

Efficient Multicast Scheme based on Hybrid ARQ and Busy Tone for Multimedia Traffic in Wireless LANs

Sunmyeng Kim^{1,*} and Young-Jong Cho²

¹ Department of Computer Software Engineering, Kumoh National Institute of Technology, Gumi, Korea

² Department of Information and Computer Engineering, Ajou University, Suwon, Korea

Received: 4 Jun. 2013, Revised: 29 Sep. 2013, Accepted: 30 Sep. 2013

Published online: 1 Jan. 2014

Abstract: Multicast in wireless LANs (Local Area Networks) is very useful for transmitting multimedia traffic from a sender to a group of receivers. The IEEE 802.11 WLAN MAC (Medium Access Control) layer does not support reliable multicast since, for multicast transmissions, it has no control packets such as the RTS, CTS, and ACK, which are used for unicast transmissions. Several protocols have been proposed to provide reliable multicast transmissions. However, they are not efficient for IEEE 802.11 WLANs due to the large number of packet transmissions and the excessive control overhead caused by the use of a large number of control packets in the error recovery process. In this paper, we propose a simple and effective scheme. The proposed scheme combines FEC (Forward Error Correction) and ARQ (Automatic Repeat reQuest) to reduce a large number of packet transmissions and to provide data reliability in the IEEE 802.11 WLANs multicast environment. It then uses busy tones to improve efficiency by reducing the excessive control overhead. Performance evaluation is conducted using both numerical analysis and simulation. The results show that the proposed scheme has good performance in terms of normalized throughput and average delay.

Keywords: Busy tone, hybrid ARQ, multicast, multimedia traffic

1 Introduction

When a sender transmits data to multiple receivers, multicasting is very useful compared to unicasting to each receiver. It can save the network bandwidth and reduce the data distribution time by transmitting data to all receivers simultaneously. Several applications such as IPTV, multi-party games, video conferencing, shared whiteboards, distance learning, military communications, and information dissemination need multicast transmissions to provide better quality of service (QoS).

The IEEE 802.11 wireless LAN (Local Area Network) is widely used for wireless access due to its easy deployment and low cost. The IEEE 802.11 standard defines a medium access control (MAC) protocol for sharing the channel among stations [1]. The distributed coordination function (DCF) was designed for a contention-based channel access. The DCF has two data transmission methods: the default basic access and optional RTS/CTS (request-to-send/clear-to-send) access. The basic access method uses a two-way handshaking (DATA-ACK) mechanism. The RTS/CTS access method

uses a four-way handshaking (RTS-CTS-DATA-ACK) mechanism to reserve the channel before transmitting long data packets. In the DCF, access to the wireless medium is controlled by the use of inter-frame space (IFS) time between the transmission of packets. Three IFS intervals have been specified by the IEEE 802.11 standard: short IFS (SIFS), point coordination function IFS (PIFS), and DCF-IFS (DIFS). Packet collisions on the medium are resolved using a binary exponential backoff algorithm. While the RTS-CTS-ACK exchange and the binary exponential backoff algorithm are only used for unicast transmissions, multicast packets are transmitted without these mechanisms. In other words, a multicast sender listens to the channel and then transmits a data packet when the channel is sensed idle for a defined period of time. After receiving the data packet, receivers do not send any feedback such as ACK. Therefore, the sender does not know whether or not the receivers receive the data packet successfully. Because of the lack of feedback, for multicast transmissions, the binary exponential backoff algorithm is useless and the contention window size is fixed. Because of this,

* Corresponding author e-mail: sunmyeng@kumoh.ac.kr

multicast transmissions in the IEEE 802.11 DCF do not provide error recovery for data packets, and thus do not guarantee reliability [2,3].

Various protocols have been proposed to enhance the reliability of multicast transmissions for the IEEE 802.11-based wireless networks [4,5,6,7,8,9]. They can be classified into two categories: one is based on ACK [4,5,6,7] and the other is based on leader [8,9]. In the ACK-based protocols, a sender retransmits data packets until it receives all the ACK packets from all receivers. In the leader-based protocols, a sender retransmits data packets when there is no ACK from a leader receiver. Previous protocols still have serious problems in reliability and efficiency due to the following three reasons [7]. First, they cause the excessive control overhead by the use of a large number of control packets in the error recovery process. Second, a sender has to retransmit the same data packets several times to satisfy all receivers. Third, in the leader-based protocols, receivers do not acquire the information such as frame type, source address, and destination address in the MAC header when receiving an erroneous packet. Therefore, it is difficult for the receivers to decide whether or not to send feedback, and what type of feedback they need to send. This may result in the malfunctioning of the receivers.

Most of previous protocols are based on the ARQ (Automatic Repeat reQuest) scheme used for unicast communications over IEEE 802.11 DCF. Receivers request the retransmission of missing data packets by sending positive acknowledgements (ACKs) and/or negative acknowledgements (NACKs). ARQ-based protocols cannot reduce the impact of independent losses from different receivers. However, when an FEC (Forward Error Correction) is used, only one retransmitted parity packet can satisfy all receivers which lose different packets [10,11,12,13]. FEC is especially effective for multimedia multicast transmissions [14,15].

In order to improve multicast reliability while minimizing feedback overhead, we propose a simple and effective scheme, called reliable multicast based on busy tone (RMBT). The proposed scheme combines FEC and ARQ mechanisms to reduce a large number of packet transmissions and to provide data reliability in the IEEE 802.11 WLAN multicast environment. FEC is the best mechanism to use for real-time multimedia applications since it does not need time to detect errors and request retransmission. The RMBT uses block erasure codes denoted as $C(n, k)$. In these codes, the original data stream is divided into blocks of k packets and then $n (> k)$ encoded packets are generated by using k original packets [10,11,12,13]. A receiver can recover all the original k packets as long as it successfully receives k distinct packets. The proposed scheme also uses short duration busy tone signals (i.e., pulses of energy) [16,17] to improve efficiency by reducing the excessive control overhead.

The paper is organized as follows. The related work is presented in Section 2. In Section 3, the proposed RMBT scheme is explained in detail. Normalized throughput is analyzed in Section 4. In Section 5, we discuss numerical and simulation results. Finally, we conclude in Section 6.

2 Related Work

Various protocols have been proposed in the literature over the IEEE 802.11 DCF to enhance the reliability and efficiency of multicast transmissions. In the most of them, the RTS/CTS extension is used and receivers are required to send feedback signals such as ACK/NACK.

Srinivas proposed a slot reservation based reliable multicast (SRB) protocol [6]. The SRB uses RTS-CTS-DATA-ACK exchange with a slot reservation based scheduling mechanism. To identify each receiver, association ID (AID) is introduced and is assigned to each receiver in the receiver's arrival order starting from 1. A sender transmits an RTS to a multicasting group's address. Receiver i transmits a CTS in the i th time slot. After receiving CTSs from receivers, the sender broadcasts a data packet. Receiver i transmits an ACK in the i th time slot. The SRB protocol schedules the transmission time of CTSs and ACKs to avoid collisions. For the receivers that do not respond with CTSs and/or ACKs, the above procedure is repeated. However, the sender expects CTSs and ACKs only from receivers with transmissions that were unsuccessful. To do this, an RTS includes a bitmap with n bits, where n is the number of receivers. The bitmap[i] is set to 1 for the unsuccessful receiver i . Only the failed receivers participate in the retransmissions and reply with CTSs/ACKs.

Wang proposed an extended implicit MAC acknowledgement (EIA) protocol in which ACK packets are eliminated and collisions of CTS packets are avoided [7]. In the EIA, ACK information is piggybacked in CTSs, so the control traffic can be decreased. A CTS is sent one after another in order. On the reception of an RTS, a receiver calculates its own CTS transmission time. This time is based on the priority assigned when a receiver joins the multicast group.

Kuri proposed a leader based protocol (LBP) in which one of the receivers is chosen as a leader for the purpose of sending a CTS and an ACK in response to an RTS and a data packet, respectively [8]. In the LBP, when a sender has a data packet to send, it transmits an RTS to receivers. If the leader receives the RTS successfully, it transmits a CTS to the sender. Otherwise, it does not send any response. Non-leader receivers transmit Not CTSs (NCTSSs) only when they receive an erroneous RTS. The NCTSSs will collide with the CTS sent by the leader. When the sender receives the CTS from the leader, it transmits a data packet to all receivers. The leader transmits an ACK if the data is correctly received, or otherwise transmits a NACK. On erroneous reception of the data packet, non-leader receivers transmit NACKs to

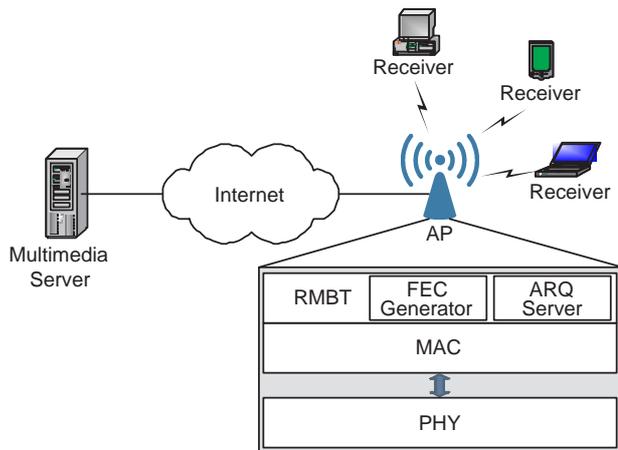


Fig. 1: System architecture

collide with the ACK from the leader, thus destroying the ACK and prompting the sender to retransmit the data packet.

The LBP has poor performance when the channel error rate is high because no sequence check is performed. In order to solve the problem, a BLBP (beacon-driven leader based protocol) is proposed [9]. The BLBP enhances the LBP with a special MAC control packet called a beacon. A beacon packet includes a sequence number from the data packet that follows it. A sender unicasts an RTS to a leader. When the RTS is successful, the leader transmits a CTS. After receiving the CTS, the leader transmits a beacon and a data packet to all receivers. The use of the beacon packet is to let non-leader receivers know the sequence number from the data packet that follows it.

3 RMBT Scheme

For reliable multicast transmissions, three important problems described in Section 1 should be solved: (1) how to decrease the excessive control overhead, (2) how to reduce a large number of packet transmissions, and (3) how to avoid the malfunctioning of receivers in the leader-based protocols. In the proposed scheme, these three problems are solved by two ways: FEC and busy tones.

3.1 System architecture

Figure 1 shows the proposed system architecture for providing reliable multicast transmissions to multimedia traffic in the IEEE 802.11 WLANs. In the architecture, data packets sent by the multimedia server are delivered through the Internet to the AP (access point), which is responsible for distributing them to all multicast receivers.

Octets: 2	2	6	6	6	1	1	1	0-2312	4
Frame Control	Duration	Address 1	Address 2	Address 3	Block Number	Block Size	Packet Index	DATA	CRC

Fig. 2: Data packet format in RMBT

The sender side of the proposed RMBT scheme is implemented in the AP and includes the FEC generator and the ARQ server. The FEC generator intercepts original multimedia packets and generates FEC parity packets used for error recovery based on block erasure codes denoted as $C(n, k)$. It then sends the parity packets along with the original packets to the ARQ server. In this paper, without distinguishing between original and parity packets, we call them data packets unless otherwise specified.

The ARQ server transmits the data packets to receivers. It first transmits only k original packets to receivers without any feedback from them. After the k th data packet transmission, it requests a feedback signal from the receivers. After receiving the data packets, and a packet has an error, each receiver performs error correction. If the received packets are not enough for error recovery, the receivers request an additional data packet transmission by using a busy tone after receiving the feedback request from the ARQ server. When the ARQ server receives the busy tone, it transmits data packets as many as the receivers request, and then also transmits a feedback request. The receivers again perform error correction and, on the reception of the feedback request, transmits a busy tone for requesting another data packet transmission if it is still not enough to recover error. This packet transmission process is repeated until all the receivers recover the original packets and do not request any more data packet transmission.

3.2 Protocol description

In this subsection, we describe the proposed protocol based on busy tones. Our protocol needs minor modifications to the unicast DCF. However, the basic operation of the proposed protocol is the same as that in the unicast DCF.

The format of the data packet in the RMBT is shown in Figure 2, which is modified from the standard data packet in the IEEE 802.11 WLANs. The format eliminates the sequence control and address4 fields and includes three new fields (block number, block size, and packet index) for identifying block order, block size, and data packet order in a block, respectively. The other fields are the same as those in the IEEE 802.11 WLANs.

For unicast transmission in the DCF, the RTS/CTS access method uses four-way handshaking (RTS-CTS-DATA-ACK) mechanism. However, the proposed protocol basically uses three-way handshaking (RTS-RTR-DATA) mechanism (see Figure 3). The CTS

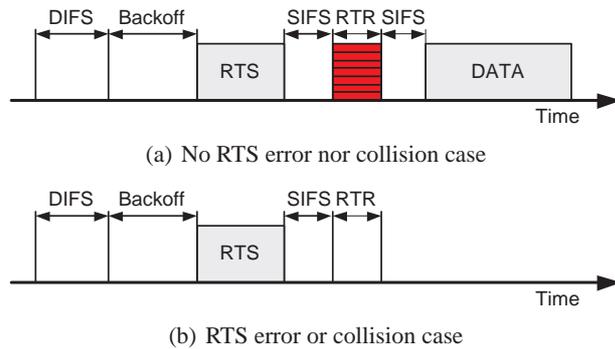


Fig. 3: Packet transmission operation in RMBT

used in the DCF is a packet, but the RTR (ready to receive) is a busy tone. Transmitting data packets involves transmission of special short RTS packet and RTR busy tone prior to the transmission of actual data packet. When a receiver successfully receives an RTS packet and is ready to receive a data packet, it transmits an RTR busy tone for a slot time (see Figure 3(a)). Otherwise, it does not transmit any response (see Figure 3(b)). Even if the RTRs from multiple receivers collide, it does not matter since the RTRs are only busy tones and, from the busy tones, a sender knows there is at least one receiver ready to receive a data packet. In the proposed protocol, all receivers do not acknowledge every data packet regardless of a successful transmission, an error, or a collision. Instead, the AP uses a feedback request busy tone, which will be explained later, for receivers to inform the AP of how many additional data packets are needed to recover data packets.

When the AP has a multicast data packet to send, it operates like the unicast DCF. If the channel is free for DIFS period, the AP decreases its backoff counter as long as the channel is idle, does not decrease when a transmission is detected on the channel, and tries to transmit an RTS packet when the backoff counter reaches zero. If the channel is determined to be busy at any time, then the backoff procedure is suspended. After sending an RTS packet, the AP does not expect to receive a CTS packet. Instead, it listens to the channel to see if any receiver transmits an RTR busy tone. If the AP senses the RTR busy tone, it transmits a multicast data packet to receivers (see Figure 3(a)). Otherwise, it again initiates its backoff procedure and repeats the process (see Figure 3(b)).

The diagram of a block transmission between the AP and receivers is shown in Figure 4. In the figure, TX i means the transmission of the i th data packet in a block, and is made up of five components: 1) DIFS; 2) backoff; 3) RTS; 4) RTR; and 5) DATA (see Figure 3). The AP first transmits only k original packets to receivers. And then it transmits a feedback request (FR) busy tone for two time slots after SIFS interval immediately following

the transmission of the k th data packet. An FR busy tone needs for a receiver to decide when to transmit a feedback signal. After receiving the FR busy tone, a receiver waits for a SIFS interval and transmits a packet request (PR) busy tone back to the AP if it has received less than k correct packets for the current block. Otherwise, it does nothing. The duration of the PR busy tone depends on how many additional packets a receiver needs to recover error. If a receiver has received i ($< k$) correct packets, the duration is $(k - i) \cdot aSlotTime$, where $aSlotTime$ is the duration of a slot time. After transmitting an FR busy tone, the AP expects PR busy tones from receivers whose transmissions were unsuccessful. In case the AP does not sense PR busy tones, the transmission of the current block is assumed to be completed. Otherwise, from the duration of the PR busy tones, the AP knows how many additional data packets the receivers need to recover error and then transmits as many data packets as the receivers request. This procedure is repeated until all the receivers recover the original packets and do not request any more packet transmission.

Figure 4 gives an example of the procedure described above. The AP transmits k data packets. Receivers 1 and 2 receive one erroneous data packet (TX 2) and two erroneous data packets (TX 1 and TX 2), respectively. Therefore, after receiving an FR busy tone from the AP, they transmit PR busy tones back to the AP, of which durations are one time slot for receiver 1 and two time slots for receiver 2. From the PR busy tones, the AP knows that the receivers need two additional data packets to recover error and then transmit two packets. The receivers receive the data packets and perform error correction again. After receiving an FR busy tone, receiver 2 transmits a PR busy tone for a time slot since one data packet is still not enough to recover error. However, receiver 1 does not transmit a PR busy tone because it has enough packets ($\geq k$) to recover. The AP transmits a data packet for receiver 2 and an FR busy tone. At this point, none of the receivers transmit any PR busy tones which results in the AP not sensing any busy tones. Therefore, the transmission of the current block is completed.

In the proposed protocol, we use an RTR busy tone and a FR busy tone. We assume they have different durations to avoid the malfunctioning of the receivers. If the durations were the same, receivers would not know exactly when to transmit their PR busy tones and would transmit them after receiving RTR busy tones from other receivers. Distinguishing an RTR busy tone and an FR busy tone from a packet transmission is very important to guarantee the proper operation. To do this, the duration of a transmission is used. The transmission time for a packet has the duration of at least three time slots, because it includes the physical preamble and header of 20 μs , which is from Table 1 in Section 5. The durations of an RTR busy tone and an FR busy tone are one time slot and two time slots, respectively. Estimating the duration is simple without any additional overhead or cost, because

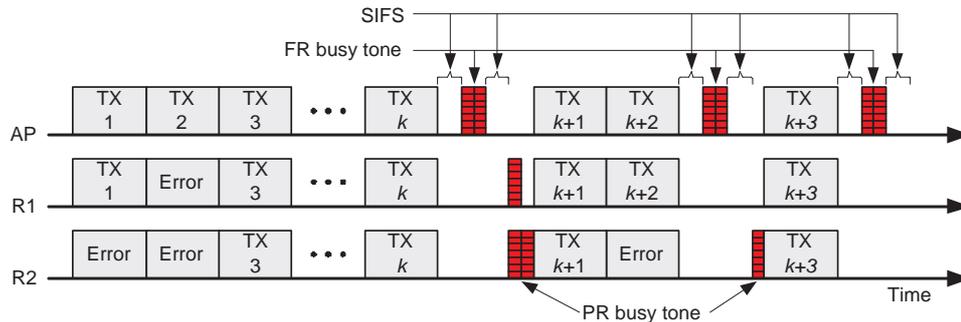


Fig. 4: Block transmission between the AP and receivers

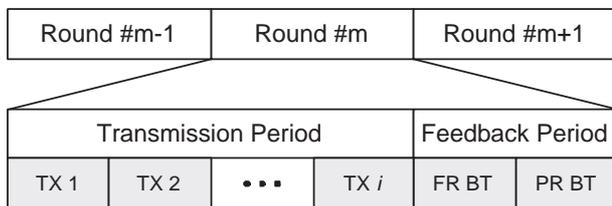


Fig. 5: Round structure in RMBT

every receiver performs carrier sensing. Each receiver, by using carrier sensing, observes the channel status and measures the duration of the busy period. Therefore, the proposed scheme can discriminate between busy tones and a packet transmission when receiving a signal.

4 Performance Analysis

In this section, we analyze the normalized throughput of the proposed scheme, which is defined as the fraction of time during which the channel is used to successfully transmit data packets in a block to all receivers.

We assume that there is no spatial and temporal correlation between the packet errors of multiple receivers. We use the random uniform packet error model. It is also assumed that an RTS packet is error-free, there are no unicast transmissions, and only the AP transmits multicast packets. Therefore, an RTS packet is always successful without error and collision.

For transmitting a multimedia block, the time axis is divided into *rounds* (see Figure 5). Each of them consists of two parts: transmission period (TP) and feedback period (FP). A TP is composed of multiple data packet transmissions (TXs). In an FP, there are an FR BT (busy tone) from the AP and a PR BT from receivers as shown in Figure 4. During each round, the AP transmits data packets in a TP, an FR busy tone in an FP, and then awaits feedback from receivers requesting additional data packets before proceeding to the next round.

Let us denote the normalized throughput by S , which is given by

$$S = \frac{k \cdot L}{N \cdot TX_{duration} + FB_{duration}}, \quad (1)$$

where N is the total number of data packets transmitted in a block, $TX_{duration}$ is the average transmission time interval of a packet, $FB_{duration}$ is the amount of average time intervals used for feedback, k is the size of a block, and L is the data packet size in time.

The receivers are supposed to be a fixed number R , and data packet error rate is P_{error} . The number of data packets, which will be transmitted during the m th round, is predetermined at the end of a feedback period in the $(m - 1)$ th round. Let $T_r(m)$ denote the maximum number of data packets requested by r receivers after the m th round. The number of data packets requested by a random receiver after the first round ($T_1(1)$) is a binomial random variable and its probability mass function is given by

$$P\{T_1(1) = i\} = P_{error}^i \cdot (1 - P_{error})^{k-i}. \quad (2)$$

In Appendix, we derive the probability mass function of $T_1(m)$ which is the number of data packets requested by a random receiver after the m th round.

The number of packets transmitted for a block (N) is given by

$$N = k + \sum_{m=2}^{\infty} \sum_{i=0}^k i \cdot P\{T_R(m-1) = i\}, \quad (3)$$

where

$$\begin{aligned} P\{T_R(m-1) = i\} &= P\{T_R(m-1) \leq i\} - P\{T_R(m-1) \leq i-1\} \\ &= [P\{T_1(m-1) \leq i\}]^R - [P\{T_1(m-1) \leq i-1\}]^R. \end{aligned}$$

The average transmission time interval of a packet ($TX_{duration}$) is $(DIFS + Backoff + DATA)$, where $DIFS$ is the DIFS time, $Backoff$ is the average backoff time when any transmission occurs, and $DATA$ is the average time interval that the channel is sensed busy due to a

packet transmission. In this analysis, since we assumed that only the AP transmits data packets, there are no collisions and the AP does not double its contention window. We can easily obtain *Backoff* as

$$Backoff = \frac{CW_{min}}{2} \cdot \sigma, \quad (4)$$

where CW_{min} is the minimum contention window and σ is the duration of a time slot. Also, *DATA* is given by

$$DATA = RTS + SIFS + RTR + SIFS + H + L + 3\delta, \quad (5)$$

where $H (= PHYhdr + MAChdr)$ is the time to transmit a packet header, δ is propagation delay, *SIFS* is the SIFS time, *RTS* is the time to transmit an RTS, and *RTR* is the time duration for an RTR busy tone (i.e., whose value is σ).

The amount of average time intervals used for feedback ($FB_{duration}$) is given by

$$FB_{duration} = \sum_{m=1}^{\infty} \sum_{i=0}^k B(i) \cdot P\{T_R(m) = i\}, \quad (6)$$

where

$$B(i) = \begin{cases} SIFS + 2 \cdot \sigma + SIFS + \sigma, & i = 0 \\ SIFS + 2 \cdot \sigma + SIFS + i \cdot \sigma, & i \geq 1. \end{cases}$$

5 Performance Evaluation

In this section, we discuss the numerical and simulation results of the proposed RMBT scheme. To validate the proposed scheme, we compare them to the results of the LBP+FEC, in which we apply FEC to the LBP. The LBP+FEC generates FEC parity packets used for error recovery based on block erasure codes, and uses the same data packet format as shown in Figure 2. If receivers receive an erroneous packet in the LBP+FEC, then they are programmed not to send any response; otherwise, they send feedback signals accordingly as follows: After receiving each data packet, the leader transmits an ACK if at least k data packets are correctly received, or otherwise transmits a NACK. When receiving less than k correct data packets, non-leader receivers transmit NACKs to collide with the ACK from the leader, or do not transmit any response otherwise. The AP assumes that the transmission of a block is completed when receiving an ACK from the leader.

System parameters used in the numerical analysis and simulation are listed in Table 1. In the IEEE 802.11 WLANs, multicast data packets are transmitted at the lowest transmission rate in order to provide a multicast service to all receivers. However, we simulated an IEEE 802.11a network with transmission rates of 54Mbps for data packets and of 6Mbps for control packets such as

Table 1: Numerical and simulation parameters

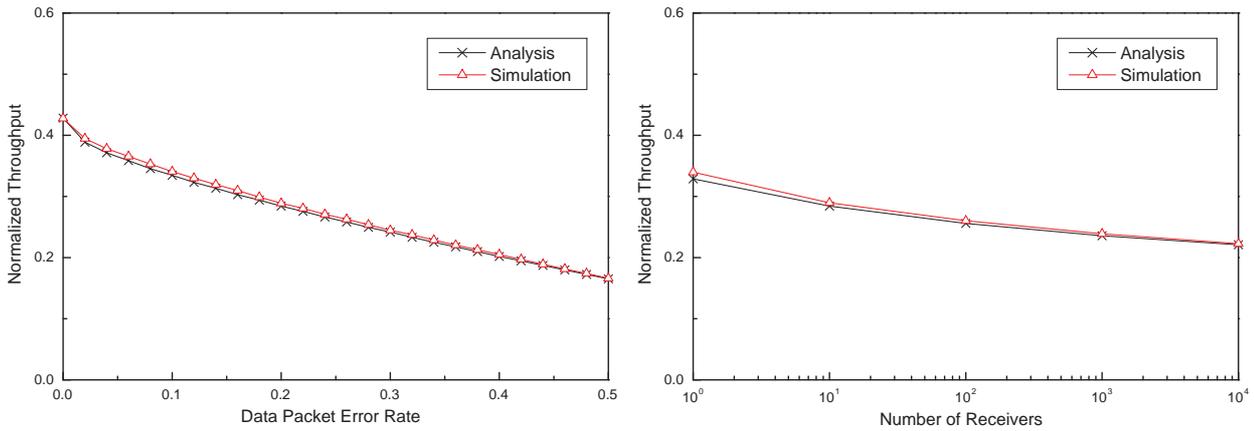
Parameter	Value
Data Bit Rate	54 Mbps
Control Bit Rate	6 Mbps
Slot Time	9 μ s
SIFS	16 μ s
Propagation Delay	1 μ s
MAC Header	25 Octets
CRC	4 Octets
PHY PLCP Preamble Length	16 μ s
PHY PLCP Header Length	5 Octets
Block Size	20 Packets
Packet Size	1500 Octets
CW_{min}	31

RTS, CTS, and ACK, respectively. In the simulation, we consider the topological reason for the fact that there are no unicast transmissions, and only the AP transmits multicast packets. Therefore, there are no collisions. In addition, we assume that the data packet error rate is independent among receivers and the error rate for control packets is 20% of the error rate given for data packets unless otherwise specified.

Main performance metrics of interest are normalized throughput, average delay, number of uncompleted receivers, and number of insufficient packets. Delay is the time elapsed from the moment a block arrives at the MAC layer queue until the block is successfully transmitted to all the intended receivers. The number of uncompleted receivers is the number of receivers with less than k correct packets when the AP finds that the transmission of a block is completed, and stops transmitting additional data packets for the current block. The number of insufficient packets is the number of data packets each uncompleted receiver needs to recover the block (i.e., k - number of data packets correctly received).

Figure 6 shows the numerical and simulation results for normalized throughput. In the figure, the error rate for control packets is 0 regardless of the error rate for data packets. We can see that the figure shows very close match between numerical and simulation results. Figure 6(a) is the result when the number of receivers is 10. Normalized throughput decreases as the data packet error rate becomes larger. Receivers lose more data packets in a block and need more additional data packets to recover the block as the data packet error rate gets higher. Therefore, the AP has to transmit more additional data packets. This results in lower normalized throughput. Figure 6(b) is the result when the data packet error rate is 0.2. This figure is similar to Figure 6(a). As the number of receivers grows, the number of transmitted packets also increases since the probability that all the receivers successfully receive packets decreases. It causes the waste of bandwidth and degrades the normalized throughput.

Figure 7 shows the performance for the different data packet error rates. The number of receivers is set to 10.



(a) Normalized throughput with different data packet error rates when the number of receivers is 10 (b) Normalized throughput with different number of receivers when the data packet error rate is 0.2

Fig. 6: Numerical and simulation performance

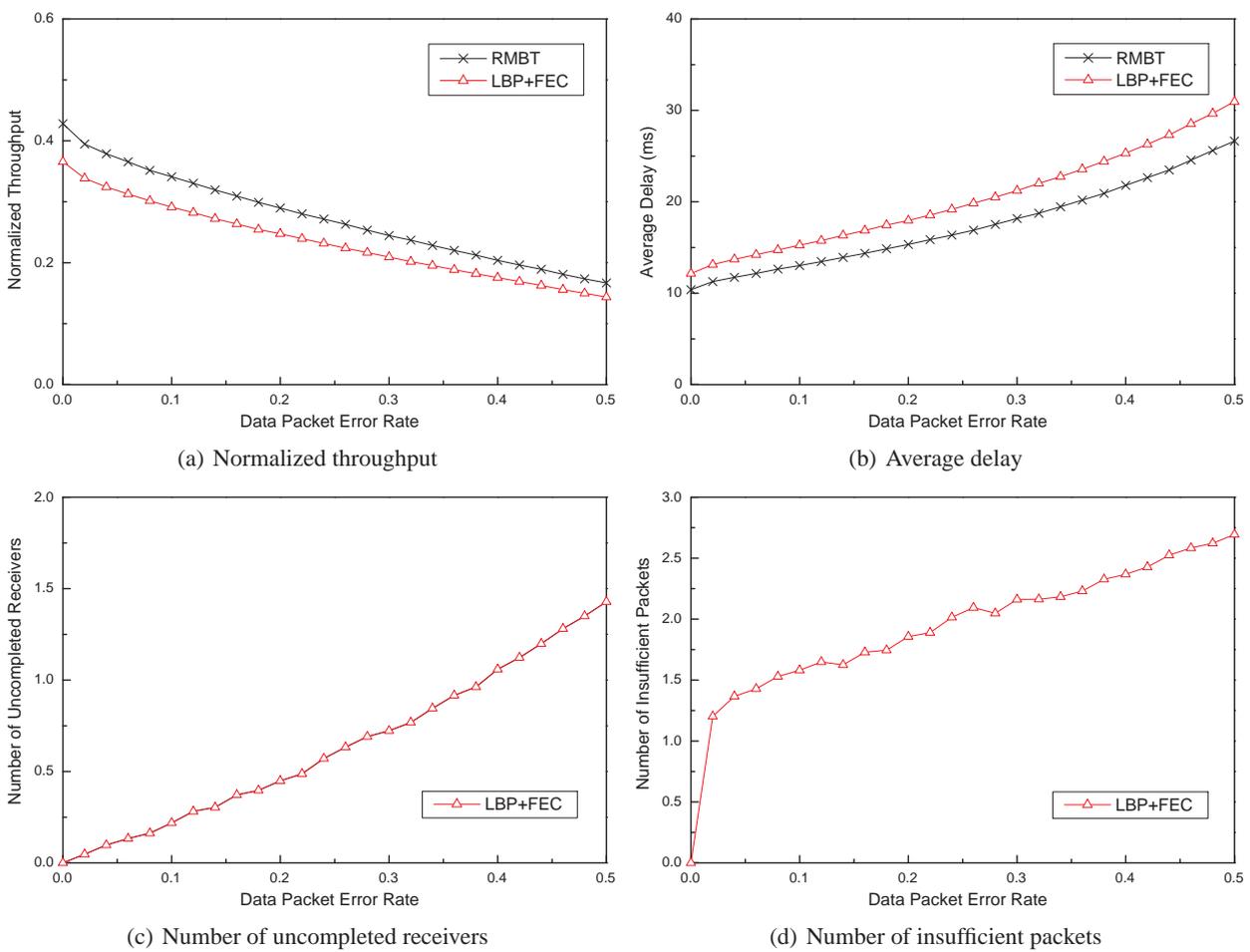


Fig. 7: Performance according to the data packet error rate with 10 receivers

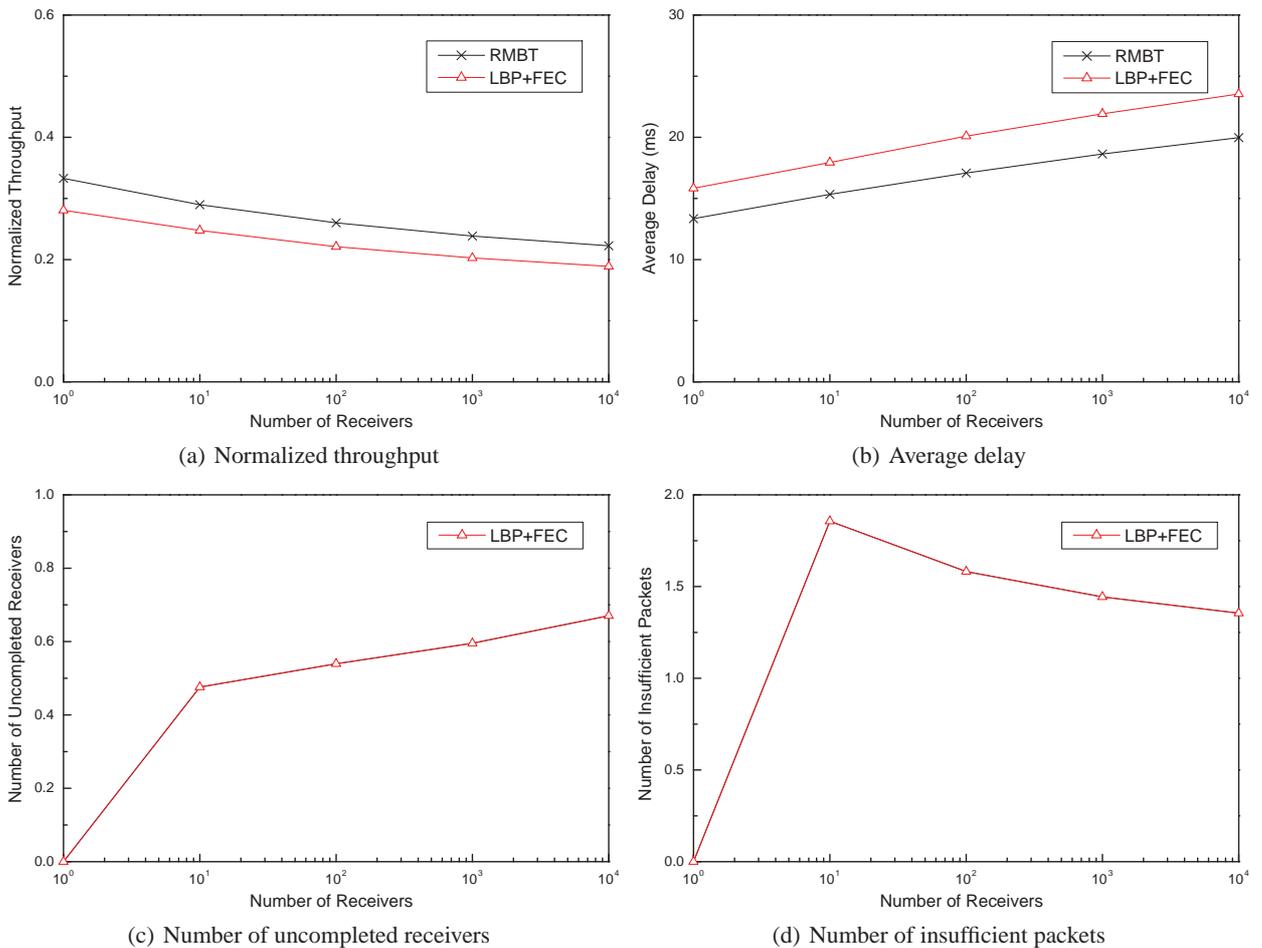


Fig. 8: Performance according to the number of receivers with data packet error rate = 0.2

Figure 7(a) depicts the normalized throughput. As explained in Figure 6(a), the throughput of both schemes decreases as the data packet error rate increases. However, the RMBT has about 4% higher throughput compared to the LBP+FEC regardless of the data packet error rate. The LBP+FEC always needs a feedback signal from receivers for each data packet, and CTS packets from the leader may be erroneous. However, the RMBT needs only one feedback signal for each round, and RTR busy tones are never lost. Therefore, the RMBT has low control overhead. This is what causes the performance difference between them.

Figure 7(b) is the results for the average delay. With the increasing data packet error rate, the average delay gets worse since more packets are necessary to complete the transmission of a block. The delay difference between the RMBT and LBP+FEC comes from the reason explained in Figure 7(a).

Figures 7(c) and 7(d) show the effect of data packet error rate on the number of uncompleted receivers and the number of insufficient packets. In the RMBT, all the

receivers execute transmit their PR busy tones when FR busy tone occurs, even when they receive erroneous data packets. Therefore, the RMBT can avoid the malfunctioning of receivers and has no uncompleted receivers and insufficient packets. Thus, here we omit the results of the RMBT. However, LBP+FEC receivers do not know when to transmit ACKs/NACKs when receiving erroneous packets. The leader transmits an ACK after receiving the k th correct packet, whereas non-leader receivers with less than k correct packets do not transmit NACKs when receiving an erroneous packet. In this case, the AP receives the ACK from the leader, and finds that the transmission of a block is completed, and then stops transmitting additional data packets for the current block. Therefore, some receivers do not receive enough data packets. From Figures 7(c) and 7(d), we can see that the results increase linearly with the increasing data packet error rate. For the data packet error rate of 0.2, about 5% of receivers cannot recover the current block and each of them needs about two packets to recover the block. This

is very serious for providing a multicast service for the IEEE 802.11 WLANs.

Figure 8 shows the performance for the different number of receivers. The data packet error rate is set to 0.2. The more receivers there are, the harder it is for all the receivers to successfully receive packets. Therefore, both schemes result in a low performance level. From Figures 8(a) and 8(b), we can observe that the normalized throughput and average delay for the RMBT scheme are better than those for the LBP+FEC. The reason for this is that the RMBT has lower control overhead than the LBP+FEC as explained in Figure 7(a). From Figure 8(c), it can be observed that the number of uncompleted receivers increases slightly when the number of receivers increases. Figure 8(d) represents the number of insufficient packets. For the given data packet error rate, the possibility that all the uncompleted receivers receive an erroneous packet at the same time becomes smaller as the number of uncompleted receivers increases, thus the number of insufficient packets also becomes decreases.

6 Conclusion

For multicast transmissions, IEEE 802.11 MAC layer has no control packets such as RTS/CTS/ACK used for unicast transmissions. Therefore, it does not provide for reliable multicast transmissions. In this paper, we proposed a scheme to support the reliable multicast transmission over the MAC layer. The proposed RMBP scheme combines FEC and ARQ mechanisms to provide data reliability, and uses busy tones to avoid the malfunctioning of receivers, and to improve efficiency by reducing the excessive control overhead. The simulation results show that the RMBT is very efficient and has higher normalized throughput and lower delay by eliminating the uncompleted receivers regardless of the data packet error rate and the number of receivers.

Appendix

In the appendix, we derive the probability mass function of $T_1(m)$ which is the number of data packets requested by a random receiver after the m th round.

The cumulative distribution functions of $T_{R-1}(m)$ and $T_R(m)$ for $m > 1$ are given by [13]

$$P\{T_{R-1}(m) \leq i\} = [P\{T_1(m) \leq i\}]^{R-1}, \quad (7)$$

$$P\{T_R(m) \leq i\} = [P\{T_1(m) \leq i\}]^R, \quad (8)$$

where $P\{T_1(m) \leq i\} = \sum_{t=0}^i P\{T_1(m) = t\}$.

The probability mass function of $T_1(m)$ can be derived from $T_1(m-1)$ and $T_{R-1}(m-1)$ as [18]

$$P\{T_1(m) = x\} = \sum_{i=x}^k \sum_{j=0}^k P\{T_1(m) = x, T_1(m-1) = i, T_{R-1}(m-1) = j\} \quad (9)$$

Because we assume that packet errors are both spatially and temporally independent among receivers, the number of packets requested by each receiver is also independent. Therefore, (9) can be rewritten as

$$\begin{aligned} P\{T_1(m) = x\} &= \sum_{i=x}^k P\{T_1(m-1) = i\} \cdot \sum_{j=0}^k P\{T_{R-1}(m-1) = j\} \\ &\quad \cdot P\{T_1(m) = x \mid T_1(m-1) = i, T_{R-1}(m-1) = j\} \\ &= \sum_{i=x}^k P\{T_1(m-1) = i\} \cdot [P\{T_{R-1}(m-1) \leq i\} \\ &\quad \cdot P\{T_1(m) = x \mid T_1(m-1) = i, T_{R-1}(m-1) \leq i\} \\ &\quad + \sum_{j=i+1}^k P\{T_{R-1}(m-1) = j\} \\ &\quad \cdot P\{T_1(m) = x \mid T_1(m-1) = i, T_{R-1}(m-1) = j\}] \\ &= \sum_{i=x}^k P\{T_1(m-1) = i\} \cdot [P\{T_{R-1}(m-1) \leq i\} \\ &\quad \cdot \binom{i}{x} P_{error}^x \cdot (1 - P_{error})^{i-x} \\ &\quad + \sum_{j=i+1}^k P\{T_{R-1}(m-1) = j\} \cdot \theta(i, j, x)] \\ &= \sum_{i=x}^k P\{T_1(m-1) = i\} \cdot [[P\{T_1(m-1) \leq i\}]^{R-1} \\ &\quad \cdot \binom{i}{x} P_{error}^x \cdot (1 - P_{error})^{i-x} \\ &\quad + \sum_{j=i+1}^k [P\{T_{R-1}(m-1) \leq j\} \\ &\quad - P\{T_{R-1}(m-1) \leq j-1\}] \cdot \theta(i, j, x)] \\ &= \sum_{i=x}^k P\{T_1(m-1) = i\} \cdot [[P\{T_1(m-1) \leq i\}]^{R-1} \\ &\quad \cdot \binom{i}{x} P_{error}^x \cdot (1 - P_{error})^{i-x} \\ &\quad + \sum_{j=i+1}^k [[P\{T_1(m-1) \leq j\}]^{R-1} \\ &\quad - [P\{T_1(m-1) \leq j-1\}]^{R-1}] \cdot \theta(i, j, x)], \quad (10) \end{aligned}$$

where

$$\theta(i, j, x) = \begin{cases} \sum_{l=0}^{j-i} \binom{j}{l} P_{error}^l \cdot (1 - P_{error})^{j-l}, & x = 0 \\ \binom{j}{j-(i-x)} P_{error}^{j-(i-x)} \cdot (1 - P_{error})^{i-x}, & x \geq 1. \end{cases}$$

Acknowledgement

This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract.

References

- [1] IEEE, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11, (1999).
- [2] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath, DirCast: a practical and efficient Wi-Fi multicast system, in Proc. IEEE ICNP'09, 161-170 (2009).
- [3] D. Dujovne and T. Turetli, Multicast in 802.11 WLANs: an experimental study, in Proc. ACM MSWiM'06, 130-138 (2006).
- [4] K. Tang and M. Gerla, MAC reliable broadcast in ad hoc networks, in Proc. IEEE MILCOM'01, 2, 1008-1013 (2001).
- [5] M. T. Sun, L. Huang, A. Arora, and T. H. Lai, Reliable MAC layer multicast in IEEE 802.11 wireless networks, Wireless Communications and Mobile Computing, 3, 439-453 (2003).
- [6] V. Srinivas and L. Ruan, An efficient reliable multicast protocol for 802.11-based wireless LANs, in Proc. IEEE WOWMOM'09, 1-6 (2009).
- [7] X. Wang, L. Wang, Y. Wang, and D. Gu, Reliable multicast mechanism in WLAN with extended implicit MAC acknowledgement, in Proc. IEEE VTC'08, 2695-2699 (2008).
- [8] J. Kuri and S. K. Kasera, Reliable multicast in multi-access wireless LANs, Wireless Networks, 7, 359-369 (2001).
- [9] Z. Li and T. Herfet, BLBP: a beacon-driven leader based protocol for MAC layer multicast error control in wireless LANs, in Proc. WiCom'08, 1-4 (2008).
- [10] L. Rizzo, Effective erasure codes for reliable computer communication protocols, ACM Computer Communication Review, 27, 24-36 (1997).
- [11] P. K. McKinley, C. Tang, and A. P. Mani, A study of adaptive forward error correction for wireless collaborative computing, IEEE Transactions on Parallel and Distributed Systems, 13, 936-947 (2002).
- [12] L. Rizzo and L. Vicisano, RMDP: an FEC-based reliable multicast protocol for wireless environments, ACM SIGMOBILE Mobile Computing and Communications Review, 2, 23-31 (1998).
- [13] H. Djangji, An efficient hybrid ARQ protocol for point-to-multipoint communication and its throughput performance, IEEE Transactions on Vehicular Technology, 48, 1688-1698 (1999).
- [14] H. Liu, S. Mathur, S. Makharia, D. Li, and M. Wu, IPTV multicast over wireless LAN using merged hybrid ARQ with staggered adaptive FEC, IEEE Transactions on Broadcasting, 55, 363-374 (2009).
- [15] A. Majumdar, D. G. Sachs, I. V. Kozintsez, K. Ramchandran, and M. Yeung, Multicast and unicast real-time video streaming over wireless LANs, IEEE Trans. CSVT, 12, 524-534 (2002).
- [16] Z. J. Haas and J. Deng, Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks, IEEE Trans. Commun., 50, 975-985 (2002).
- [17] S. Kim, R. Huang, and Y. Fang, Deterministic priority channel access scheme for QoS support in IEEE 802.11e wireless LANs, IEEE Transactions on Vehicular Technology, 58, 855-864 (2009).
- [18] S.W. Yuk and D.H. Cho, Parity-based reliable multicast method for wireless LAN environments, in Proc. IEEE VTC'99, 2, 1217-1221 (1999).



Sunmyeng Kim

received the B.S., M.S., and Ph.D. degrees in information and communication from Ajou University in 2000, 2002, and 2006, respectively. From May 2006 to February 2008, he was a Postdoctoral Researcher in electrical and computer engineering with the University of Florida, Gainesville. Since 2008, he has been a professor of the Department of Computer Software Engineering at Kumoh National Institute of Technology. His research interests include resource management, wireless LANs and PANs, wireless mesh networks, and quality-of-service enhancement.



Young-Jong Cho

received the B.S. degree in Electronic Engineering from Seoul National University, in 1983 and M.S and Ph.D. degrees in Electronic Engineering from KAIST in 1985 and 1989, respectively. Since 1996, he has been a professor of the Department of Information and Computer at Ajou University. From 1985 to 1996, he had been a principal engineer at LG Co., Korea. In 1993, he joined the AT&T Bell Lab. as a researcher and in 2003, he was a visiting professor at GMU. His research interests include performance analysis of broadband communication systems and development of high speed Internet devices.