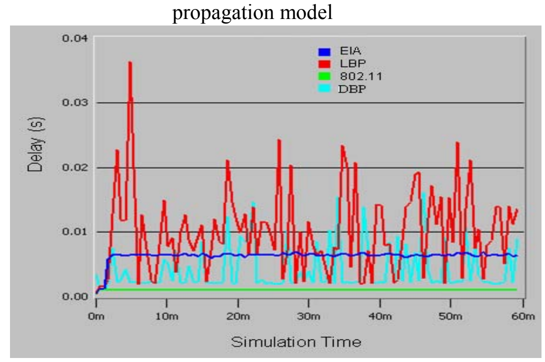

Fig. 8. Instantaneous delay with simulation time with a two ray ground propagation model (group size = 10)

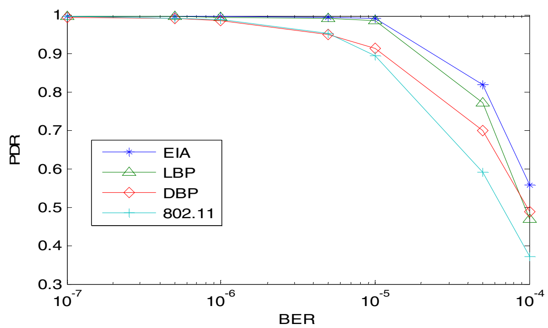

Fig. 9. Packet Delivery Ratio versus BER with 10 nodes

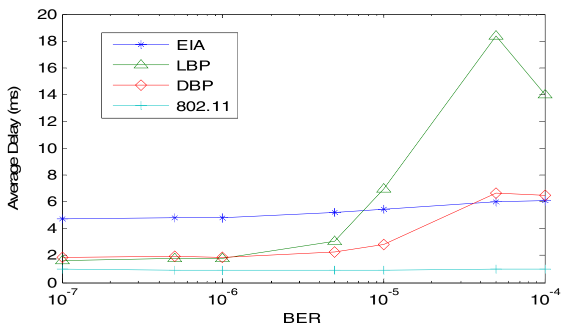

Fig. 10. Average delay versus BER with 10 nodes

Fig.6 compares the packet delivery ratio for EIA, LBP, DBP and 802.11. As evident from the figure, EIA achieves a higher packet delivery ratio than other protocols. The high PDR in EIA is mainly contributed by two aspects. （1） The AP can acquire feedback from all group members, and it will not transmit the new DATA frame until every intended node receives the packet successfully. However, in other NFB-based algorithms, AP may not get the feedback from all nodes because of the capture effect, time out and so on. （2） Unnecessary retransmission is avoided by piggybacking *Framerv_seq* within MCTS. In NFB-based protocols, when the received packet is in error, the group member sends a NAK frame to request retransmission, regardless of whether this erroneous frame has been received successfully before or not, due to the unknown sequence number. This will lead to unnecessary retransmission. In EIA, because of the announcement from SEQ in MCTS, AP can judge correctly whether to retransmit the previous packet or not. As a result, EIA can effectively eliminate the redundant retransmission.

Fig. 7 compares the delay for EIA, LBP, DBP and 802.11. To calculate average packet delay we have considered only the packets which have been successfully received. Since no recovery mechanism or channel acquisition mechanism is carried out in 802.11, the delay in it is the lowest. For EIA, LBP, DBP the average packet delay lies below 35ms, where the delay associated with EIA is higher than DBP, mainly because AP has to wait the MCTS frames from all the group members.

Next we are going to compare the packet delay jitter for EIA, LBP, DBP and 802.11. From Fig. 8 we can see that although the packet delay for EIA is higher than DBP, the delay jitter is lowest. In LBP and DBP, the receivers need to response two control frame (CTS/ NCTS and ACK/NAK) in a round of transmission. There is always a non-zero probability that ACK/NAK packets are lost or collided, and the RTS/CTS/DATA/ACK dialog may be unnecessarily initiated once again. With our scheme such possibilities are eliminated. At the same time, MCTS is responded in sequence, so that the uncertainty of the arriving time for control frame is decreased.

Fig. 9 sketches the system performance as PDR versus BER. From this figure, we can see that EIA outperform other algorithms in varying channel condition. Especially when the BER is high, the advantage of EIA is more obvious.

Fig. 10 sketches the system performance as average packet delay versus BER. When the BER is low, the delay for EIA is higher than other protocols. That is because AP will not send DATA packet until it receives the MCTS from all group members. In contrast, the AP will transmit data as soon as it receives one CTS frame in LBP and DBP. However, when BER is high, the delay for EIA is lower than LBP and DBP. For one thing, the probability that ACK/NAK packets are lost is eliminated in our scheme. For another thing, unnecessary retransmission is avoided because the *Framerv_seq* is piggybacked in MCTS.

VI. Conclusion

In this paper we present a novel PFB-based multicast protocol, EIA, to support multicasting in IEEE 802.11 networks. The proposed EIA address two important problems in reliable multicast: (1) How to alleviate the collision of multiple MCTS frames and multiple ACK frames; (2) How to improve the efficiency. The main idea of EIA is: (1) Piggyback the acknowledgement information in MCTS; (2) Response MCTS in order. Through simulation work we have shown that EIA improves the performance of multicast packet delivery in WLAN. As future work, we plan to investigate the issue of multiple sources for the same multicast session.

References

[1] J. Kuri and S. K. Kasera, "Reliable multicast in Multi-Access Wireless LANs," ACM/Kluwer Wireless Networks Journal, vol. 7, no.4, pp. 359-369, August 2001.

[2] S.K.S. Gupta, V. Shankar, " Reliable multicast MAC Protocol for Wireless LANs", ICC 2003, pp93-pp97, 2003.

[3] K. Tang and M. Gerla, "MAC Reliable Broadcast in AdHoc Networks, " Proc. IEEE MILCOM 2001, pp. 1008-1013, Oct. 2001.

[4] M.T. Sum, L.Huang, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," Proc. of ICPP, pp. 527-536, Aug. 2002.

[5] Hrishikesh Gossain, Nagesh Nandiraju, "Supporting MAC layer Multicast in IEEE 802.11 Based MANETs: Issues and Solutions", LCN'04, 2004.

[6] Jiawei Xie and Anil K, "Improving the reliability of IEEE 802.11 broadcast scheme for Multicasting in Mobile Ad hoc networks", WCNC, pp 126-130, 2005.

[7] Shweta Jain and Samir R. Das, " MAC Layer Multicast in Wireless Multihop Networks", IEEE Communication System Software and Middleware, 2006.

[8] Chong-Wei Bao and Wanjiun Liao, "Performance Analysis of Reliable MAC-Layer Multicast for IEEE 802.11 Wireless LANs," IEEE ICC, pp 1378-1382, 2005.

[9] S.K.S Gupta et al, "Reliable Multicast MAC Protocol for Wireless LANs,", IEEE ICC, pp 93-97, 2003.

[10] V. Bharghavan et al, "MACAW: A Media Access Protocol for Wireless LANs," Proc. ACMSIGCOMM1994, pp. 212-225, Aug. 1994.

[11] Romit Roy Choudhury et al, "Implicit MAC Acknowledgment: An Optimization to 802.11", IEEE WTS , 2005.