Cooperative MAC Protocol based on TXOP and Block ACK in IEEE 802.11e WLANs

Sunmyeng Kim
Department of Computer Software Engineering, Kumoh National Institute of Technology
1 Daehak-ro, Gumi, Gyeongbuk 730-701, South Korea
sunmyeng@kumoh.ac.kr

Abstract—The IEEE 802.11 MAC protocol cannot support quality of service (QoS) requirements. In order to provide QoS in the IEEE 802.11 MAC protocol, the IEEE 802.11e has been standardized. These standards provide multiple transmission rates, which can be changed dynamically according to the channel condition. When using multiple transmission rates, the capacity of wireless LAN improves, but the performance anomaly problem may occur. Cooperative communications were introduced to alleviate the performance anomaly problem with the help of relay nodes with higher transmission rates. Previous cooperative communications protocols are based on the IEEE 802.11 DCF MAC protocol. That is, none of them takes the IEEE 802.11e EDCA into consideration. In this paper, we apply the EDCA features such as TXOP and block ACK for cooperative communications. Simulation results show that the proposed protocol works well and improves network performance.

Keyword-Cooperative Communications, MAC, TXOP, Block ACK, IEEE 802.11e

I. INTRODUCTION

The IEEE 802.11 wireless LAN 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 nodes [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 the two-way handshaking (DATA-ACK) mechanism. The RTS/CTS access method uses the four-way handshaking (RTS-CTS-DATA-ACK) mechanism to reserve the channel before transmitting long data packets. This technique is introduced to avoid the hidden terminal problem.

The widespread use of multimedia applications requires new features such as high bandwidth and small average delay in wireless LANs. Unfortunately, the IEEE 802.11 MAC protocol cannot support quality of service (QoS) requirements [2, 3]. In order to support multimedia applications with tight QoS requirements in the IEEE 802.11 MAC protocol, the IEEE 802.11e has been standardized [4]. It introduces a contention-based new channel access mechanism called enhanced distributed channel access (EDCA). The EDCA supports the QoS by introducing four access categories (ACs). To differentiate the ACs, the EDCA uses a set of AC specific parameters, which include minimum contention window, maximum contention window, and arbitration inter-frame space (AIFS). The EDCA also introduces a TXOP (Transmission Opportunity) parameter to provide service differentiation and QoS of the traffic. A node can continuously transmit multiple packets for the duration of a TXOP. In the IEEE 802.11 MAC protocol, each data packet is immediately acknowledged after a successful transmission. This causes high overhead. The IEEE 802.11e MAC defines the block ACK scheme to reduce the ACK transmission overhead by integrating multiple ACKs for a number of data packets into a bitmap that is contained in a block ACK packet.

The most fundamental method available to enhance the capacity of wireless LAN is providing higher transmission rate at the physical layer. IEEE 802.11a/b/g were standardized to expand the physical layer capable of offering higher transmission rates. These standards provide multiple transmission rates, which can be changed dynamically according to the channel condition. To utilize several rates, it is required to deploy rate adaptation schemes at the MAC layer [5].

When using multiple transmission rates, the capacity of wireless LAN improves, but the performance anomaly problem may occur [6]. In a wireless LAN using carrier sense multiple access with collision avoidance (CSMA/CA), the probability of channel access is same regardless of the transmission rates of nodes. When a node gets an opportunity to access a channel, a node with lower transmission rate tends to occupy more channel time than a node with higher transmission rate. Therefore, when there are more nodes with lower transmission rate, then overall network performance decreases. That is, in a wireless LAN supporting multiple transmission rates, the network performance is affected by nodes with lower transmission rates.

Cooperative communications were introduced to alleviate the performance anomaly problem with the help of relay nodes with higher transmission rates [7], [8]. The cooperative communications are based on the fact that the transmission is much faster when sending data packets to a destination node through a relay node with higher transmission rate, rather than sending data directly to the destination node at low transmission rate. To apply the cooperative communications in wireless LAN, several MAC protocols have been proposed [7]-[18]. They are based on the IEEE 802.11 DCF MAC protocol. That is, none of them takes the IEEE 802.11e EDCA into consideration. Therefore, in the previous protocols, a node can transmit only one data packet when it gains the right to access the channel.

To the best of our knowledge, none of the existing work has focused on cooperative communications in the IEEE 802.11e EDCA. Therefore, we propose a novel cooperative MAC protocol for QoS enhancement in WLANs based on the IEEE 802.11e EDCA MAC protocol. It is called QC-MAC (QoS Cooperative MAC). In the proposed protocol, a node can transmit multiple data packets consecutively until the duration of transmission exceeds the specific TXOP time period. By using the Block ACK procedure, a receiver acknowledges a block of received data packets.

The paper is organized as follows. In Section II, we give a brief introduction of the IEEE 802.11e EDCA and the CoopMAC protocol, which is one of typical cooperative MAC protocols. In Section III, the proposed QC-MAC protocol is presented in detail. In Section IV, performance studies are carried out through simulation results. Finally, we draw a conclusion in Section V.

II. RELATED WORK

In this Section, we summarize the IEEE 802.11e EDCA and the CoopMAC protocol proposed in [7].

A. IEEE 802.11e EDCA

The IEEE 802.11 MAC protocol cannot support QoS requirements. In order to enhance the QoS support of the IEEE 802.11 WLAN, the IEEE 802.11e has been standardized. It introduces a new medium access method called hybrid coordination function (HCF), which combines a contention-based enhanced distributed access mechanism (EDCA) and a controlled channel access mechanism (HCCA). The EDCA is an enhanced variant of the DCF. In the DCF, all stations contend for the channel with the same priority. On the other hand, the EDCA supports several priority levels by introducing an access category (AC) concept. A node has up to four ACs to support eight user priorities. Each AC is implemented as a separate queue. Each packet arrives at the MAC layer with a priority from higher layer, and is mapped to one AC according to the priority. AC 3, AC 2, AC 1, and AC 0 are for voice, video, best-effort data, and background traffic, respectively. In order to differentiate the ACs, the EDCA uses a set of AC specific parameters, which include minimum contention window (CWmin[i]), maximum contention window (CWmax[i]), arbitration inter-frame space (AIFS[i]), and transmission opportunity (TXOP[i]) for AC i ($i = 0, \dots, 3$). The AIFS is, at least, distributed inter-frame space (DIFS) long and is calculated with the AIFS number (AIFSN[i]). The duration of AIFS[i] is defined by AIFS[i] = SIFS + AIFSN[i] * *aSlotTime*, where SIFS is a short inter-frame space, and *aSlotTime* is the duration of a slot time. For $0 \leq i < j \leq 3$, the EDCA has CWmin[i] ≥ CWmin[j], CWmax[i] ≥ CWmax[j], and AIFSN[i] ≥ AIFSN[j]. Note that, in the preceding inequalities, at least one must be “not equal to.” Once a TXOP is obtained using a backoff, a station is allowed to transmit more than one data packets consecutively during the TXOP. The EDCA assigns a smaller CW and shorter AIFS to higher priority classes in order to ensure that in most cases, higher priority classes experiences lower mean waiting and backoff times than lower priority ones. Therefore, in the EDCA, support of QoS can be achieved statistically by reducing the probability of medium access for lower priority classes.

In the IEEE 802.11e, a block ACK scheme is defined in order to overcome overheads by reducing the number of control packets for multiple data transmissions. Basically, the block ACK scheme allows multiple data transmissions without an immediate acknowledgement separated by SIFS time period. The single acknowledgement packet, block ACK (BA), is sent by a receiver for a block of data packets transmitted by a source. A block ACK selectively acknowledges or negatively acknowledges all the transmitted data packets at once by using a bitmap.

B. CoopMAC Protocol

In the CoopMAC protocol, a source node at first sends data packets to a helper node with higher transmission rate, which forwards them to an AP (Access Point) to improve network throughput and to reduce transmission delay.

Each node maintains a CoopTable, which has information such as transmission rate between a source node and a helper node, transmission rate between a helper node and a destination node, and update time of entries. A source node overhears the transmissions of other nodes, and then estimates their transmission rates to update the CoopTable.

After overhearing transmissions and making estimates, helper nodes are stored in the CoopTable if the following condition is met:

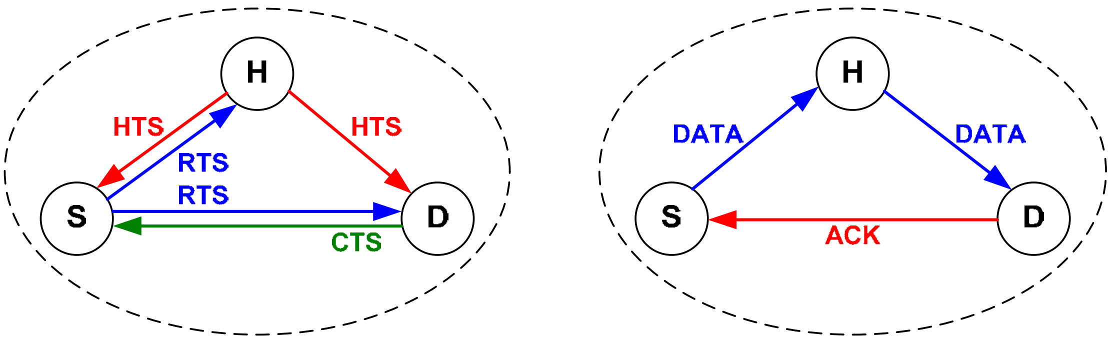
$$\frac{1}{R_{S,H}}+\frac{1}{R_{H,D}}>\frac{1}{R_{S,D}} \tag{1}$$

where, $R_{S,H}$, $R_{H,D}$, $R_{S,D}$ denote transmission rate between a source node (*S*) and a destination node (*D*), that between the source node and a helper node (*H*), and that between the helper node and the destination node, respectively.

When there are data packets to send in a queue, a source node looks for helper node candidates in the CoopTable. If there are one or more helper node candidates, then a node with the least packet transmission time is selected as a helper node. Packet transmission time is $L/R_{S,H}+L/R_{H,D}$. In here, overhead is omitted and L is the size of data packet in bits.

After successfully finding a helper node, the source node sends a CoopRTS packet to the selected helper node. The helper node checks whether it can provide the service wanted by the source node after receiving the CoopRTS packet. If so, the helper node sends a HTS (Helper ready To Send) packet. Finally, the destination node sends a CTS packet to the source node.

After receiving the CTS packet, the source node sends a data packet to the helper node, which forwards the packet to the destination node. However, if there is no need of cooperative transmission between the source node and the destination node, or if the source cannot find a helper node successfully, then the CoopMAC protocol acts as the legacy 802.11 DCF. Fig. 1 shows the packet exchange procedure in the CoopMAC protocol.



(a) Control packet exchange (b) Data packet exchange

Fig. 1. Flow of packet exchange in the CoopMAC protocol

III.QC-MAC PROTOCOL

In this Section, we present a basic idea of our proposed QC-MAC protocol. Although the proposed QC-MAC protocol has the similar procedure of exchanging packets to that of the CoopMAC protocol, it uses a different method in transmitting data packets.

In the proposed protocol, the TXOP and block ACK features of the EDCA are applied for cooperative communications to improve network performance and to overcome overheads by reducing the number of control packets for multiple data transmissions.

We describe how to select helper nodes in subsection III.A, and then how to decide block size in subsection III.B. We describe new packet format and data transmission procedure adopted in the proposed protocol in subsection III.C.

A. Helper Node Selection

As shown in Fig. 2, each node maintains a table, referred to as the *QCTable* (QoS Cooperative Table). A node overhears transmissions of packets such as RTS, CTS, DATA, and ACK by other nodes, and then updates its QCTable. The QCTable contains 5 fields. Data in the first field is MAC address of a helper node. In the time field, time of the last packet received from the helper node is recorded. In the transmission rate fields, transmission rates ($R_{S,H}$, $R_{H,D}$) between source node S and helper node H, and between helper node H and destination node D are stored, respectively. In the last field, channel credit (C_H) of the helper node is stored. The channel credit tracks the channel status of the particular helper node. This value is used to calculate the block size, which is the number of data packets to be transmitted and to be acknowledged by a single block ACK. How to calculate the size is described in detail in subsection III.B.

MAC Address of Helper	Time	Transmission Rate (*S-H*)	Transmission Rate (*H-D*)	Channel Credit
H_1	T_1	$R_{S,H1}$	$R_{H1,D}$	C_{H1}
…	…	…	…	…
H_n	T_n	$R_{S,Hn}$	$R_{Hn,D}$	C_{Hn}

Fig. 2. Format of the QCTable

When there are data packets to send in a queue, a source node looks for helper node candidates in the QCTable. If there are one or more helper node candidates, then a node with the least packet transmission time is selected as a helper node. Packet transmission time is $L/R_{S,H} + L/R_{H,D}$. In here, overhead is omitted and L is the size of data packet in bits.

B. Block Size Decision

After selecting a helper node, a source node decides the block size based on channel credit in the QCTable. Channel credit is the channel status of the selected helper node. The start value of channel credit is 50, which may be changed from 1 to 100 according to the status of the channel. This value can be changed by using the success ratio of packet delivery. The success ratio of packet delivery at helper node *i* is as follows:

$$Succ_i = \frac{N_{ack}}{N_{tx}} \tag{2}$$

where, N_{tx} is the total number of data packets sent from a source node and a helper node; N_{ack} is the number of ACK packets. The value of $Succ_i$ is located between 0 and 0.5. For cooperative communications, a source node sends data packets to a helper node, which in turn sends them to a destination node. When receiving all data packets without error, the destination node sends ACK packet for each data packet. If there is no error as above, to deliver a data packet, it is necessary to send the data twice, while delivering ACK packet for one time. Therefore, in this case, $Succ$ is 0.5. If there is an error, then it is necessary to consider the retransmission of the data packet, this value decreases.

For the obtained $Succ_i$, the range of values is readjusted by using the following equation:

$$rSucc_i = \langle Succ_i \cdot 20 \rangle - 5 \tag{3}$$

where, $\langle x \rangle$ is rounded off value of x and $rSucc_i$ has a value between -5 and 5. A source node changes the channel credit (C_i) by using $rSucc_i$ as follows:

$$C_i = C_i + rSucc_i \tag{4}$$

By using the channel credit of helper node *i*, a source node decides block size (BS) as follows:

$$BS = \left\lceil \frac{C_i}{100} \cdot MBS \right\rceil \tag{5}$$

where, $\lceil x \rceil$ is raised value of x , and MBS is the maximum block size. In the denominator, 100 is the maximum value of the channel credit.

C. Data Transmission Procedure

Before describing the data transmission procedure, we introduce new packet formats and timers in the proposed QC-MAC protocol.

For cooperative communications, a 1-byte Relay Control field is added to existing frame format of IEEE 802.11 as shown in Fig. 3. The Relay Control field is composed of six subfields. The first subfield, Last Data Flag, shows whether the current data packet is the last one among data packets delivered during TXOP time period. That is, '0' shows that the current data packet is not the last data, while '1' shows that it is the last data. The next four subfields represent Status Bitmap. These four bits are the number of data packets to be acknowledged. Each bit in the Status Bitmap shows the status of a data packet (success/failure of delivery). '0' means the reception of erroneous data, while '1' means the reception of data without error. The last subfield is the block size determined at subsection III.B. Whether to use each subfield in the Relay Control field may be differed according to the type of packet. For RTS, HTS and CTS, only the last subfield, Number of DATAs, is used. For ACK packet, all the subfields except the first one are used. For a data packet, every subfield is used.

Although it is possible to include errors while delivering, a data packet does not contain any error when each node sends it. Thus, its Status Bitmap is set to 1. In the proposed method, on the assumption that the maximum block size is 4, the Relay Control field is defined as 1 byte. If the maximum block size is larger, then the size of Relay Control field can be adjusted accordingly, so the proposed method can be scaled easily according to diverse environments.

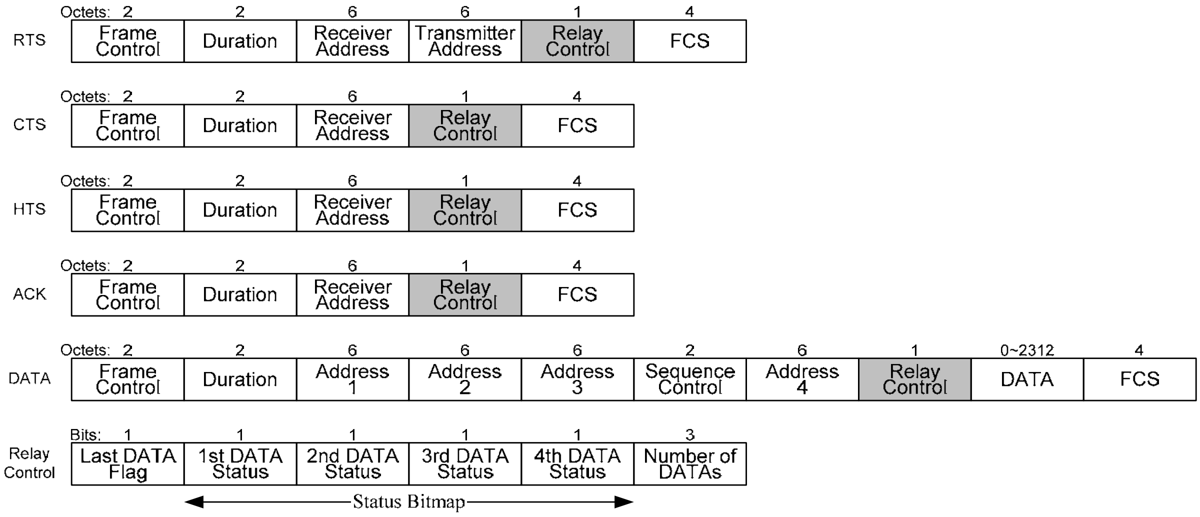


Fig. 3. Frame format

In addition to the timers (CTS Timeout and ACK Timeout) offered at IEEE 802.11, we also use additional two timers (Helper Timeout and Source Timeout). A source node and a helper node use Helper timeout and Source timeout, respectively. The source node shall wait Helper timeout amount of time without receiving a data packet from the helper node before concluding that the data packets the source node transmitted failed. Helper timeout is determined by (SIFS + 2·*aSlotTime*). The helper node shall wait Source timeout amount of time without receiving data packets from the source node. Source timeout is determined by (SIFS + 1·*aSlotTime*).

After selecting a helper node, a source node sends data packets to a destination node according to the RTS-HTS-CTS-DATA-ACK procedure. First, a source node sends an RTS packet to the helper node and the destination node. After receiving the RTS packet, the helper node sends a HTS packet to the source node and the destination node. After receiving the HTS packet, the destination node sends CTS packet to the helper node and the source node. After that, the source node sends data packets to the helper node, which forwards them to the destination node. The source node checks whether the data packets are transmitted successfully to the helper node by receiving the data packet from the helper node. After receiving the data packets, the destination node sends an ACK packet. After relaying the data packets, the helper node receives the ACK packet from the destination node to check whether the relayed data packets are sent successfully. If the relay transmission of the data packets is failed, then the packets are retransmitted by the helper node, not by the source node.

Hereafter, we describe in detail operating procedures of the proposed method according to possible errors, on the assumption that the block size is 3.

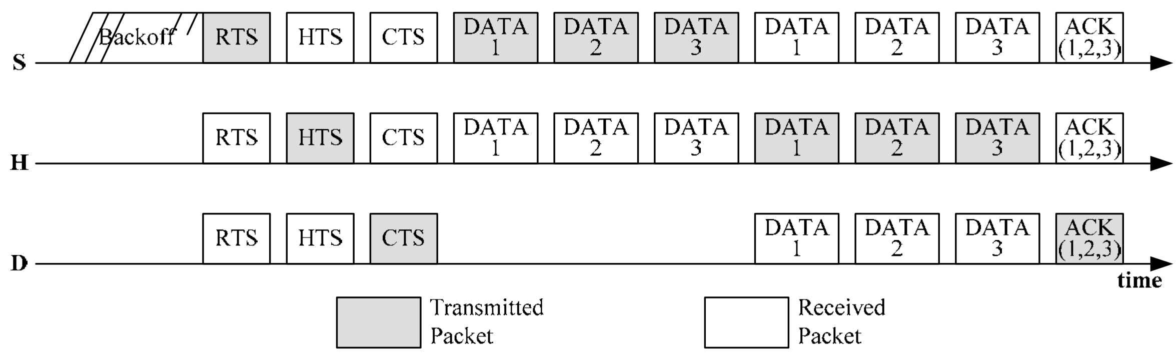


Fig. 4. Normal procedure without error

Fig. 4 shows a normal operation procedure without error. Assume that there are many data packets to send in a queue of a source node. When there are data packets to deliver, the source node (S) performs backoff process to access the channel. When backoff counter becomes 0, it sends RTS packet to the selected helper node (H) and the destination node (D). When H receives the RTS packet, it sends HTS packet to S and D. And then, D sends CTS packet to S and H. Relay Control field value included in RTS, HTS and CTS is 00000011. That is, the Number of DATAs subfield is set to 3, and the remaining subfields are set to 0. When S receives the CTS packet, it sends three data packets (DATA 1, 2 and 3) consecutively to H. Again, H sends these packets to D. Relay Control field value of the data packets sent by S and H is 01110011. That is, the Last Data Flag subfield is set to 0 as the current packet is not the last one among data packets to send during TXOP limit. And the first three bits of the Status bitmap are set to 1, the last bit is set to 0. The Number of DATA subfield is set to 3. After receiving three data packets, D sends ACK packet. The Relay Control field value of ACK packet is 01110011. As three data packets are received without error, the first three bits are set to 1 at the Status Bitmap, and the Number of DATA is set to 3. After receiving ACK packet, S sends the following three data packets (DATA 4, 5 and 6) consecutively to H. After that, the procedure is the same as that described above. Although it is not depicted in the figure, there is SIFS time between packet deliveries.

To simplify the explanation, we omit the backoff and RTS-HTS-CTS delivery processes in figures from now on. That is, we describe operations of exchanging data packets in the TXOP limit by a source node, which acquires channel access through channel competition.

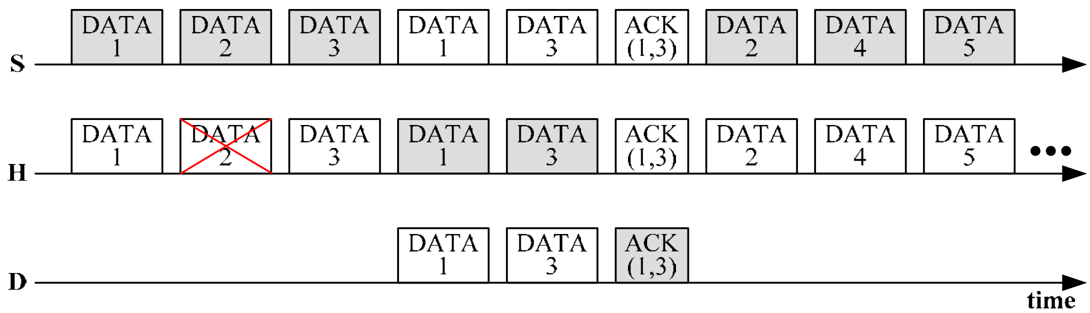


Fig. 5. Example of occurring errors in some data packets sent by source node S to helper node H

Fig. 5 shows an example of occurring errors in some data packets sent by source node S to helper node H. S sends three data packets (DATA 1, 2 and 3) consecutively to H. During the delivery, error occurs in DATA 2. Thus, H sends only DATA 1 and 3 to D. Relay Control field value of the data packets sent by H is 01010011. That is, as H receives erroneous DATA 2, he second bit of the Status bitmap is set to 0. After receiving the data packets sent by H, S can check whether the delivery is successful through this bitmap. When D receives DATA 1 and 3 without error, it sends ACK packet. Relay Control field value in the ACK packet is 01010011. After receiving the ACK packet, S sends DATA 2, 4 and 5. After that, the procedure is the same as that described above.

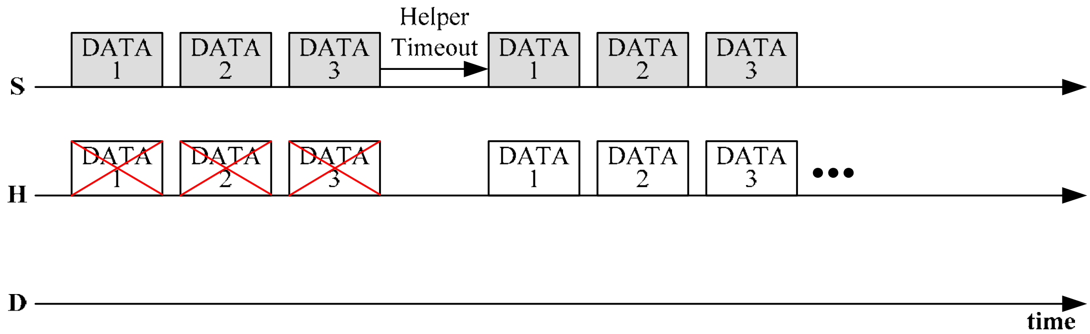


Fig. 6. Example of occurring error in every data packet sent by S to H

Fig. 6 shows an example of occurring error in every data packet sent by S to H. After sending DATA 1, 2 and 3, S waits until H forwards the data packets to D. However, as error occurs in every packet, H does not send them to D. Therefore, Helper Timeout occurs at S, and S retransmits DATA 1, 2 and 3. After that, the procedure is the same as that in normal procedure. Helper Timeout is time duration of waiting a signal from H by a source node after sending the data. It does not want to get accurate data reception, but want to receive a signal. Thus, when H sends a signal, the Helper Timeout does not occur.

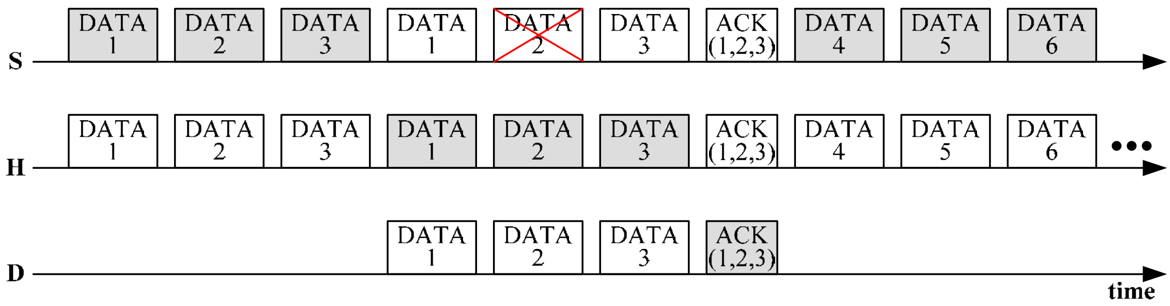


Fig. 7. Example of error occurring at S in some data packets sent by H

Fig. 7 shows that in some data packets sent by H to S and D, errors are occurred at S. S sends three data packets (DATA 1, 2 and 3) consecutively to H. Again, H sends these packets to D. Among the data packets sent by H, S receives erroneous DATA 2. However, from Relay Control field value of DATA 1 and 3 received without error, it can be confirmed that all the data packets sent by S are transmitted to H without error. Relay Control field value of the data packets sent by S and H is 01110011. Since the Status bitmap value is 111, it shows that all three data packets sent by S are delivered to H without error. After receiving three data packets, D sends ACK packet. After receiving the ACK packet, S sends the next three data packets (DATA 4, 5 and 6) consecutively to H.

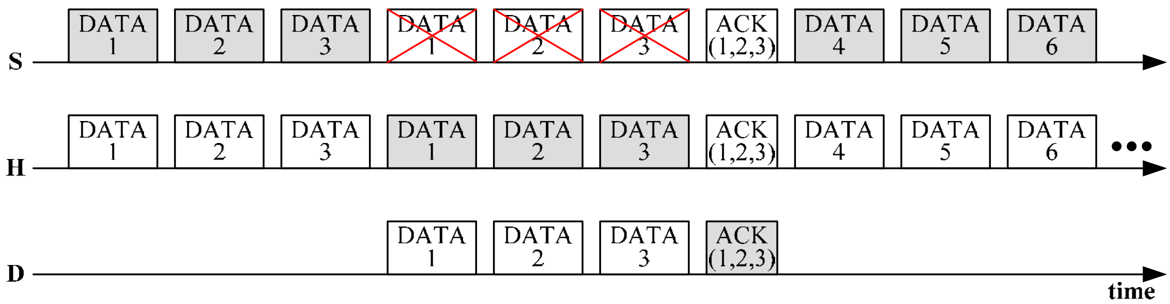


Fig. 8. Example of error occurring at S in every data packet sent by H

Fig. 8 shows an example, in which error occurred at every data packet sent by H to S and D. H receives every data packet sent by S without error. However, every data packet received by S from H is erroneous. Since S receives the signal from H, the Helper Timeout does not expire at S. Through this signal, S can confirm that among the data packets sent by it, H receives at least one packet without error. Then, S waits to receive ACK packet from D. Since D receives DATA 1, 2 and 3 without error, it sends ACK packet with Status bitmap 111. Through the received ACK packet, S confirms that DATA 1, 2 and 3 is transmitted successfully, and then sends the next data packet (DATA 4, 5 and 6).

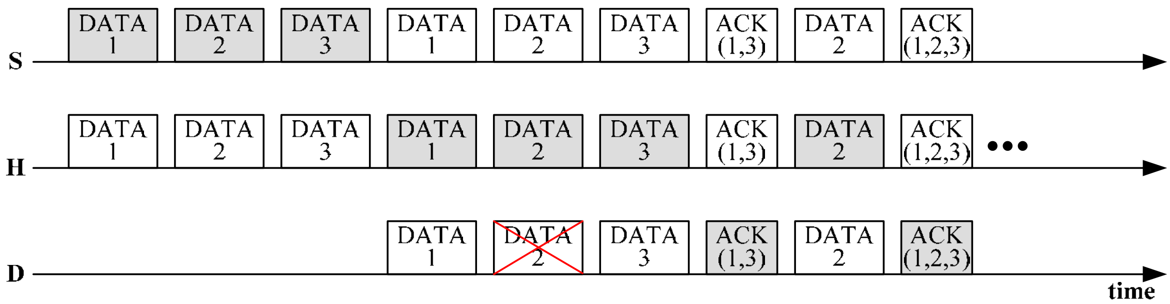


Fig. 9. Example of errors occurred in some data packets received by D from H

Fig. 9 shows an example of errors occurred in some data packets received by D from H. Since D received erroneous DATA 2, it sends ACK packet including status information on DATA 1 and 3. Value in Relay Control field included in the ACK packet is 01010011. In Figs. 7 and 8, after receiving the ACK packet, S sends the next data packets. However, in Fig. 9, after receiving the ACK packet, S does not send the next data packets, but waits for H to retransmit DATA 2. S knows from the data packets received from H that H receives DATA 2 without error. If H receives DATA 2 with error from S, then S sends DATA 2, 4 and 5 after receiving the ACK packet. H retransmits the erroneous DATA 2. When receiving DATA 2 without error, D sends ACK packet including status data of DATA 1, 2 and 3.

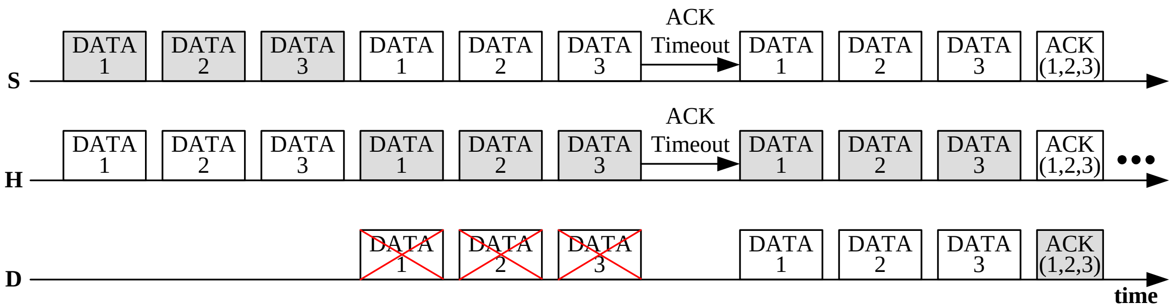


Fig. 10. Example of errors occurred in all data packets received by D from H

Fig. 10 shows an example of errors occurred in all data packets received by D from H. In this case, D does not send ACK packet. And ACK timeout occurs at S and H. Since S knows that H receives the data packets without error, it does not retransmit the data packets, even if ACK timeout occurs. H retransmits every data packet sent before, if ACK timeout occurs. When receiving the data packets without error, D sends ACK packet.

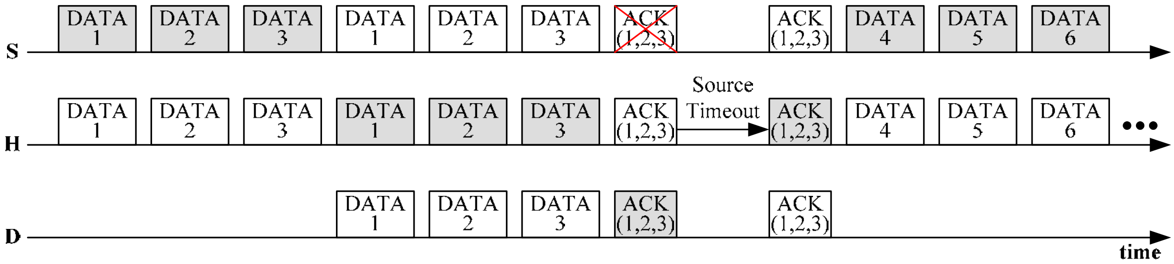


Fig. 11. Example of error occurred in ACK packet received by S from D

Fig. 11 shows an example of error occurred in ACK packet received by S from D. D receives DATA 1, 2 and 3 from H without error and sends ACK packet. H receives the ACK packet without error, but S receives erroneous ACK packet. Since H receives the ACK packet for every data packet sent before, it sets Source Timeout, and waits for S to send next data packets (DATA 4, 5 and 6). However, as S does not receive the ACK packet, it does not send the next data packets. Instead, as S received DATA 1, 2 and 3 from H previously, it sets Helper Timeout, and waits for H to retransmit the data packets or send ACK packet. Since Source Timeout is shorter than Helper Timeout, Source Timeout occurs first and H sends ACK packet received from D. When receiving the packet, S sends the next data packets (DATA 4, 5 and 6). If Helper Timeout occurs, then S retransmits all the data packets not received ACK packets to H.

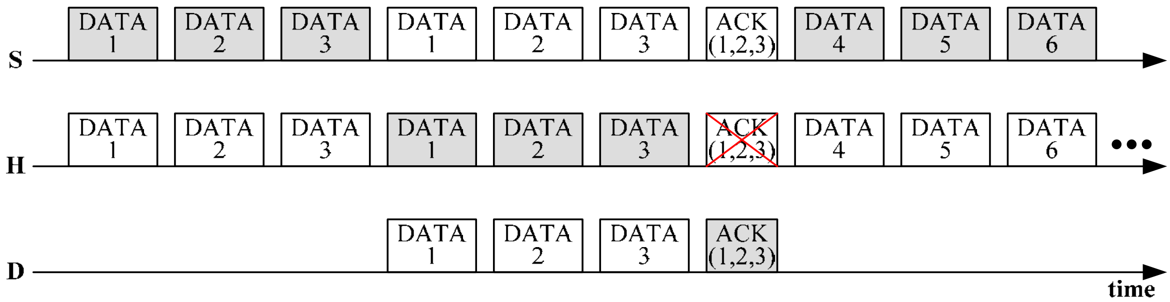


Fig. 12. Example of error occurred in ACK packet received by H

Fig. 12 shows an example of error occurred in ACK packet received by H. D receives DATA 1, 2 and 3 from H without error and sends ACK packet. S receives the ACK packet without error, but H receives it with error. H sets Source Timeout and then waits for S to send the next data packets. Since S receives the ACK packet without error, it sends the next packets, DATA 4, 5 and 6. Before occurring the Source Timeout, H receives the data packets from S. By receiving the data packets, H confirms that Status Bitmap value of the received ACK packet with error is 111. After completing the delivery of DATA 1, 2 and 3, it proceeds to the delivery process of DATA 4, 5 and 6.

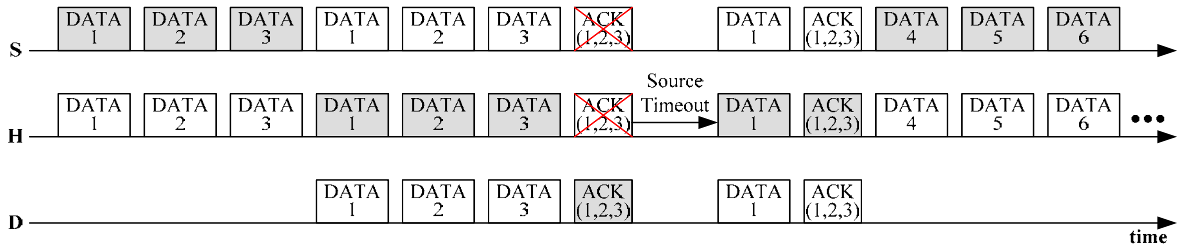


Fig. 13. Example of error occurred in ACK packet received by S and H

Fig. 13 shows an example of error occurred in ACK packet received by S and H. D receives DATA 1, 2 and 3 from H without error and sends ACK packet. Both S and H receive erroneous ACK packet. Thus, H sets Source Timeout and waits for S to send the data packets. In addition, S sets Helper Timeout and waits for H to send the data packets. Since Source Timeout is shorter than Helper Timeout, Source Timeout occurs first, and then H sends the data packet (e.g., DATA 1 in Fig. 13) with the lowest sequence among data packets sent previously. As D receives redundant data packet, it resends the ACK packet sent before. After receiving the retransmitted ACK packet, S sends the next data packets.

Until now, we explain data transmission procedures in diverse erroneous situations. Other situations not mentioned here can be solved by combining the above procedures.

IV. SIMULATION RESULTS

Let us discuss the simulation results of the proposed TRCCL protocol. To validate the proposed protocol, we compare them to the results of the CoopMAC protocol. In the simulation, we consider the topology shown in Fig. 14. Nodes are randomly deployed within the transmission range of an AP. Data rate of each node is determined according to the distance to the AP (refer to TABLE I). We use the Block size of 3 and TXOP limit of 8000*us*. Every node sends data packets to the AP.

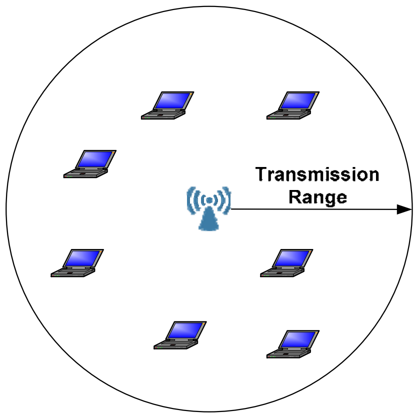


Fig. 14. Simulation topology

TABLE I
Data rates and transmission ranges of 802.11g [19]

Data rate (Mbps)	6	9	12	18	24	36	48	54
Range (m)	122	107	96	85	75	61	42	31

In the simulations, we use the negative exponential distribution to get the lengths of the data packet inter-arrival times. The average inter-arrival time of the distribution with arrival rate parameter λ is $1/\lambda$. In the simulation, the average inter-arrival time is set to 6000 us ($\lambda = 0.00016667$). A constant data packet size of 1500 bytes is used.

Main performance metrics of interest are throughput and delay. Delay is the time elapsed from the moment a packet arrives at the MAC layer queue until the packet is successfully transmitted to the destination node including the receipt of acknowledgement.

Figs. 15 and 16 show the results of simulation. In the figures, PKT(n) means that each node generates n data packets at each packet arrival time. That is, the larger n value is, the more data packets are generated, and the larger the volume of transmission becomes.

Fig. 15 shows the throughput based on the number of nodes. The proposed QC-MAC protocol always shows better performance than existing CoopMAC protocol. In the QC-MAC protocol, as PKT(n) is increasing, the throughput is also improving. This is because that as the larger PKT(n) becomes, the larger the number of packets generated by each node, and therefore the volume of transmitted data is increasing. However, as the number of nodes is increasing, the increase of the throughput becomes slower. In addition, it can be seen that PKT(n) values become converging. As the number of nodes is increasing, the probability of collision is also increasing, thus the throughput cannot be growing continuously. In the CoopMAC protocol, the throughput of PKT(3) is lower than those of PKT(5) and PKT(7), only when the number of nodes is small. And as the number of nodes becomes increasing, there is little difference in terms of throughput. As the number of nodes is growing, difference in the throughput of the two protocols becomes larger. In the proposed protocol, TXOP and Block ACK features of EDCA are applied to cooperative communications. Accordingly, it is possible to send multiple data packets with a backoff process and exchanging of control packets. In this way, the overhead is reduced, while the performance is improved. However, the CoopMAC protocol does not apply these features, it has larger overhead, and as a result, has poor performance. Also, in the proposed protocol, the number of data packets waiting in the queue of each node can be reduced rapidly thanks to the features of applied EDCA. Thus, the probability of collision between nodes is reduced, which further improves the performance.

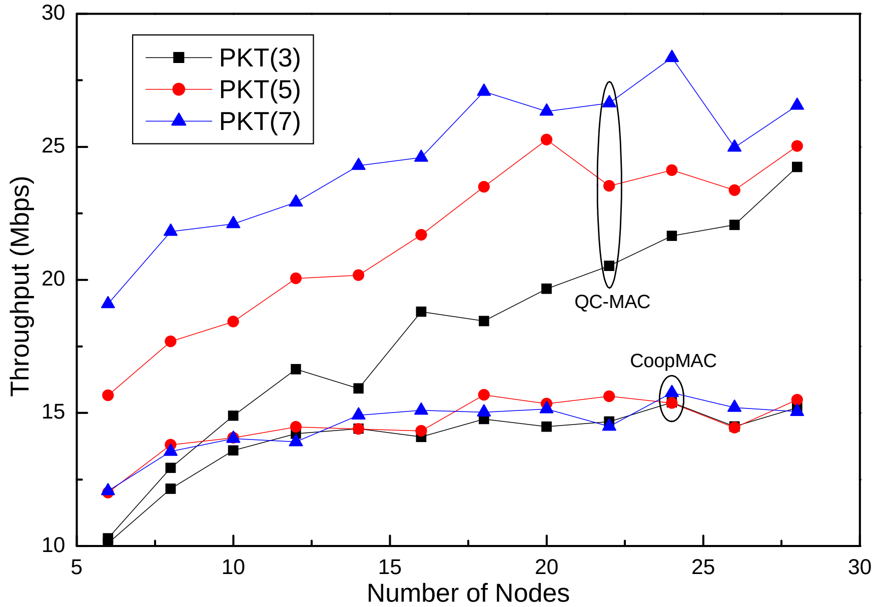


Fig. 15. Throughput according to the number nodes

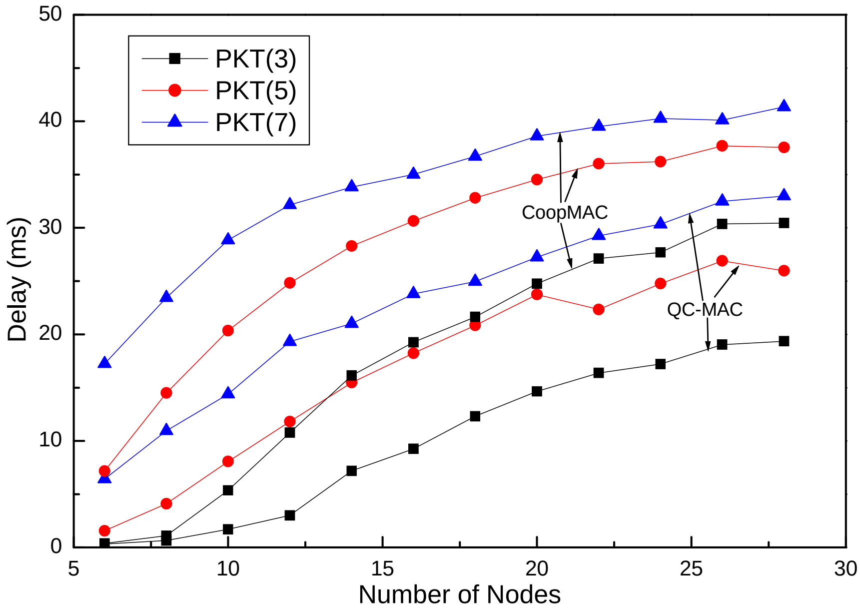


Fig. 16. Delay according to the number nodes

Fig. 16 shows delay according to the number of nodes. In both the two protocols, the lower *n* is, the lower the delay time becomes. When *n* is large, the volume of data generated is also increased. As the waiting time in the queue is also increasing, the delay time is increasing. The figure shows that the QC-MAC protocol always has lower delay performance than the CoopMAC protocol. This is because TXOP and block ACK of EDCA are applied to the proposed protocol. That is, it is possible to send multiple data packets with just one backoff process, the delay time becomes low. However, the CoopMAC protocol only supports the transmission of one data packet through one backoff process, the delay time becomes longer.

V. CONCLUSION

The IEEE 802.11 & 11e standards provide multiple transmission rates, which can be changed dynamically according to the channel condition. When using multiple transmission rates, the capacity of wireless LAN improves, but the performance anomaly problem may occur. Cooperative communications were introduced to alleviate the performance anomaly problem with the help of relay nodes with higher transmission rates. None of previous cooperative communications protocols has focused on cooperative communications in the IEEE 802.11e EDCA. We proposed a novel cooperative MAC based on the IEEE 802.11e EDCA MAC protocol. The proposed protocol applies the TXOP and block ACK features of the EDCA for cooperative communications to improve network performance and to overcome overheads by reducing the number of control packets for multiple data transmissions. In the proposed protocol, a node can transmit multiple data packets consecutively until the duration of transmission exceeds the specific TXOP time period. By using the Block ACK procedure, a receiver acknowledges a block of received data packets. Simulation results show that the proposed protocol works well and improves network performance.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0025495).

REFERENCES

[1] Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11, Jun. 1999.

[2] Y. Xiao, “A simple and effective priority scheme for IEEE 802.11,” IEEE Communications Letters, vol. 7, no. 2, pp. 70-72, Feb. 2003.

[3] H. Zhai, X. Chen, Y. Fang, “How well can the IEEE 802.11 wireless LAN support quality of service?,” IEEE Transactions on Wireless Communications, vol. 4, no. 6, pp. 3084-3094, Nov. 2005.

[4] IEEE, Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment: medium access control (MAC) quality of service enhancements, IEEE Standard 802.11e, 2005.

[5] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-hop wireless networks,” in Proc. ACM MobiCom’01, pp. 236-251, Jul. 2001.

[6] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in Proc. IEEE INFOCOM’03, vol. 2, pp. 836-843, 2003.

[7] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, “CoopMAC: a cooperative MAC for wireless LANs,” IEEE Journal on Selected Areas in Communications, vol. 25, no. 2, pp. 340-354, Feb. 2007.

[8] T. Guo and R. Carrasco, “CRBAR: cooperative relay-based auto rate MAC for multi-rate wireless networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 12, pp. 5938-5947, Dec. 2009.

[9] H. Zhu and G. Cao, “On improving the performance of IEEE 802.11 with relay-enabled PCF,” Mobile Networks Applications, vol. 9, pp. 423-434, Aug. 2004.

[10] H. Zhu and G. Cao, “rDCF: a relay-enabled medium access control protocol for wireless ad hoc networks,” IEEE Transactions on Mobile Computing, vol. 5, pp. 1201-1214, Sep. 2006.

[11] S. Zou, B. Li, H. Wu, Q. Zhang, W. Zhu, and S. Cheng, “A relay aided media access (RAMA) protocol in multi-rate wireless networks,” IEEE Transactions on Vehicular Technology, vol. 55, pp. 1657-1667, Sep. 2006.

[12] L. Guo, X. Ding, H. Wang, Q. Li, S. Chen, and X. Zhang, “Cooperative relay service in a wireless LAN,” IEEE Journal on Selected Areas in Communications, vol. 25, no. 2, pp. 355-368, Feb. 2007.

[13] W.-S. Lim, D.-W. Kim, and Y.-J. Suh, “PR-MAC: a practical approach for exploiting relay transmissions in multi-rate WLANs,” IEEE Transactions on Wireless Communications, vol. 9, no. 1, pp. 66-71, Jan. 2010.

[14] B. Zhang, Z. Zheng, X. Jia, and K. Yang, “A distributed collaborative relay protocol for multi-hop WLAN accesses,” in Proc. IEEE GLOBECOM 2010, pp. 1-5, Dec. 2010.

[15] B. Zhang and X. Jia, “Multi-hop collaborative relay networks with consideration of contention overhead of relay nodes in IEEE 802.11 DCF,” IEEE Transactions on Communications, vol. 61, no. 2, pp. 532-540, Feb. 2013.

[16] J. Liu, W. Wang, Z. Zheng, X. Zhang, C. Chen, and X. Shen, “Lifetime extended cooperative MAC protocol for wireless LANs,” in Proc. IEEE GLOBECOM 2012, pp. 5476-5481, Dec. 2012.

[17] S. Nischal and V. Sharma, “A cooperative ARQ scheme for infrastructure WLANs,” in Proc. IEEE WCNC 2013, pp. 428-433, Apr. 2013.

[18] X. Zhang, L. Guo, and X. Wei, “An energy-balanced cooperative MAC protocol based on opportunistic relaying in MANETs,” Computers & Electrical Engineering, vol. 39, no. 6, pp. 1894-1904, Aug. 2013.

[19] R. Sadeghi, J. P. Barraca, and R. L. Aguiar, “Cooperative performance bounds of Wireless Local Area Networks,” in Proc. IEEE WPMC 2011, pp. 1-5, Oct. 2011.

AUTHOR PROFILE

Sunmyeng Kim received the B.S., M.S., and Ph.D. degrees in information and communication from Ajou University, Suwon, Korea, 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. In March 2008, he then joined the Department of Computer Software Engineering, Kumoh National Institute of Technology, Gumi, Korea, as an associate professor. His research interests include resource management, wireless LANs and PANs, wireless mesh networks, UWSNs and quality of service enhancement.