Design of Efficient Multicast Protocol for IEEE 802.11n WLANs and Cross-Layer Optimization for Scalable Video Streaming

Wan-Seon Lim, *Member*, *IEEE*, Dong-Wook Kim, and Young-Joo Suh, *Member*, *IEEE*

Abstract—The legacy multicasting over IEEE 802.11-based WLANs has two well-known problems—poor reliability and low-rate transmission. In the literature, various WLAN multicast protocols have been proposed in order to overcome these problems. Existing multicast protocols, however, are not so efficient when they are used combining with the frame aggregation scheme of IEEE 802.11n. In this paper, we propose a novel MAC-level multicast protocol for IEEE 802.11n, named Reliable and Efficient Multicast Protocol (REMP). To enhance the reliability and efficiency of multicast services in IEEE 802.11n WLANs, REMP enables selective retransmissions for erroneous multicast frames and efficient adjustments of the modulation and coding scheme (MCS). In addition, we propose an extension of REMP, named scalable REMP (S-REMP), for efficient delivery of scalable video over IEEE 802.11n WLANs. In S-REMP, different MCSs are assigned to different layers of scalable video to guarantee the minimal video quality to all users while providing a higher video quality to users exhibiting better channel conditions. Our simulation results show that REMP outperforms existing multicast protocols for normal multicast traffic and S-REMP offers improved performance for scalable video streaming.

Index Terms—Multicast, IEEE 802.11n, scalable video coding, cross-layer optimization.

1 Introduction

The IEEE 802.11 wireless local area network (WLAN) [1] has been one of the most popular wireless access technologies due to its advantages of the high data rate, low cost, and easy deployment. Recently, most mobile devices such as laptops, PDAs, and mobile phones equip the WLAN interface for low cost and high speed Internet connectivity. In IEEE 802.11n [2], the physical layer (PHY) data rate increases to 600 Mbps. Also, IEEE 802.11n introduces an enhanced medium access control (MAC) protocol in order to reduce MAC layer overhead. One of key MAC enhancements of 802.11n is the aggregate MAC protocol data unit (A-MPDU) aggregation, which maximizes the throughput at the MAC layer.

With the successful deployment of IEEE 802.11 WLANs and increase in applications that require multicast services such as IPTV and Internet streaming, multicast communications over IEEE 802.11 WLANs have received much attention. However, there are two well-known problems in the multicast protocol of the IEEE 802.11 standard. First, multicast frames are transmitted as a simple broadcasting mechanism without acknowledgments from receivers. Due to the absence of automatic repeat request (ARQ) mechanisms, the reliability of multicast frames cannot be guaranteed, especially when the probability of collisions or bit errors is high. Second, a low and fixed transmission rate is used for multicast transmissions. Although there have been several rate adaptation mechanisms for unicast transmissions in WLANs, they cannot be directly applied to multicast transmissions since the sender does not receive any feedbacks from receivers.

In order to improve the reliability of multicast transmissions in WLANs, various ARQ mechanisms have been proposed [6], [7], [8], [9], [10], [11], [18], [19], [20], [21]. They can be classified into individual ARQ for each multicast receiver, unicast-like ARQ, and negative acknowledgment-based (NAK-based) ARQ. Due to its effectiveness, the NAK-based ARQ mechanism is the most widely adopted to various 802.11 multicast protocols. In addition, for using the highest possible data rate enabling the provisioning of reliable multicast transmissions, several rate adaptation mechanisms have been proposed. In those mechanisms, the data rate is determined by either local acknowledgment information at the sender [9] or explicit feedbacks from receivers [10], [11].

In this paper, we focus on multicast transmissions over 802.11n WLANs. Most of existing multicast protocols for WLANs have been designed based on the legacy 802.11 MAC. Since 802.11n MAC has a backward compatibility with the legacy 802.11 MAC, the existing multicast protocols can be used in 802.11n WLANs as well. However, unfortunately, the existing multicast protocols have serious problems in 802.11n WLANs when the A-MPDU aggregation is used for multicast transmissions. First, the reliable multicast protocols that use the NAK-based ARQ mechanism may result in unnecessary retransmissions for A-MPDU frames. This is because the block acknowledgment mechanism for A-MPDU transmissions conflicts with the NAK-based ARQ mechanism. Besides, rate adaptation mechanisms used in existing multicast protocols cannot select appropriate data rates for A-MPDU transmissions

• *The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-dong, Nam-gu, Pohang 790-784, Republic of Korea. E-mail: {kiki, shkim80, yjsuh}@postech.ac.kr.*

Manuscript received 18 Oct. 2010; revised 17 Feb. 2011; accepted 18 Mar. 2011; published online 28 Apr. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-10-0481.
Digital Object Identifier no. 10.1109/TMC.2011.95.

TABLE 1
Summary of Acronyms

A-MPDU	Aggregate MAC Protocol Data Unit
ARQ	Automatic Repeat Request
BA	Block Acknowledgement
MBA	Multicast Block Acknowledgement
MCA	Multicast Channel Acknowledgement
MCS	Modulation and Coding Scheme
MFR	Multicast Feedback Request
MPDU	MAC Protocol Data Unit
MTA	Multicast Transmission Announcement
NAK	Negative Acknowledgment
RIFS	Reduced Interframe Space

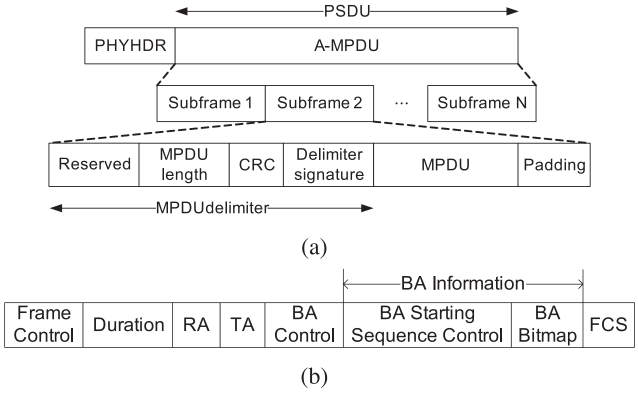

Fig. 1. MAC frame format in IEEE 802.11n. (a) A-MPDU. (b) Block Ack.

since they have been designed based on the legacy data transmission (i.e., single MPDU transmission.).

In order to improve the performance of multicast services in IEEE 802.11n WLANs, we propose a novel MAC-level multicast protocol, named Reliable and Efficient Multicast Protocol (REMP). By considering the A-MPDU aggregation, we introduce an advanced feedback mechanism for multicast transmissions. Basically, when the channel condition is stable, an AP exchanges control frames with a selected multicast receiver for each A-MPDU transmission. On the other hand, when the channel condition is dynamic, the AP exchanges control frames with all the multicast receivers. Based on feedbacks in the control frames, the AP selectively retransmits erroneous multicast frames and efficiently adjusts the modulation and coding scheme (MCS) under varying channel conditions.

Although REMP can provide reliable multicast services to multicast receivers, it may heavily penalize receivers exhibiting better channel conditions. To handle this issue, we further consider the use of the scalable video coding technique that supports the flexible adaptation of video quality based on a hierarchical layer structure. In this paper, we extend REMP for scalable video streaming over IEEE 802.11n WLANs via *cross-layer* optimization. The extension of REMP, referred to as scalable REMP (S-REMP), aims at guaranteeing the minimal video quality to all users while providing a higher video quality to users exhibiting better channel conditions. In S-REMP, different layers of scalable video are transmitted with different MCSs according to the network load.

The remainder of this paper is organized as follows. In Section 2, we describe background and related work of this paper. Section 3 presents the motivation of this work. Sections 4 and 5 describe the protocol operations of REMP and S-REMP, respectively. The performance of the proposed protocols is evaluated through extensive simulations in Section 6, and then we close the paper with concluding remarks in Section 7. The acronyms used in this paper are summarized in Table 1.

2 Background and Related Work

In this section, we first introduce the frame aggregation mechanism, which is one of the key features of the IEEE 802.11n MAC protocol. Then, we introduce enhanced multicast mechanisms for IEEE 802.11 WLANs and describe why they have limitations for using in IEEE 802.11n WLANs.

2.1 Frame Aggregation in IEEE 802.11n

IEEE 802.11n supports up to 600 Mbps at the PHY layer by using the MIMO technology and wider channel bandwidth. With the current 802.11 MAC, however, the maximum achievable throughput is bounded to 50 Mbps when the frame size is 1 KB, although the PHY rate increases to infinite [14]. This throughput limitation is due to the large overhead of IEEE 802.11 MAC. Frame aggregation is one of the main techniques in 802.11n to overcome the large overhead. There are two methods to perform frame aggregation: aggregate MAC protocol service unit (A-MSDU) and aggregate MAC protocol data unit. In A-MSDU aggregation, it is allowed to concatenate multiple MSDUs in a single MPDU. A-MSDU aggregation improves the efficiency of the MAC layer when there are many small MSDUs. However, in error-prone channels, the entire A-MSDU should be retransmitted when there are corruptions in any subframes.

In A-MPDU aggregation, multiple MPDU subframes are joined with a single leading PHY header. In contrast to A-MSDU, each subframe in A-MPDU is verified by its own FCS field and acknowledged individually by Block ACK (BA). Fig. 1a shows frame formats of an A-MPDU frame. For A-MPDU, delimiters are inserted before each MPDU and padding bits are added at the tail. After processing the whole A-MPDU, the receiver sends BA to the sender. Fig. 1b shows frame formats of a BA frame. The Block Ack Starting Sequence Control field contains the first sequence number for which this BA is sent, and the BA Bitmap field is used to indicate the received status of up to 64 MPDUs.

2.2 ARQ Mechanisms for Multicast

In IEEE 802.11 WLANs, multicast frames are simply broadcasted without ARQ mechanisms. Therefore, multicast frame losses frequently occur due to channel errors or collisions. To overcome this limitation, various ARQ mechanisms have been proposed. As mentioned in Section 1, they are classified into three categories: individual ARQ for each multicast receiver [6], [7], [18], unicast-like ARQ [19], [20], and NAK-based ARQ [8], [9], [10], [11], [21].

In the individual ARQ mechanism, each multicast receiver sends CTS and ACK[1] to an AP for a multicast data

1. In infrastructure mode WLANs, AP becomes a multicast sender in usual cases. We focus on the infrastructure mode only in this paper.

frame transmission. Tang et al. proposed Broadcast Medium Window (BMW) that treats a multicast transmission as multiple unicast transmissions to each receiver [6]. In the Batch Mode Multicast MAC protocol (BMMM) [7], an AP sequentially exchanges RTS/CTS with receivers and then transmits a multicast data frame. In Slot Reservation based Reliable Multicast protocol (SBR) [18], an AP schedules CTS and ACK transmissions of receivers for minimizing the control frame overhead sent by the AP. However, SBR still needs n times CTS and ACK transmissions for a data transmission where n receivers are members of a target multicast group.

In the unicast-like ARQ mechanism [19], [20], only a selected receiver for a multicast group sends CTS and ACK to an AP while other receivers just overhear transmissions without sending control frames. Although this approach can reduce the control frame overhead significantly compared to the individual ARQ mechanism, it just provides semi-reliability. When the channel condition is dynamic, data losses may frequently occur at nonselected receivers.

The NAK-based ARQ mechanism is first proposed in the Leader-Based Protocol (LBP) [8]. The main idea of NAK-based ARQ mechanism is reducing the control frame overhead by allowing an AP to receive multiple feedbacks from receivers in a single time slot. In LBP, an AP selects a receiver for a multicast group as a leader. After a successful reception of the data frame, the leader sends ACK in reply while other receivers do nothing. In contrast, after the reception of an erroneous data frame, both the leader and other receivers send NAK. As a result, if the data frame was successfully delivered to all receivers, the AP receives ACK from the leader. Otherwise, if one or more receivers did not receive the data frame, then the AP receives NAK or corrupted frame. The NAK-based ARQ mechanism of LBP provides reliable multicast data transmissions with small control frame overhead. As a results, it is widely adopted to various multicast protocols [9], [10], [11], [21].

However, combining the NAK-based ARQ with the A-MPDU aggregation results in inefficient retransmissions. Suppose that an AP sends multiple multicast data frames via an A-MPDU. After sending the A-MPDU, nonleaders that cannot decode one or more MPDUs in the A-MPDU will send NAK to the AP. When the AP receives a corrupted frame due to nonleaders' NAK transmissions, it cannot determine which MPDUs were lost at the receivers. Therefore, in such case, the AP should retransmit all MPDUs within the A-MPDU again even though only one MPDU was lost. This inefficient retransmission seriously decreases the network throughput especially when an A-MPDU contains many MPDUs. In REMP, we propose an advanced ARQ mechanism to overcome this inefficient retransmission problem. According to the new ARQ mechanism, the AP can determine which MPDUs need to be retransmitted with a limited control frame overhead.

2.3 Rate-Adaptive Multicast (RAM)

Existing rate adaptation mechanisms for unicast transmissions [3], [4], [5] cannot be directly applied to multicast scenarios due to the absence of MAC-level feedbacks (CTS or ACK) in multicasting. In order to make the AP obtain some feedbacks from multicast receivers, supplementary MAC-level signaling methods are needed. In [9], Choi et al. proposed Leader-based Multicast with the Auto Rate Fallback (LM-ARF) protocol that combines the NAK-based ARQ mechanism of LBP with the rate adaptation mechanism of ARF. In particular, the transmission rate increases to a higher rate after consecutive successful transmissions (i.e., reception of ACKs) and falls back to a lower rate after consecutive transmission failures (i.e., reception of NAKs or corrupted frames).

Rate-Adaptive Multicast [10] and Auto-Rate Selection mechanism for Multicast (ARSM) [11] adopt the closed-loop approach, i.e., the AP determines the transmission rate based on the explicit channel feedback from the receivers. In RAM and ARSM, a transmission rate for a multicast group is decided according to the signal-to-noise ratio (SNR) of the receiver that has the worst channel condition. Since RAM and ARSM utilize explicit feedbacks from receivers, they can adapt the transmission rate more accurately than LM-ARF.

In the closed-loop approach, the efficiency of rate adaptation heavily depends on the accuracy of the analytical model used for throughput estimation. However, existing models in RAM and ARSM, which were designed for the 802.11b WLANs, cannot accurately estimate multicast throughput in 802.11n WLANs when the A-MPDU aggregation is applied to multicast frames. In REMP, we propose an efficient rate adaptation mechanism based on the new throughput model for 802.11n multicast.

2.4 Utilizing Scalable Video Coding

The main idea of rate-adaptive multicast is to use the highest possible transmission rate for multicast transmissions while maintaining a certain level of reliability for all multicast receivers. Even though such mechanism is a good approach for exploiting multirate capability of WLANs, it may penalize stations exhibiting good channel conditions.

One good approach for solving the problem is to utilize the scalable video coding technique that supports flexible adaptation of video quality based on a hierarchical layer structure. With the scalable video coding, a single video sequence is encoded in a base layer and several enhancement layers. The base layer provides the information required for displaying the minimum quality video while the enhancement layers are used to increase the quality of the video. An enhancement layer is useful only if all the other corresponding lower layers are available at the user. The scalable extension of H.264/AVC standard, also known as H.264/SVC, is the standard defined by the Joint Video Team (JVT) of ISO/IEC and ITU-T [12].

The scalable video coding technique can be utilized in multicast as follows: The packets containing the base layer are transmitted with a relatively low rate, while those containing enhancement layers are transmitted with higher rates. In this way, it is ensured that all users of a multicast group get at least the minimum video quality while users with good channels get improved video quality. Given this approach of combining the scalable video coding with the multicast protocol, a challenging problem is to decide a proper transmission rate for each layer. Deb et al. [13] have tried to solve the problem in IEEE 802.16 WiMAX networks that have a very different MAC mechanism from WLANs. In [11], Villalón et al. proposed hierarchical ARSM, referred to as H-ARSM, which applies the scalable video coding to multicasting in WLANs. In H-ARSM, the AP first selects the rate corresponds to the base layer by taking account of a station with the lowest channel quality, and then applies one step higher rate to the enhancement layer.

However, H-ARSM cannot be directly applied to IEEE 802.11n WLANs where A-MPDU aggregation is enabled. Note that all MPDUs in an A-MPDU become a PSDU preceded by a PHY header as shown in Fig. 1a, and a MSC used for the PSDU is indicated in the PHY header. In other words, all aggregated MPDUs in the A-MPDU will be transmitted with the same PHY transmission rate. Therefore, if MPDUs that contain different layers are aggregated in an A-MPDU, the AP cannot differentiate the transmission rate for each layer as in H-ARSM.

3 Motivation

The IEEE 802.11n standard does not introduce additional features for multicast services, that is, it inherits the limitations on multicast transmissions from previous standards. Clearly, a simple and straightforward solution for multicasting in IEEE 802.11n WLANs is to adopt existing multicast protocols, which are based on the legacy 802.11 MAC. However, as we described in the previous section, applying existing multicast protocols to IEEE 802.11n WLANs raises some problems when the A-MPDU aggregation is used for multicast data frames.

To the best of our knowledge, the Double Piggyback Mode Multicast Protocol (DPMM) [15] is the only multicast protocol designed for the IEEE 802.11n MAC. In DPMM, the AP first sends Multicast RTS (MRTS) to multicast receivers, and three selected receivers, referred to as cluster heads, sequentially reply with Multicast CTS (MCTS). By measuring the signal strength of the MCTS from cluster heads, the AP estimates the channel condition for the multicast group. Then, the AP chooses an MCS according to the predefined packet delivery ratio threshold, and sends an A-MPDU frame with the selected transmission rate. At the next transmission stage, upon receiving MRTS, each cluster head piggybacks the BA information on the MCTS. By checking MCTS, the AP can know the sequence numbers of lost MPDUs at the previous transmission stage.

In DPMM, if a certain cluster head successfully receives 10 data frames consecutively, then it is removed from a cluster heads set and the AP randomly selects a new receiver as a cluster head. Due to the randomness of cluster head selection, the AP may overestimate the channel condition of multicast receivers. Moreover, since only three selected cluster heads are allowed to send MCTS, the AP cannot know all receivers' BA information. As a result, the reliability of data transmissions is not guaranteed to noncluster heads receivers.

In this paper, we propose a new multicast protocol for IEEE 802.11n WLANs, called REMP. REMP introduces an enhanced ARQ mechanism and efficient MCS selection mechanism for multicast transmissions with A-MPDU aggregation. The main idea of REMP is to dynamically adjust the number of control frame transmissions per data transmission. When the channel condition is stable, an AP exchanges control frames with a selected multicast receiver for each A-MPDU transmission as in the NAK-based ARQ mechanism. However, when the channel condition is dynamic, the AP exchanges control frames with all the multicast receivers just as the individual ARQ mechanism. Based on feedbacks in the control frames sent by multicast receivers, the AP selectively retransmits erroneous multicast frames and efficiently adjusts MCS under varying channel conditions. In addition, for providing efficient scalable video streaming over IEEE 802.11n WLANs, we propose an extension of REMP, named S-REMP. In S-REMP, different layers of scalable video are transmitted with different MCSs by splitting an A-MPDU frame into two sub A-MPDU frames.

4 Reliable and Efficient Multicast Protocol

In this section, we describe operations of REMP under a single basic service set (BSS) scenario which is composed of an 802.11n AP and multiple 802.11n stations. In REMP, there is no interaction among adjacent APs (e.g., time synchronization or exchange of control frames). If multiple APs coexist in the same channel, they share the wireless medium according to CSMA/CA with exponential backoff just like in the distributed coordination function (DCF) mode of the legacy 802.11 MAC.

4.1 REMP Operation

REMP basically adopts the leader-based approach, that is, an AP maintains a list of receivers for each multicast group[2] and selects one leader per multicast group. For a multicast group management, the AP maintains GroupTable that has six fields. Group address field and Leader address (LA) field denote the MAC addresses of the multicast group and leader, respectively. Address list field contains addresses of multicast receivers, and SNR list field contains the corresponding SNR values of multicast receivers. Timestamp field stores the time of the last transmission for the multicast group. T_{delay} field stores the recent waiting time for channel access. The initial values of Timestamp field and T_{delay} field are zero and $CW_{min}/2 \times tSlotTime$, respectively, where CW_{min} is the size of minimum congestion window size and *tSlotTime* is the time length of one slot time.

To select a leader for a multicast group, AP periodically performs a leader selection procedure by using Multicast Feedback Request (MFR) and Multicast Channel Acknowledgment (MCA) frames whose frame formats are shown in Figs. 2a and 2b, respectively. The leader selection procedure is initiated when an AP tries to transmit the first data frames for a multicast group. In this case, there is no leader for the multicast group (i.e., the Leader address field of the multicast group is empty). First, the AP sends MFR to the multicast group receivers. If there are R multicast group receivers, then the duration field of MFR is set to $R \times T_{MCA} + (R+1) \times tSIFS$, where *tSIFS* is the time length of SIFS, and T_{MCA} is the transmission time of a MCA. Receiver Address (RA) and Transmitter Address (TA) fields represent the MAC address of the multicast group and the AP, respectively, Type field is set to zero, and the following two fields represent the number of receivers in the multicast group and the addresses of receivers, respectively.

Upon receiving MFR, stations that do not join the multicast group set the network allocation vector (NAV)

2. As in the existing multicast protocols for WLANs, the receivers may send explicit link-level join/leave messages, or the AP may perform "IGMP snooping."

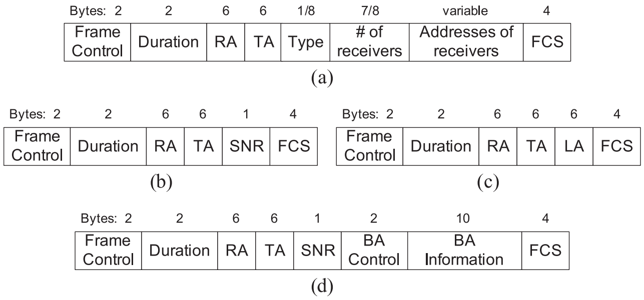

Fig. 2. Control frame formats in REMP. (a) MFR. (b) MCA. (c) MTA. (d) MBA.

value by the duration field of MFR. On the other hand, the multicast receivers in the group report the current SNR value to the AP by sending MCAs. To avoid collisions, MCAs are transmitted one by one according to the order in MFR. In particular, ith receiver waits for a time that equals to $(i-1) \times T_{MCA} + i \times tSIFS$ before sending MCA. Based on the received MCAs, the AP selects a receiver that has the worst SNR as the leader. After that, the AP updates the Leader address and SNR list fields in GroupTable and starts the leader selection timer for the multicast group. When the leader selection timer expires, the AP performs the leader selection procedure again for the multicast group and then restarts the timer. Note that this leader selection procedure is independently performed for each multicast group maintained by the AP.

Now we describe the data transmission procedure of REMP when the AP has an A-MPDU frame destined to a multicast group G. After a channel contention, the AP first sends a Multicast Transmission Announcement (MTA) whose frame format is shown in Fig. 2c. In MTA, RA, TA, and leader Address fields represent the MAC addresses of G, AP, and leader of G, respectively. Therefore, receivers receiving the MTA frame can learn that G will be the target of the following A-MPDU transmission and who is the leader of G. When the AP starts sending MTA, if the Timestamp value of G in the table is not zero, the difference between the current time and the stored Timestamp value becomes $T^{new}_{delay,G}$ and the T_{delay} field of G ($T_{delay,G}$) is updated by

$$T_{delay,G} = (1-\alpha) \cdot T^{old}_{delay,G} + \alpha \cdot T^{new}_{delay,G}, \quad (1)$$

where $0 \leq \alpha \leq 1$ and $T^{old}_{delay,G}$ is the previous value of $T_{delay,G}$ stored in GroupTable. After sending MTA, the AP waits for a reduced interframe space (RIFS) and then sends an A-MPDU that contains multiple multicast data frames with an appropriate MCS for the current channel conditions of receivers. (A detailed algorithm for the MCS selection will be introduced in Section 4.3.)

After a SIFS receiving the A-MPDU, the leader of G replies with a Multicast BA (MBA) frame to notify the current SNR as well as to acknowledge the received A-MPDU to the AP. (MBA has the same format as the legacy BA of 802.11n except for the additional SNR field as shown in Fig. 2d.) Other receivers of G, referred to as nonleaders, do nothing if all MPDUs in the A-MPDU are successfully received. Otherwise, to interrupt the MBA transmission, they send NAK that is a dummy frame that has the same frame length as MBA. Note that a receiver of G that successfully received MTA before receiving the A-MPDU can easily determine whether it is the leader of the multicast group or not. However, if a receiver of G that failed to receive MTA due to some problems (e.g., collisions) receives one or more MPDUs in the A-MPDU, then it cannot determine who the leader of G is. In such case, the receiver works as nonleader.

Following operations are decided according to a type of received control frame at the AP, and they are classified into three cases. Fig. 3 shows examples of three different cases where three stations belong to G and station A is the leader of G. The first case is that the AP does not receive any frame from the receivers as shown in Fig. 3a. Since the leader always sends MBA after the A-MPDU transmission, this case occurs when either the leader notification was failed (i.e., the leader did not receive MTA) or MBA sent by the leader was lost at the AP due to the bad channel condition. In this case, since the AP cannot receive any feedback, it transmits MTA and the A-MPDU again after channel contention.

The second case is that the AP receives MBA from the leader as shown in Fig. 3b. Since receiving MBA means that in case nonleaders did not send NAKs, the AP can expect that all nonleaders successfully received all MPDUs in the A-MPDU. In this case, the AP updates the SNR value of the leader and stores the current time at the Timestamp field of G in GroupTable. Meanwhile, it deletes successfully delivered MPDUs from its transmission queue according to the BA bitmap field of MBA. MPDUs remaining in the transmission queue are retransmitted at the next transmission stage.

The third case is that AP receives NAK or a corrupted frame. This case occurs when one or more nonleaders send NAKs. For example, in Fig. 3c, the AP receives a corrupted frame after the A-MPDU transmission due to the NAK transmission from station B. In this case, the AP cannot know which MPDUs need to be retransmitted. A simple retransmission policy for this case is to send the transmitted A-MPDU again just like the first case. However, as we

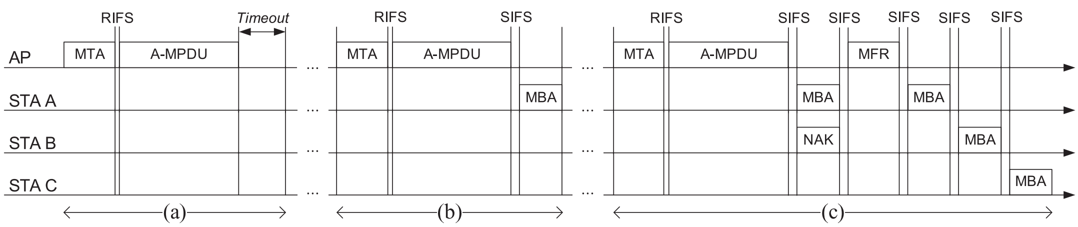

Fig. 3. Timing diagram in REMP where station A is a leader. (a) AP does not receive any frame. (b) AP receives MBA from the leader. (c) AP receives a corrupted frame and performs the leader change procedure.

discussed in Section 2.2, this results in inefficient retransmissions. In order to eliminate unnecessary MPDU retransmissions, the AP should receive individual acknowledgment from each multicast receiver. Beside, receiving NAK or a corrupted frame implies that channel conditions of nonleaders have changed. To select appropriate MCS for the next transmission stage, the SNR values of G in GroupTable need to be updated directly without waiting for the next leader selection procedure.

For efficient retransmissions and MCS selection, a leader change procedure is performed in the third case. The frame exchange sequence of the leader change procedure is the same as that of the leader selection procedure, except that multicast receivers send MBAs rather than MCAs. As shown in Fig. 3c, after a SIFS receiving NAK or a corrupted frame, the AP starts the leader change procedure for G by sending MFR with setting Type field to one. (Note that in the leader selection procedure, Type field is set to zero.) Upon receiving MFR whose Type field is one, each multicast receiver of G determines the current SNR, and sequentially replies with MBA for the transmitted A-MPDU. After receiving MBA from the last receiver, the AP deletes MPDUs that all receivers successfully received. In addition, the AP selects a new leader of G, updates Leader address, SNR list, and Timestamp fields for G in GroupTable, and restarts the leader selection timer for G.

One problem of the ARQ mechanism of REMP is a control frame overhead under highly dynamic channel conditions. If the station with the worst SNR keeps changing, the leader change procedure can be performed too often, which may cause many control frame transmissions. In order to cope with this problem, we define an optional system parameter r, the maximum number of receivers that participate in the leader change procedure. Before initiating the leader change procedure, if r is smaller than the total number of receivers in a target multicast group, the AP selects r receivers that have bad SNR based on GroupTable. Then, the AP inserts addresses of the selected receivers to MFR. Upon receiving MFR, only selected receivers send MBA to the AP. The value of r may be adjusted according to the channel condition or the number of receivers. In this paper, however, we only consider a case where r is set to infinite (i.e., all receivers always participated in the leader change procedure) due to space limitations.

4.2 Throughput Analysis of REMP

In this section, we derive a throughput model for REMP to develop an efficient MCS selection algorithm. We make the following assumptions to simplify the analysis;

1. AP generates multicast data frame continuously and its queue is never empty,
2. control frames are transmitted by the basic MCS (MCS 0) and they are always delivered without error,
3. PLCP header of each frame is also delivered without error, and
4. the air propagation delay is negligible.

The expected throughput of multicast group G is defined as the ratio of the expected delivered data payload to the expected transmission time that is composed of the waiting time for channel access for G ($T_{delay,G}$) and transmission time for control and data frames (T_{frame}). $T_{delay,G}$ is maintained in the AP's GroupTable as illustrated above, and T_{frame} is calculated based on MCS mode, SNR of receivers, and the size of MPDUs as follows.

Let $bps(m)$ the bit-per-second for MCS index m. Taking into account the frame format shown in Fig. 1a, the transmission time when a single subframe (composed of delimiter and following MPDU) in A-MPDU is transmitted by MCS index m can be expressed by

$$T_{subframe}(l,m) = \frac{(lDelimiter + lMACOverhead + l)\cdot 8}{bps(m)}, \quad (2)$$

where $lDelimiter$, $lMACOverhead$, and l are the lengths of the MPDU delimiter, the overhead of MPDU (i.e., the MAC header and FCS), and the payload length, respectively. For a given A-MPDU A, let $N(A)$ represent the number of MPDUs and l_A^i represent the payload length of ith MPDU in A. Then, the transmission time for the A-MPDU A becomes

$$\begin{aligned} T_{AM}(A,m) = {} & tPLCPPreamble + tPLCPHeader \\ & + \sum_{i=1}^{N(A)} [T_{subframe}(l_A^i, m)], \end{aligned} \quad (3)$$

where $tPLCPPreamble$ and $tPLCPHeader$ are the transmission times for PLCP preamble and header, respectively.

To take into account the effect of the leader change procedure, we need to know the probability that there is at least one NAK transmission after an A-MPDU transmission. Let $P_{err}^m(h,s)$ be the packet error probability for an h-octet long frame to be transmitted using MCS mode m under SNR value s. Since an MPDU is extracted when the preceding MPDU delimiter has no error, the probability of transmission failure for an MPDU whose payload length is l can be obtained by

$$\begin{aligned} P_{err}^{MPDU}(l,m,s) = 1 - & \left[1 - P_{err}^m(lDelimiter, s)\right] \\ & \cdot \left[1 - P_{err}^m(lMACOverhead + l, s)\right]. \end{aligned} \quad (4)$$

Let the channel condition vector $\hat{s}_G = \{s_G^1, \ldots, s_G^{N(G)}\}$ denote the SNR value of each receiver where $N(G)$ is the number of receivers in G, and s_G^1 represent the SNR value of leader of G. Since a nonleader sends NAK when it fails to receive one or more MDPUs within the A-MPDU, the probability that a nonleader whose SNR is s sends NAK is $\prod_{i=1}^{N(A)}[1 - P_{err}^{MPDU}(l_A^i, m, s)]$. Thus, the probability that there will be at least one NAK transmission becomes

$$P_{NAK}(A,m,G) = 1 - \prod_{j=2}^{N(G)} \left[\prod_{i=1}^{N(A)} \left[1 - P_{err}^{MPDU}\left(l_A^i, m, s_G^j\right)\right]\right]. \quad (5)$$

As a result, T_{frame} can be expressed by

$$\begin{aligned} T_{frame}(A,m,G) = {} & T_{MTA} + tRIFS + T_{AM}(A,m) + tSIFS \\ & + T_{MBA} + P_{NAK}(A,m,G)\cdot(tSIFS \\ & + T_{MFR}^{N(G)} + N(G)\cdot(tSIFS + T_{MBA})), \end{aligned} \quad (6)$$

where $T_{MFR}^{N(G)}$ denotes the transmission time of MFR where there are $N(G)$ multicast receivers and T_{MBA} denotes the transmission time of MBA.

Now we compute the average length of successfully delivered data to multicast receivers. We can say that a MPDU is successfully delivered to all receivers if there is no retransmission request from any receivers. Let $P_{retx}(l, m, s)$ be the probability that a receiver having channel condition s requests a retransmission for a MPDU whose payload length is l. Since the AP sends MPDUs until all receivers successfully receives it, some receivers may receive duplicated MPDUs. For duplicate MPDUs, retransmissions are not needed. Therefore, $P_{retx}(l, m, s)$ becomes

$$P_{retx}(l, m, s) = \begin{cases} 0, & \textit{if duplicated}, \\ P_{err}^{MPDU}(l, m, s), & \textit{otherwise}. \end{cases} \tag{7}$$

Note that the AP knows whether MPDU is duplicated for each receiver or not based on the previously received MBAs. So the average length of successfully delivered data can be calculated by

$$D(A, m, G) = \sum_{i=1}^{N(A)} \left[l_A^i \cdot \prod_{j=1}^{N(G)} \left(1 - P_{retx}\left(l_A^i, m, s_G^j\right)\right) \right]. \tag{8}$$

Finally, the expected throughput is derived as

$$TP(A, m, G) = \frac{D(A, m, G)}{T_{delay,G} + T_{frame}(A, m, G)}. \tag{9}$$

4.3 MCS Selection Algorithm in REMP

The proposed MCS selection algorithm in REMP aims at finding an MCS that maximizes the average multicast throughput while keeping the MPDU error probability at the leader for a given threshold (P_{target}). The algorithm is executed when AP starts to transmit MTA, and the result is applied to the following A-MPDU transmission.

Algorithm 1 shows MCS selection algorithm written in pseudocode. For a given A-MPDU A, AP first finds l_{max}, the length of the longest MPDU in A. Then, m_{Alg1}^{max} is set to the highest m that satisfies the following condition:

$$P_{err}^{MPDU}\left(l_{max}, m, s_G^1\right) \le P_{target}. \tag{10}$$

The reason why we choose l_{max} is that a longer MPDU has a higher error probability than a shorter MPDU. For the **while** loop (steps 8-13), MCS index m is increased until m_{Alg1}^{max}. In each iteration, we check the value of $TP(A, m, G)$, which is the estimated average throughput with the current value of m. If it is the maximum throughput until now, then m_{Alg1} is changed to the current value of m. At the end of the **while** loop, m_{Alg1} is returned and then used as the selected MCS for A-MPDU A.

Algorithm 1. MCS selection in REMP
1: **Input**
2: A: A-MPDU to be transmitted
3: G: target multicast group
4: **begin procedure**
5: Initialize $m \leftarrow 0$, $m_{Alg1} \leftarrow 0$, $TP_{max} \leftarrow 0$
6: $l_{max} \leftarrow \max\left\{l_A^1, \ldots, l_A^{N(A)}\right\}$
7: $m_{Alg1}^{max} \leftarrow$ the largest m that satisfies (10)
8: **while** $(m \le m_{max}^{Alg1})$ **do**
9: **if** $TP(A, m, G) > TP_{max}$ **then**
10: $TP_{max} \leftarrow TP(A, m, G)$, $m_{Alg1} \leftarrow m$
11: **end if**
12: $m \leftarrow m + 1$
13: **end while**
14: return m_{Alg1}
15: **end procedure**

The computational complexity of (9) depends on $N(A)$ and $N(G)$, and AP calculates (9) $m_{Alg1}^{max} + 1$ times during the **while** loop (steps 8-13). Note that the values of $N(A)$ and m_{Alg1}^{max} are limited values according to the IEEE 802.11n specification, and $N(G)$ is also limited in common WLAN environments. Therefore, it is possible for AP to complete the execution of Algorithm 1 in real time.

5 Scalable REMP

In this section, we propose S-REMP, an extension of REMP for utilizing the scalable video coding in multicasting over 802.11n WLANs. S-REMP inherits basic features of REMP such as the leader selection/change procedures and GroupTable management. In this section, we only describe additional features of S-REMP.

5.1 S-REMP Operation

S-REMP aims at providing high-quality video to receivers whose channel condition is good when the available bandwidth of AP is not sufficient to transmit the entire video frames to all multicast receivers. The main idea of S-REMP is to split a given A-MPDU to be transmitted into two sub A-MPDUs when the network load becomes heavy. Base layer video frames are always included in the first sub A-MPDU, and enhancement layer video frames are included in either the first or the second sub A-MPDU depending on the network load. The first sub A-MPDU is transmitted with a low MCS to guarantee a reliable transmission to all multicast receivers, while the second sub A-MPDU is transmitted with a higher MCS to avoid packet drops by reducing the transmission time.

In S-REMP, we define five additional fields in GroupTable. *Drop* field represents whether one or more data frames have dropped or not since the last data transmission for the corresponding multicast group. In particular, *Drop* field of a multicast group is reset to 0 for each data transmission for the multicast group, and set to 1 when a data frame destined to the multicast group is dropped due to overloading of the AP's queue. *LoadUp* and *LoadDown* fields represent the network load, and they are updated according to the status of Drop field before data transmissions. K field denotes the number of enhancement layers included in the second sub A-MPDU and m_{Alg2} field represents a MCS index for the second sub A-MPDU. Initial values of these new fields are 0.

Fig. 4 shows the flow chart of AP in S-REMP where AP has an A-MPDU frame destined to a multicast group G. When AP starts to send MTA, it first runs Algorithm 1 just like in REMP, and then checks $Drop_G$, which is the *Drop* field of multicast group G in GroupTable. If $Drop_G$ is 1, it means that at least one multicast data frame destined to G has been dropped after the previous data transmission. In this case, AP increases $LoadUp_G$ field by one and resets $LoadDown_G$ and $Drop_G$ to 0. Otherwise, if $Drop_G$ is 0, AP increases $LoadDown_G$ field by one and resets $LoadUp_G$ and $Drop_G$

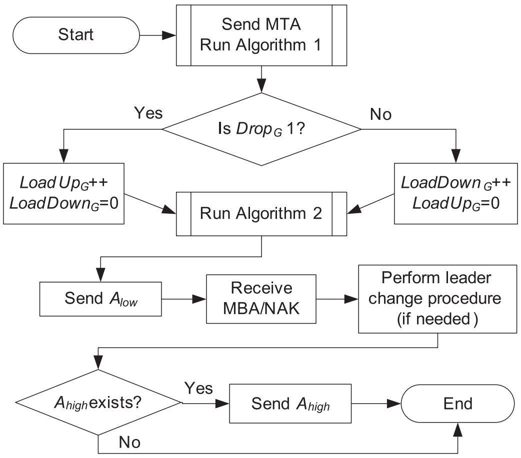

Fig. 4. Flow chart of AP in S-REMP.

fields to 0. After that, AP runs Algorithm 2, which updates K_G and $m_{Alg2,G}$. In Algorithm 2, the values of K_G and $m_{Alg2,G}$ are decided according to $LoadUp_G$ and $LoadDown_G$ fields. (The detailed description of Algorithm 2 will be provided in the next section.)

Algorithm 2. A-MPDU split and MCS selection

Input
  A: A-MPDU to be transmitted
  G: target multicast group
  m_{Alg1}: result of Algorithm 1
begin procedure
$l_{max} \leftarrow \max\{l_A^1, \ldots, l_A^{N(A)}\}, s_{max} \leftarrow \max\{s_G^1, \ldots, s_G^{N(G)}\}$
$m_{max}^{Alg2} \leftarrow$ the largest m that satisfies (12)
if $m_{Alg2,G} \leq m_{Alg1}$ **then** $K \leftarrow 0, m_{Alg2,G} \leftarrow m_{Alg1}$ **end if**
if $LoadUp_G == ThreshUp$ **then**
  $LoadUp_G \leftarrow 0$
  if $m_{Alg2,G} \neq m_{Alg2}^{max}$ **then** $m_{Alg2,G} \leftarrow m_{Alg2,G} + 1$ **end if**
  elseif $K_G \neq N_e$ **then**
    Calculate T_{cur} by (13), $K_G \leftarrow K_G + 1$
    $m_{Alg2,G} \leftarrow$ the smallest m that satisfies (11) and (14)
  end elseif
end if
elseif $LoadDown_G == ThreshDown$ **then**
  $LoadDown_G \leftarrow 0$
  if $m_{Alg2,G} \neq m_{Alg1}+1$ **then** $m_{Alg2,G} \leftarrow m_{Alg2,G}-1$ **end if**
  elseif $K_G > 1$ **then**
    Calculate T_{cur} by (13), $K_G \leftarrow K_G - 1$
    $m_{Alg2,G} \leftarrow$ the largest m that satisfies (11) and (15)
  end elseif
  elseif $K_G == 1$ **then** $K_G \leftarrow 0$, $m_{Alg2,G} \leftarrow m_{Alg1}$ **end elseif**
end elseif
return K_G and $m_{Alg2,G}$
end procedure

After running Algorithm 2, AP splits the given A-MPDU A into two sub A-MPDUs, A_{low} and A_{high}, before a data transmission. Let N_e denote the number of enhancement layers for a given scalable video stream. A_{low} includes MPDUs containing base layer video frames and low $N_e - K_G$ enhancement layer $(\text{first}, \ldots, (N_e - K_G)\text{th}$ enhancement

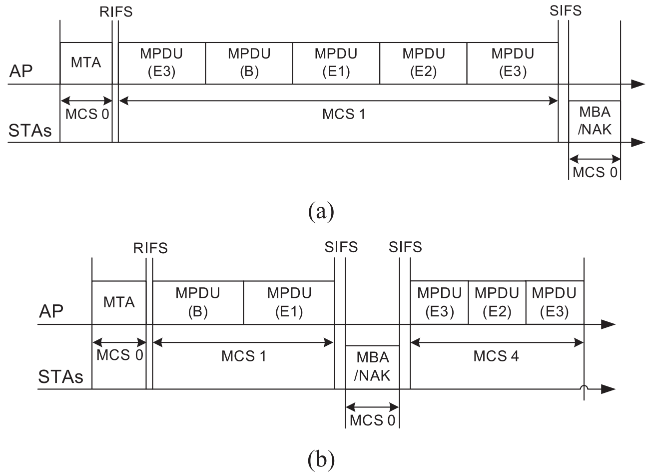

Fig. 5. Timing diagram in S-REMP where $N_e = 3$ and $m_{Alg1} = 1$. (a) $K = 0$. (b) $K = 2$ and $m_{Alg2,G} = 4$.

layer) video frames, and A_{high} included the other MPDUs. If K_G is 0, A_{low} becomes the same as A, and A_{high} does not be constructed. After RIFS from the end of the MTA transmission, AP transmits A_{low} with m_{Alg1} and the multicast receivers in G reply with MBA or NAK. If AP receives NAK or a corrupted frame, then it performs the leader change procedure just like in REMP. Erroneous MPDUs in A_{low} will be retransmitted at the next transmission.

If A_{high} does not exist (i.e., K_G is 0), the transmission procedure ends here. Note that in this case, a frame exchange sequence becomes the same as that of REMP. Otherwise, with a gap of SIFS, AP transmits A_{high} with $m_{Alg2,G}$. Here, the transmission of A_{high} is not followed by MBA or NAK and is regarded as the legacy multicast transmission in IEEE 802.11n. In contrast to the case of A_{low}, MPDUs in A_{high} are deleted from AP's queue after the transmission of A_{high} without waiting for retransmission.

Fig. 5 shows examples of data transmissions in S-REMP with varying the value of K_G, where N_e is 3. In the figure, B represents an MPDU including the base layer video frame and Ei represents an MPDU including the ith enhancement layer video frame. The leader change procedure is not performed in these examples. In Fig. 5a, K_G is 0 and m_{Alg1} is 1. As mentioned above, when K_G is 0, the frame exchange sequence becomes the same as that of REMP since there is no transmission of A_{high}. In Fig. 5b, K_G is 2 and m_{Alg1} and $m_{Alg2,G}$ are 1 and 4, respectively. A_{high} includes the second and third enhancement layer video frames and is transmitted with MCS 4. As shown in the figure, the total transmission time is reduced compared to Fig. 5a. Although some multicast receivers having bad channel condition may not successfully receive the second and third enhancement layer video frames, the packet drop problem at the AP's queue can be alleviated with the help of reduced transmission time.

5.2 A-MPDU Split and MCS Selection Algorithm in S-REMP

Now we provide a detailed description of Algorithm 2. The value of $m_{Alg2,G}$ has the following constraint:

$$m_{Alg1} \leq m_{Alg2,G} \leq m_{Alg2}^{max}, \tag{11}$$

where m_{Alg1} is the result of Algorithm 1, and m_{Alg2}^{max} is obtained as follows (steps 6-7). AP first finds the length of the longest MPDU in A (l_{max}) and the highest SNR in vector $\hat{s}_G$ (s_{max}). Then, AP sets m_{Alg2}^{max} to the highest m that satisfies the following condition:

$$P_{err}^{MPDU}(l_{max}, m, s_{max}) \leq P_{target}. \tag{12}$$

After that, AP compares the current $m_{Alg2,G}$ stored in GroupTable with m_{Alg1}. If $m_{Alg2,G}$ is smaller or equal to m_{Alg1}, then K_G is reset to 0 and $m_{Alg2,G}$ is changed to m_{Alg1} (step 8).

Next AP determines the current network load through $LoadUp_G$ and $LoadDown_G$ in GroupTable. If $LoadUp_G$ is equals to a predefined threshold *ThreshUp*, then AP determines the current network load is heavy. In this case, in order to decrease the transmission time, AP adjusts K_G or $m_{Alg2,G}$ as follows (steps 9-16). If $m_{Alg2,G}$ is not m_{Alg2}^{max}, then AP increase $m_{Alg2,G}$ by one and the algorithm ends. This results in that A_{high} is transmitted with a higher MCS compared to the previous transmission. If $m_{Alg2,G}$ is equal to m_{Alg2}^{max} and K_G is not N_e, more video frames have to be included in A_{high}. For this, AP calculates T_{cur}, the expected data transmission time with the current parameters, according to

$$T_{cur} = \begin{cases} T_{AM}(A_{low}, m_{Alg1}), & \text{if } K \text{ is } 0, \\ T_{AM}(A_{low}, m_{Alg1}) + tSIFS & \\ \quad + T_{AM}(A_{high}, m_{Alg2,G}), & \text{otherwise.} \end{cases} \tag{13}$$

Then, AP increases K_G by one (it results in that A_{high} contains one more enhancement layer), and sets m_{Alg2} to the smallest m that satisfies (11) and the following condition:

$$T_{AM}(A_{low}, m_{Alg1}) + tSIFS + T_{AM}(A_{high}, m) < T_{cur}. \tag{14}$$

According to the above condition, we can expect that the transmission time decreases.

If $LoadDown_G$ is equals to a predefined threshold $ThreshDown_G$, then AP tries to decrease K_G or $m_{Alg2,G}$ in order to increase the transmission time (steps 17-25). If $m_{Alg2,G}$ is not $m_{Alg1} + 1$, then AP decrease $m_{Alg2,G}$ by one and the algorithm ends. This results in that A_{high} is transmitted with a lower MCS compared to the previous transmission. Otherwise, AP checks whether K_G is 1 or not. If K_G is larger than 1, AP calculates T_{cur} by (13). Then, AP decreases K_G by one and sets $m_{Alg2,G}$ to the biggest m that satisfies (11) and the following condition:

$$T_{AM}(A_{low}, m_{Alg1}) + tSIFS + T_{AM}(A_{high}, m) > T_{cur}. \tag{15}$$

If K_G is 1, then AP resets K_G to 0 and $m_{Alg2,G}$ to m_{Alg1}.

The computational complexity of Algorithm 2 depends on $N(A)$ and the difference between m_{Alg1}^{max} and m_{Alg2}^{max}. Compared to Algorithm 1, AP can complete the execution of Algorithm 2 in shorter time since the complexity of Algorithm 2 does not depend on $N(G)$.

6 Performance Evaluation

This section is composed of two sections. In Section 6.1, we evaluate the performance of REMP and compare it with DPMM and the legacy 802.11n multicast with normal multicast traffic (i.e., scalable video coding is not applied).

TABLE 2
Data Rates of Mandatory MCSs in IEEE 802.11n

MCS index	0	1	2	3	4	5	6	7
Data rate (Mbps)	6.5	13	19.5	26	39	52	58.5	65

In Section 6.2, we evaluate the performance of S-REMP and compare it with REMP, DPMM, and the legacy 802.11n multicast when the scalable video coding is applied.

6.1 Performance Evaluation for Normal Multicast Traffic

We performed simulations using the ns-2 simulator for the performance comparison of REMP with DPMM and the legacy 802.11n multicast protocol. P_{target} and the time out value for leader selection timer of REMP are set to 0.9 and 5 seconds, respectively. (With these values, REMP shows good performance in most of simulation scenarios. The effect of them is not presented in this paper due to space limitations.) For simplicity, we only consider eight mandatory MCSs among 77 MCSs. Table 2 shows data rates of mandatory MCSs (index 0-7) for 20 MHz channel bandwidth, single spatial stream, and 800 ns guard interval. Note that, in the legacy 802.11n multicast protocol, there is no MCS selection procedure and thus multicast frames are always transmitted with MCS 0.

For the performance study, we evaluate the following metrics: *multicast throughput*, *unicast throughput*, *control overhead*, and *delay*. *Multicast throughput* is given by the average throughput received by multicast receivers from AP. *Unicast throughput* is the average throughput received by AP from each unicast sender. From this metric, we can estimate the bandwidth not used (i.e., available to the unicast senders) by each multicast protocol. *Control overhead* is defined as the ratio between the total bits of transmitted control frames and the total bits of transmitted multicast data frames. *Delay* is defined as the average elapsed time between the time when AP starts to transmit a data frame to through the wireless channel and the time when the frame is successfully received at a station. *Fairness index* is calculated by $(\sum_{i=1}^{N} x_i)^2 / (N \sum_{i=1}^{N} x_i^2)$ where N is the total number of multicast receivers and x_i is the achieved throughput of receiver i [22]. The fairness index lies between 0 and 1, and if it is close to 1, then better fairness is achieved.

In the first scenario, there exists an AP, 10 multicast receivers of the same multicast group, and five uplink senders. The multicast receivers move within a circle according to the random waypoint model, and the AP is located at the center of the circle. The radius of the circle varies from 50 to 200 meters. To observe results in dynamic channel conditions, we set the speed of nodes to 5 m/s and pause time to zero. The AP generates multicast traffic for the multicast group with constant bit rate 5 Mbps (the bitrate of MPEG-2 video for SDTV). Five uplink senders locate at 100 m from the AP, which transmit unicast frames from greedy sources as background traffic. For unicast transmissions, the uplink senders do not change MCS but always use MCS 3, a robust MCS at 100 m from the AP.[3] The

3. Since we use the two-ray ground reflection model provided in ns-2, the channel quality of a station is decided by the distance between the AP and station.

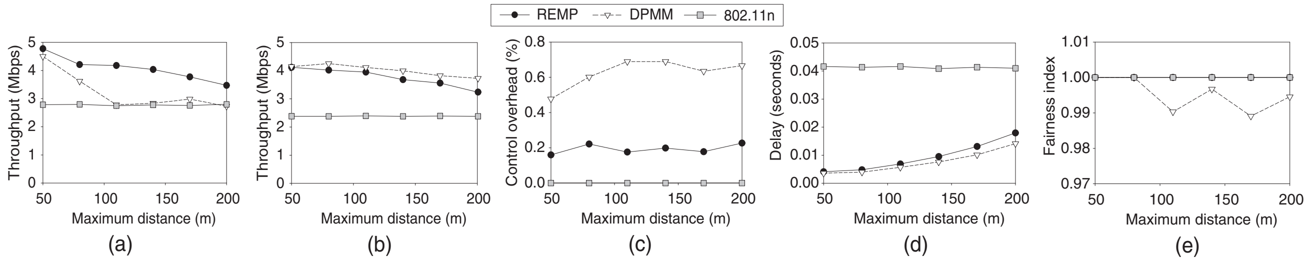

Fig. 6. Performance comparisons with varying the maximum distance between AP and multicast receivers where the number of multicast receiver is 10. (a) Multicast throughput. (b) Unicast throughput. (c) Control overhead. (d) Delay. (e) Fairness index.

packet size for both multicast and unicast traffic is fixed to 1,024 bytes.

Fig. 6 shows the simulation results of legacy 802.11n multicast, DPMM, and REMP as a function of maximum distance between the AP and multicast receivers. From Fig. 6a, we can see that the multicast throughput is not affected by the distance from the AP in the case of 802.11n multicast since the AP always uses MCS 0 even if the channel quality of multicast receivers is good. With the help of the MCS adjustment, DPMM achieves higher multicast throughput than 802.11n multicast when multicast receivers are close to the AP (i.e., the channel quality is good), while it shows very comparable performance to 802.11n multicast as the distance increases. However, DPMM shows much worse performance than REMP in all distance ranges. Since the AP in DPMM can receive feedbacks from only three receivers that are selected as cluster heads for each data transmission, it cannot perform an efficient MCS selection and retransmissions.

Fig. 6b shows the unicast throughput. REMP and DPMM outperform 802.11n multicast since they make the AP use shorter time for multicast transmissions by selecting higher MCS. Comparing REMP and DPMM, DPMM shows a little bit higher uplink throughput. This is because REMP selects more robust MCS compared to DPMM that selects MCS based on the channel quality of only three receivers. Selecting robust MCS increases reliability, but it requires more transmission time for a multicast data frame. However, the throughput difference between REMP and DPMM in Fig. 6b is much smaller than the difference in Fig. 6a. That is, REMP shows much improved multicast performance in spite of a little performance penalty in uplink performance.

Fig. 6c shows the control overhead for multicast traffic. As we can see from the figure, REMP causes much less control overhead than DPMM. The reason is that three control frames are transmitted from multicast receivers for every A-MPDU transmission in DPMM, while only one control frame is transmitted from the leader if there is no MPDU loss at nonleaders in REMP.

Fig. 6d shows the delay for multicast traffic. With the help of MCS adjustment, REMP and DPMM show smaller delay than 802.11n multicast. REMP shows slightly higher delay than DPMM due to the difference of retransmission policy. In REMP, the AP retransmits multicast data frames for all multicast receivers, while it does for only three receivers in DPMM. The retransmission procedure of REMP increases reliability but requires longer delay compared to DPMM. However, the difference between REMP and DPMM is not so significant.

Fig. 6e shows the fairness index. REMP and 802.11n multicast show constant fairness index which is closed to 1, i.e., all multicast receivers achieve almost the same multicast throughput (their results are overlapped in the figure). In 802.11n multicast, since the AP always uses MCS 0, robust data delivery is guaranteed to all receivers regardless of receiver's channel condition. In REMP, with the help of enhanced ARQ mechanism, robust data delivery is also guaranteed while the AP adjusts MCS for multicast transmission. DPMM shows slightly lower fairness index than others. In DPMM, receivers under lower channel condition sometimes fail to receive multicast data due to its inefficient ARQ mechanism.

In the second scenario, we studied the performance when the number of multicast receivers varies from 5 to 30 and the maximum distance is fixed to 150 m. Other simulation settings are equal to the first scenario. Fig. 7 shows the simulation results. From the results, we can see that the performance of 802.11n multicast and DPMM is almost independent of the number of multicast receivers. In contrast, the performance of REMP degrades (i.e., the multicast throughput and unicast throughput decrease and the control overhead and delay increase) gradually as the increase in the number of receivers, except the fairness performance. In REMP, when there are many receivers in a

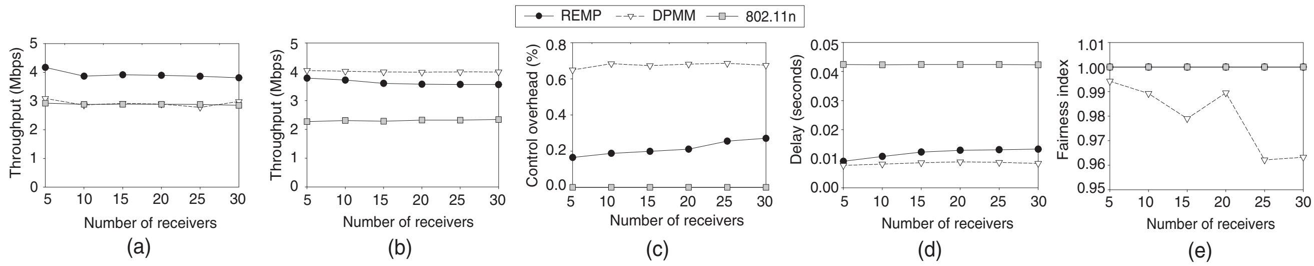

Fig. 7. Performance comparisons with varying the number of multicast receiver where the maximum distance between AP and multicast receivers is 150 m. (a) Multicast throughput. (b) Unicast throughput. (c) Control overhead. (d) Delay. (e) Fairness index.

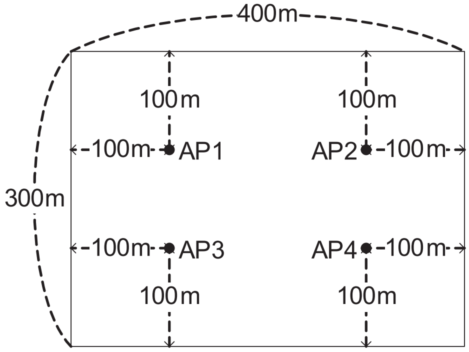

Fig. 8. Simulation environment for multi-AP scenario.

multicast group, the probability of collision between MBA and NAK becomes high and thus the leader change procedure is performed frequently. In addition, the number of transmitted control frames during the leader change procedures also increases as the increase in the number of receivers. However, the performance degradation according to the increase in the number of receivers is not so significant. In terms of fairness, REMP and 802.11n multicast show better performance than DPMM just like in the first scenario.

In the third scenario, we studied the performance when multiple APs exist. Fig. 8 shows simulation environments. A network size is $400\ \mathrm{m} \times 300\ \mathrm{m}$, and four APs that operate in the same channel are located as shown in the figure. Since distances between APs are smaller than the transmission range of AP (250 m), each AP can sense transmissions from other APs. During simulations, 30 fixed multicast receivers are randomly distributed and they associate with the closest AP. All the receivers belong to the same multicast group and each AP generates multicast traffic for the multicast group with constant bit rate of 5 Mbps. In order to focus on the effect of interference among APs, unicast uplink senders do not exist in this scenario.

Table 3 shows results in the multi-AP scenario. 802.11n multicast shows the worst performance in terms of multicast throughput and delay since the AP always uses MCS 0. REMP outperforms DPMM in all metrics. From the results, we can expect that REMP works efficiently with dense AP deployment.

6.2 Performance Evaluation for Multicast Traffic with H.264/SVC

In this section, we compare the performance of S-REMP with REMP, DPMM, and legacy 802.11n multicast when H.264/SVC is applied. For this, we use Joint Scalable Video Model (JSVM), the reference open source software for H.264/SVC coding/decoding released and maintained by the MPEG/ITU Joint Video Team [16]. The simulation process can be summarized in three steps. First, a YUV video is encoded in the H.264/SVC format by using the JSVM Encoder. Second, in the ns-2 simulator, an AP transmits the encoded video to multicast receivers through various multicast protocols. Each multicast receiver stores successfully received video data in a separated file. Third, based on the received video data, YUV video files for each receiver are generated through the JSVM decoder. A test video sequence is 4CIF-size YUV video (soccer) at 30 fps and it is repeated during 50 seconds. Through JSVM, we encode the 50 seconds-long video with three quality-layers, base-layer (BL) and two enhancement layers (EL1 and EL2). Bit rates of BL, EL1, and EL2 are 688.7, 829.2, and 1,358.1 kbps, respectively. GOP size is 16 and quantization parameters of BL, EL1, and EL2 are 38, 32, 28, respectively. In addition, we use the Scalable Video coding streaming Evaluation Framework (SVEF) [17] for error concealment of decoded video.

TABLE 3
Simulation Results in Multi-AP Scenario

Metric	802.11n	DPMM	REMP
Multicast throughput (Mbps)	1.64	4.58	4.99
Control overhead (%)	0	0.05	0.02
Delay (msec)	42	1.48	1.21
Fairness index	0.99	0.97	1

TABLE 4
Distance between AP and Each Multicast Receiver

Scenario	MR1	MR2	MR3	MR4	MR5
Scenario 1	50m	75m	100m	125m	150m
Scenario 2	50m	90m	130m	170m	210m

For ns-2 simulations, we set up an IEEE 802.11n WLAN that consists of one AP, five multicast receivers, and five unicast uplink senders. In order to see the effect of wireless channel quality on video quality. The five multicast receivers are referred to as MR1, MR2, MR3, MR4, and MR5, and the distance between the AP and each receiver is shown in Table 4. As shown in Table 4, two simulation scenarios have different distances between the AP and multicast receivers. In each scenario, the distance from all uplink senders to the AP is the same as the distance from the AP to MR3. MCSs for uplink transmissions in scenario 1 and scenario 2 are fixed to MCS 3 and MCS 2, respectively (due to the distance difference of 30 m between the two scenarios). Other simulation settings are the same as the simulation scenarios in Section 6.1. In S-REMP, $ThreshUp$ and $ThreshDown$ are set to 2 and 10, respectively.

Fig. 9 shows the delivery ratio of each receiver in scenario 1 with legacy 802.11n multicast, DPMM, REMP, and S-REMP. The delivery ratio of a multicast receiver is

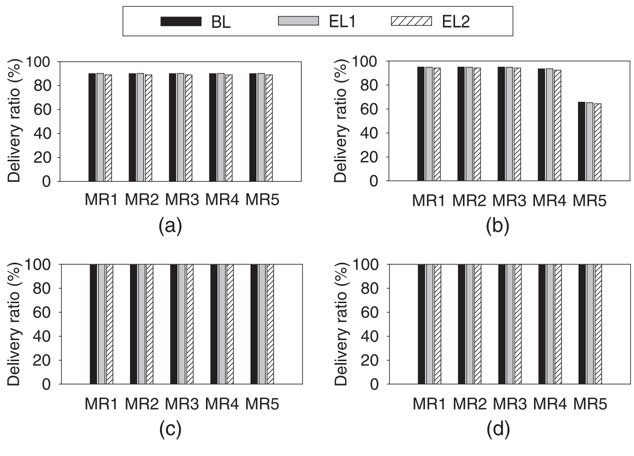

Fig. 9. Delivery ratio of each receiver in the scenario 1. (a) 802.11n. (b) DPMM. (c) REMP. (d) S-REMP.

TABLE 5
Average PSNR(dB) of Each Protocol in Scenario 1

Receiver ID	802.11n	DPMM	REMP	S-REMP
MR1	31.38	34.53	36.86	36.85
MR2	31.38	34.53	36.86	36.85
MR3	31.38	34.53	36.86	36.85
MR4	31.38	33.56	36.86	36.85
MR5	31.38	26.95	36.86	36.85

TABLE 6
Average PSNR(dB) of Each Protocol in Scenario 2

Receiver ID	802.11n	DPMM	REMP	S-REMP
MR1	25.72	24.70	28.99	36.84
MR2	25.72	24.70	28.99	34.81
MR3	25.72	24.70	28.99	33.76
MR4	25.72	24.49	28.99	33.51
MR5	25.72	23.51	28.99	33.35

defined as the ratio of the length of received video data by a receiver to the length of video data encoded by the JSVM encoder. As shown in Fig. 9a, with 802.11n multicast, all multicast receivers achieve about 90 percent delivery ratio for each layer. Fig. 9b shows that DPMM has a significant unfairness problem among receivers. MR5 that has the worst channel condition achieves much lower delivery ratio than the others. Fig. 9c shows the delivery ratio of REMP, where all multicast receivers achieve about 100 percent delivery ratio for each layer with the help of the efficient MCS selection and retransmission procedures. Note that in REMP, the AP always selects MR5 as the leader of the multicast group and thus it can select a robust MCS for the multicast group. From Fig. 9d, we can see that S-REMP also shows about 100 percent delivery ratio for each layer. In scenario 1, the AP does not split A-MPDUs before transmissions since the available bandwidth is high enough to transmit all layers.

Now we present a quantitative assessment of video quality perceived by a user using Peak Signal-to-Noise Ratio (PSNR). PSNR is commonly used as a quality measurement between the original and reconstructed video, where higher PSNR means better video quality. Table 5 shows the average PSNR of each video frame in scenario 1. As we can expect, the PSNR is closely related to the delivery ratio. DPMM shows better results than 802.11n multicast except the case of MR5, and REMP and S-REMP provide higher PSNR than the other protocols.

Fig. 10 shows the delivery ratio of each receiver in scenario 2. Note that in scenario 2, the channel qualities of the multicast receivers except MR1 degrade compared to that in scenario 1. Comparing Fig. 10a with Fig. 9a, we can see that the delivery ratio of 802.11n multicast decreases in scenario 2. This is because the available bandwidth for downlink transmissions is reduced due to the use of a lower MCS by uplink senders. Fig. 10b shows that DPMM achieves even worse performance than 802.11n multicast due to inefficient MCS adjustments and retransmissions. Fig. 10c shows that REMP does not achieve 100 percent delivery ratio, but it still shows better performance than the 802.11n multicast and DPMM.

Fig. 10d shows the scalable video-aware feature of S-REMP. As we can see from Fig. 10c, in scenario 2, the available bandwidth at the AP is not high enough to transmit all the layers when REMP is applied. Therefore, in S-REMP, frames that contain BL are always transmitted with a robust MCS while frames that contain EL1 or EL2 frames are transmitted with higher MCS when frames are dropped due to the overloading of the AP's queue. As a result, all the multicast receivers achieve almost 100 percent delivery ratio for BL, while they achieve different delivery ratios for EL1 and EL2 according to the channel condition.

The average PSNR values in scenario 2 are presented in Table 6. REMP shows higher PSNR values than 802.11n multicast and DPMM. In contrast to scenario 1, S-REMP outperforms REMP. In S-REMP, an acceptable video quality (over 30 dB) is always guaranteed to all receivers. Apparently, this is because S-REMP provides high delivery ratio for BL to all the multicast receivers. Note that according to a hierarchical layer structure of scalable video coding, the impact of BL on video quality is much higher than that of EL1 or EL2. PSNR at each receiver in S-REMP decreases as the increase of receiver's distance from the AP due to the differences in the delivery ratio for EL1 and EL2. However, we can see that PSNR of MR5 in S-REMP is much higher than PSNR of MR1 in REMP.

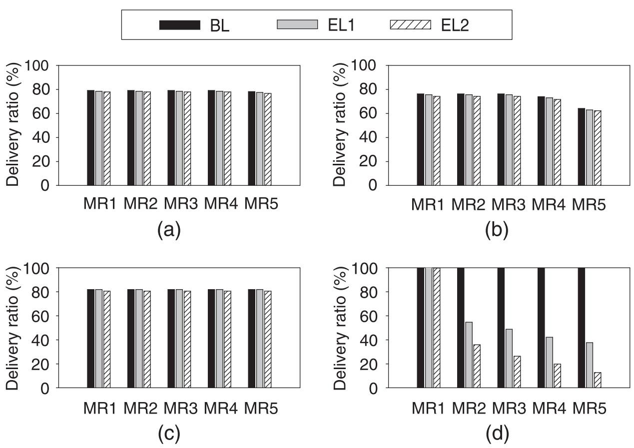

Fig. 10. Delivery ratio of each receiver in the scenario 2. (a) 802.11n. (b) DPMM. (c) REMP. (d) S-REMP.

7 Conclusion

In this paper, we proposed a novel MAC-level multicast protocol named REMP that enhances the reliability and efficiency of multicast transmissions in IEEE 802.11n WLANs. In REMP, AP selectively retransmits erroneous multicast frames and dynamically adjusts MCS under varying channel conditions based on the advanced feedback mechanism from multicast receivers. In addition, we proposed S-REMP, an extended version of REMP, for efficient delivery of scalable video over IEEE 802.11n WLANs. In S-REMP, different layers of scalable video can be transmitted with different MCSs to provide the minimal video quality to all users while providing a higher video quality to users exhibiting better channel conditions. Via extensive simulation results, we proved the effectiveness of the proposed protocols.

As our future work, we are planning to implement REMP and S-REMP by modifying open-source WLAN drivers and measure the performance in real systems.

Acknowledgments

The authors wish to thank the editor and anonymous reviewers for their helpful comments that improved the quality of the paper. This research was supported by a World Class University (WCU) program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology (R31-2010-000-10100-0), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2010-0024938), and the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA) (NIPA-2011-C1090-1131-0009).

References

[1] *IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE, 1999.

[2] *IEEE 802.11n-Draft, Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput, Draft 4.0,* IEEE, 2008.

[3] A. Kamerman and L. Monteban, "WaveLAN II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical J.,* vol. 2, no. 2, pp. 118-133, 1997.

[4] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," *Proc. ACM MobiCom,* July 2001.

[5] B. Sadeghi, V. Kanodia, A. Sabharwa, and E. Knightly, "OAR: An Opportunistic Auto-Rate Media Access Protocol for Ad Hoc Networks," *Proc. ACM MobiCom,* Sept. 2002.

[6] K. Tang and M. Gerla, "MAC Reliable Broadcast in Ad Hoc Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '01),* Oct. 2001.

[7] M.T. Sun, L. Huang, A. Arora, and T.H. Lai, "MAC Layer Multicast in IEEE 802.11 Wireless Networks," *Proc. Int'l Conf. Parallel Processing (ICPP '02),* Aug. 2002.

[8] J. Kuri and S.K. Kasera, "Reliable Multicast in Multi-Access Wireless LANs," *ACM Wireless Networks,* vol. 7, no. 4, pp. 359-369, 2001.

[9] S. Choi, N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Leader-Based Rate Adaptive Multicasting for Wireless LANs," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '07),* Nov. 200.

[10] A. Basalamah, H. Sugimoto, and T. Sato, "A Rate Adaptive Multicast Protocol for Providing MAC Layer Reliability in WLANs," *IEICE Trans. Comm.,* vol. E89-B, no. 10, pp. 2733-2740, 2006.

[11] J. Villalón, P. Cuenca, L. Barbosa, Y. Seok, and T. Turletti, "Cross-Layer Architecture for Adaptive Video Multicast Streaming over Multirate Wireless LANs," *IEEE J. Selected Areas in Comm.,* vol. 25, no. 4, pp. 699-711, May 2007.

[12] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable H.264/MPEG4-AVC Extension," *Proc. IEEE Int'l Conf. Image Processing (ICIP '06),* Oct. 2006.

[13] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-Time Video Multicast in WiMAX Networks," *Proc. IEEE INFOCOM,* Apr. 2008.

[14] Q. Ni, T. Li, T. Turletti, and Y. Xiao, "AFR Partial MAC Proposal for IEEE 802.11n," IEEE 802.11n Working Document 802.11-04-0950-00-000n, Aug. 2004.

[15] X. Wang, L. Wang, Y. Wang, and Y. Zhang, "Supporting MAC Layer Multicast in IEEE 802.11n: Issues and Solutions," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '09),* Apr. 2009.

[16] Joint Scalable Video Model, Reference Software, http://ip.hhi.de/imagecom_GI/savce/downloads/SVC-Reference-Software.htm, 2011.

[17] SVEF: Scalable Video-Streaming Evaluation Framework, http://svef.netgroup.uniroma2.it, 2011.

[18] V. Srinivas and L. Ruan, "An Efficient Reliable Multicast Protocol for 802.11-Based Wireless LANs," *Proc. IEEE Int'l Symp. World of Wireless, Mobile and Multimedia Networks and Workshops (WoWMoM '09),* June 2009.

[19] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath, "Dircast: A Practical and Efficient Wi-Fi Multicast System," *Proc. IEEE 17th Int'l Conf. Network Protocols (ICNP '09),* Oct. 2009.

[20] N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Leader-Based Multicast Service (LBMS) in IEEE 802.11v Networks," *Proc IEEE Seventh Consumer Comm. and Networking Conf. (CCNC '10),* Jan. 2010.

[21] Z. Li and T. Herfet, "Beacon-Driven Leader Based Protocol over a GE Channel for MAC Layer Multicast Error Control," *Int'l J. Comm., Network and System Science,* vol. 1, no. 2, pp. 144-153, 2008.

[22] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *J. Computer Networks and ISDN,* vol. 17, pp. 1-14, 1989.

Wan-Seon Lim received the BS, MS, and PhD degrees in computer science and engineering from the Pohang University of Science and Technology (POSTECH), Korea. He is currently a postdoctoral researcher at POSTECH Information Research Laboratories (PIRL). His research interests include wireless LAN MAC protocols, ad hoc networks, and video streaming over warless networks. He is a member of the IEEE.

Dong-Wook Kim received the BS degree in computer engineering from Kyoungpook National University, Daegu, Korea, in 2005. He is currently working toward the PhD degree in the Department of Computer Science and Engineering at the Pohang University of Science and Technology (POSTECH), Korea. His research interests include wireless LAN MAC protocols, wireless mesh networks, and mobile IP.

Young-Joo Suh received the BS and MS degrees in electronics engineering from Hanyang University, Seoul, Korea, in 1985 and 1987, respectively, and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 1996. He is currently a professor in the Department of Computer Science and Engineering/the Division of IT Convergence Engineering at the Pohang University of Science and Technology (POSTECH), Korea. From 1988 to 1990, he was a research engineer at the Central Research Center of LG Electronics, Inc., Seoul, Korea. From 1990 to 1993, he was an assistant professor in the Department of Computer Science and Engineering at Chung-Cheong College, Korea. After receiving the PhD degree, he worked as a post-doctoral researcher in the Computer Systems Research Laboratory in the School of Electrical and Computer Engineering at the Georgia Institute of Technology from 1996 to 1997. From 1997 to 1998, he was a research fellow at the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. His current research interests include wireless LAN MAC protocols, mobility management, ad hoc networks, and 4G wireless mobile networks. He is a member of the IEEE and the IEEE Communications Society.