Wireless Netw (2016) 22:2727–2739
DOI 10.1007/s11276-015-1121-y

RM-MAC: a reservation based multi-channel MAC protocol for wireless sensor networks

Dohoo Pyeon[1] · Ingook Jang[1] · Hyunsoo Yoon[1] · Dongwook Kim[2]

Published online: 21 November 2015

Abstract Convergecast is a general communication pattern of wireless sensor networks (WSNs) in which sensed data is collected from outlying sensor nodes and is transmitted to a sink node. In this paper, we consider periodic convergecast, in which data packets are generated and transmitted repeatedly with a certain interval. To support the periodical packet delivery efficiently, we propose a reservation based multi-channel MAC protocol (RM-MAC) which employs a time reservation mechanism. Periodical packet delivery allows receiver nodes to recognize when the next packet is transmitted, which makes the nodes reserve the time for the next packet transmission. Consequently, our proposed mechanism can guarantee minimum delivery latency from source nodes to a sink node. Furthermore, RM-MAC provides a collision resolution mechanism which coordinates the reserved periodical transmissions among multiple sender nodes. It can prevent repeated collisions caused by the *multiple-senders-single-receiver problem*. We implement RM-MAC by using an *ns-2* simulator to compare its performance with other protocols. The simulation results show that RM-MAC outperforms existing protocols in terms of energy efficiency and packet delivery latency.

Keywords Wireless sensor networks · Multi-channel MAC protocols · Duty cycling technique · Periodic convergecast

1 Introduction

In traditional wireless sensor networks (WSNs), nodes usually operate on single-channel radio interface since it is enough to report the sensed information with low data rates. As, recently, high data rate applications such as multimedia data based environment monitoring and real-time critical event detection have emerged in WSNs, conventional single-channel communications are exposed to not only a large number of collisions due to simultaneous sending among multiple nodes but also serious interference due to the transmissions of adjacent nodes, with just one shared wireless medium [1]. On the other hand, in multi-channel communications, adjacent nodes can simultaneously transmit/receive packets through different wireless channels which helps to reduce collisions and interference. Therefore, multi-channel communications have become promising for the high data rate applications in WSNs.

A number of multi-channel MAC protocols have been proposed in the previous literature. Each node in [2] and [3] collects the channel assignment information of its neighbor nodes to avoid potential collisions, and nodes in [4–6] switch their channels for every transmission for channel diversity. They show better network performance

✉ Dongwook Kim
kimdw@nslab.kaist.ac.kr

Dohoo Pyeon
dhpyeon@nslab.kaist.ac.kr

Ingook Jang
ikjang@nslab.kaist.ac.kr

Hyunsoo Yoon
hyoon@kaist.ac.kr

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, Korea

[2] Network Business Division, Samsung Electronics, 416, Maetan 3-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-742, Korea

compared to single-channel MAC protocols in terms of throughput, delivery ratio, and delivery latency. Furthermore, they improve energy efficiency by reducing the number of collisions and interference under high traffic load.

When these multi-channel MAC protocols are applied in duty-cycled WSNs, they show some inherent limitations in gathering sensed data. Duty cycling is an essential mechanism for energy efficiency in WSNs [7], which allows nodes to alternate between active and sleep states. As an inherent weakness, it increases transmission delay because a sender cannot deliver a data packet until the receiver wakes up from a sleep state [8]. In addition, if multiple senders try to transmit their data packets to a common intended receiver simultaneously when the receiver wakes up, a collision may occur on the receiver. We refer to this collision as a *multiple-senders-single-receiver problem*. This problem leads to delivery latency and energy dissipation by retransmissions. Typically, in WSNs, sensed data is transmitted from many sensor nodes to a sink node as *many-to-one* and *multi-hop* communications as shown in Fig. 1(a) [9, 10]. Thus, end-to-end packet delivery latency can be increased by the transmission delay accumulated with every hop, and collisions caused by the multiple-senders-single-receiver problem occur repeatedly. Therefore, to resolve these problems, it is necessary to design an efficient multi-channel MAC protocol for data gathering.

Periodic convergecast is a typical communication pattern to gather sensed data in a sink node, and data packets periodically are generated and transmitted with a certain interval [9]. Representative applications of periodic convergecast are environmental or habitat monitoring [11] in which all nodes report sensed data while they are alive, mobile target tracking [12] in which sensor nodes transmit the location of a target while it is within their detection ranges, and forest fire detection [13] to monitor environmental values, such as temperature or smoke that exceed a given threshold. We name a set of transmissions from a source node to a sink node as a *periodic data stream*, which is represented as $\langle T_1, T_2, \ldots, T_n \rangle$ as shown in Fig. 1(b). In the periodic data stream, data packets are periodically generated and transmitted for a certain time [14], so each receiver can easily predict when the next data packet in the periodic data stream will be transmitted by a sender. If the receiver wakes up at the predicted transmission time, the next data packet will be immediately transmitted without any waiting by the sender. Furthermore, the receiver can coordinate the transmission time of multiple senders to avoid the collisions caused by the multiple-senders-single-receiver problem.

In this paper, we propose a reservation based multi-channel MAC protocol (RM-MAC) to support periodic convergecast efficiently. The proposed RM-MAC employs a *reservation mechanism* to predict the traffic pattern of periodic convergecast. The reservation mechanism helps RM-MAC to achieve much higher energy efficiency and lower packet delivery latency than conventional protocols regardless of heavy traffic. The reservation mechanism in RM-MAC is similar to that of MRMAC [14], which has been proposed for event-driven applications. However, MRMAC operates on single-channel communications, so it is hard to support high data rate applications, especially in heavy traffic. In addition, the reservation mechanism in MRMAC results in considerable communication overhead. On the other hand, the proposed RM-MAC operates on multi-channel communications, and we introduce a method to reduce the communication overhead. Although RM-MAC uses a reservation scheme which is similar to MRMAC, RM-MAC reduces communication overhead and improves network performance with more energy efficiency compared to MRMAC.

The remainder of this paper is organized as follows. Existing multi-channel MAC protocols for WSNs are discussed in Sect. 2, and we describe the details of RM-MAC in Sect. 3, we present the simulation results and compare

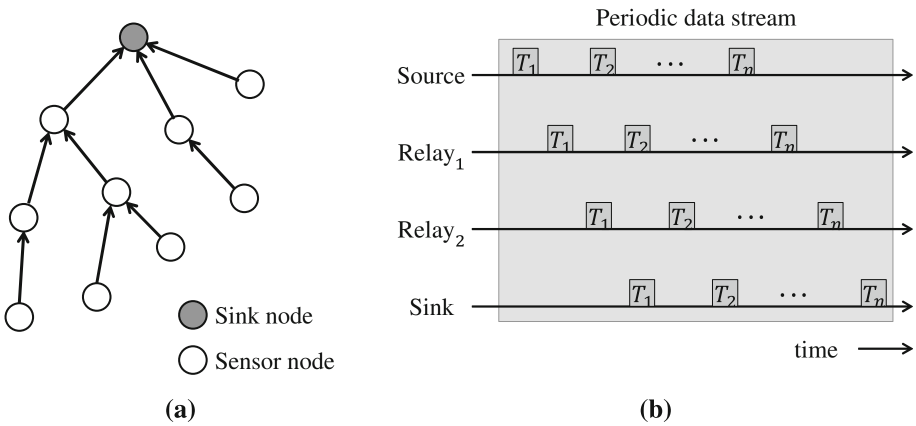


Fig. 1 **a** Data gathering communications in WSNs; **b** periodic data stream in periodic convergecast

the performance through the *ns-2* simulator in Sect. 4, and we conclude this paper in Sect. 5.

2 Related work

Multi-channel MAC protocols in WSNs should arrange a rendezvous channel and time for communications. The rendezvous channel can be chosen by fixed, semi-dynamic, and dynamic channel assignment schemes [1]. The fixed and semi-dynamic approaches assign a static channel to a node for communications. Each node in the fixed approach permanently chooses a channel, and the semi-dynamic channel assignment allows each node to select a constant channel which may be changed. On the other hands, in the dynamic approach, each node changes its channel before each wake-up or transmission. The multi-channel MAC protocols are categorized as synchronous and asynchronous approaches according to arranging rendezvous time. Synchronous protocols use time synchronization where all nodes simultaneously wake up for communications. Contrastively, each node in asynchronous protocols independently wakes up following its own schedule, so a sender should know wake-up time of a receiver to transmit data.

MMSN [2] and MC-LMAC [3] are one of the synchronous protocols based on the semi-dynamic channel assignment. Each node in these protocols collects 2-hop neighbor nodes' information to choose a different channel from the neighbor nodes' channel. It helps adjacent nodes to avoid potential collisions. However, collecting the information leads to additional overhead and complexity due to exchange of control messages.

Synchronous approaches with the dynamic channel assignment such as Y-MAC [5] and MuChMAC [4] have also been proposed. Each node in Y-MAC switches its channel by using its hopping channel sequence which is predefined and is known to neighbor nodes. When a sender intends to transmit a data packet to a receiver, the sender wakes up on the first channel of the receiver's channel sequence. After the transmission, both of the sender and the receiver switch to the next channel of the receiver's channel sequence if the sender wants to transmit more data packets. This scheme helps to improve throughput in high traffic loads, but Y-MAC requires tight time synchronization with more complexity and overhead. Unlike Y-MAC, each node in MuChMAC chooses a random channel in every time-slot by using a pseudo random generator, and MuChMAC adopts a short preamble scheme used in X-MAC [15]. The short preamble scheme helps MuChMAC to reduce the overhead of the tight time synchronization. However, these two protocols do not consider channel conditions, so nodes may attempt to transmit/receive packets on channels with bad conditions.

SA-MAC [16] is an asynchronous protocol based on the semi-dynamic channel assignment. SA-MAC adopts a preamble-based approach where nodes utilize a microframe preamble (*MFP*). A sender transmits MFPs on its channel before transmitting a data packet. A receiver wakes up and scans all channels for receiving a MFP from potential senders. The receiver transmits an acknowledgement packet (ACK) to the sender after receiving a MFP, and then they transmit/receive data. SA-MAC reduces potential interference and effectively handles network topology changes such as joining new nodes. However, scanning all channels on every wake-up and transmitting MFPs become large overhead. In addition, the preamble-based approach typically consumes more energy than a receiver-initiated approach [17].

In the receiver-initiated approach, a receiver transmits a wake-up beacon message to potential senders when the receiver wakes up. A sender who wants to transmit a data packet to the receiver transmits the data packet after receiving the wake-up beacon. The receiver transmits an ACK beacon to the sender after receiving the data packet. Recently, EM-MAC [6] have been proposed based on the receiver-initiated approach and the dynamic channel assignment. Each node in EM-MAC wakes up by using its own schedule which consists of the next wake-up time and channel. The schedule is generated by a common pseudo random function, and a receiver informs a sender of parameters of the function by using an ACK beacon. Consequently, the sender can predict the future wake-up schedules of the receiver. To avoid transmissions on channels with bad conditions, each node maintains a *blacklist* which includes IDs of bad channels. Nodes try to avoid switching to the bad channels in order to reduce transmission failure. EM-MAC achieves lower duty cycle, delivery latency, and higher packet delivery ratio under wireless interference and jamming.

As we mentioned in Sect. 1, the above MAC protocols commonly have two problems; long delivery latency and a number of collisions on cross nodes. MRMAC [14] have been proposed to resolve these problems, but it is a single-channel protocol and only a conceptual ideal without any details of implementations. Therefore, the proposed RM-MAC provides the reservation mechanism which is able to improve network performances and the feasibility.

3 RM-MAC

In this section, we describe the detailed design of RM-MAC. RM-MAC is based on the receiver-initiated approach. Each node in RM-MAC wakes up according to both a *wake-up schedule* and a *reservation schedule*. The wake-up schedule is the set of the wake-up channel and time for receiving data packets from potential senders. Each node determines its future wake-up schedules by using a common pseudo random generator with its node ID. Since a node can obtain the ID of a neighbor node through neighbor discovery mechanisms such as in [18, 19], and [20], the node can know the wake-up schedules of the neighbor node. On the other hand, the reservation schedule consists of a reservation channel and time which are made by a pair of a sender and a receiver when they transmit/receive a periodic data stream. When each data packet of the periodic data stream is transmitted, the sender and the receiver make a reservation schedule for the next data packet transmission. They wake up on the reservation schedule, and then the sender transmits the next data packet to the receiver immediately.

3.1 Reservation mechanism

A sender and a receiver use a *reservation mechanism* with the symbols in Table 1 to make a reservation schedule. The reservation mechanism comprises three steps: (i) a *reservation proposal*, (ii) a *reservation decision*, and (iii) *reservation notification*. In the first step, the sender transmits a *proposal* $\mathcal{P}$ with a data packet, which consists of a channel and time. The sender also transmits its *list of reservation schedules* $\mathcal{L}$ to avoid overlapping reservations. After receiving them, the receiver decides a channel and time of a *decision* $\mathcal{D}$ in the reservation decision step. $\mathcal{D}$ is transmitted with an ACK beacon, and then the sender and the receiver add it to their $\mathcal{L}$ in the reservation notification step. They use the *RM-algorithm* to determine $\mathcal{P}$ and $\mathcal{D}$, which is described in the Sect. 3.2.

If a collision occurs in the reservation mechanism, the data packet is retransmitted by using a binary exponential back-off in the receiver-initiated approach [21]. When a receiver detects a collision by using the start of frame delimiter (*SFD*) and the clear channel assessment (*CCA*), the receiver transmits a wake-up beacon including a back-off window size. Then, a sender chooses a random back-off time and retransmits the data packet after the random back-off time. The retransmission is repeated within the limited number of retransmissions until the data packet is successfully transmitted. The limited number of retransmissions is determined according to a channel condition (described in Sect. 3.4), and the back-off window size is doubled in every retransmission. The sender also uses a random back-off time to retransmit the data packet when an ACK beacon collision occurs. In this case, the sender tries the retransmission only once by using a small value as the back-off window size. Then, if the sender cannot receive the ACK beacon in the retransmission, the sender goes to sleep and then wakes up on the earliest wake-up schedule of the receiver to retransmit the data packet.

Table 1 Symbols for RM-MAC

Symbol	Description
$\mathcal{P}$	A proposal from a sender, $(ch^{\mathcal{P}}, t^{\mathcal{P}})$
$\mathcal{D}$	A decision by a receiver, $(ch^{\mathcal{D}}, t^{\mathcal{D}})$
$\mathcal{L}_i$	A list of reservation schedules of node i
$\mathcal{R}_i^k$	The kth entry of $\mathcal{L}_i$
$\mathcal{T}$	The time for a reserved transmission

Figure 2 shows the reservation mechanism in a simple network with two channels where a source node S transmits a periodic data stream to a destination node D through relay nodes N_1 and N_2. In this example, we assume that four nodes have no reservations and that no collision occurs. When the first data packet is generated, node S wakes up on a wake-up schedule of node N_1 and then transmits $(ch^{\mathcal{P}}, t^{\mathcal{P}})$ and $\mathcal{L}_S$ with the first data packet to node N_1. Node N_1 decides and transmits $(ch^{\mathcal{D}}, t^{\mathcal{D}})$ with an ACK beacon to node S. Then, the decision is added to $\mathcal{L}_S$ and $\mathcal{L}_{N_1}$, respectively. In this example, $ch^{\mathcal{D}}$ and $t^{\mathcal{D}}$ are decided to Ch_1 and t_2, so nodes S and N_1 simultaneously wake up on Ch_1 at t_2 to transmit/receive the second data packet immediately.

The reservation mechanisms from node N_1 to node D are similar to that between nodes S and N_1. The second data packet transmissions between nodes N_1 and N_2 and nodes N_2 and D are reserved to $(Ch_1, t_2 + \mathcal{T})$ and $(Ch_1, t_2 + 2\mathcal{T})$, respectively. Here, $\mathcal{T}$ is the time for each reserved transmission, which is determined by the limited number of retransmissions. Consequently, the second data packet is transmitted from node S to node D with reduced delivery latency.

The third data packet is reserved through the second data packet transmission, and future-generated data packets are also gradually reserved. As a result, each data packet in the periodic data stream is transmitted from node S to node D within $3\mathcal{T}$. If there are n relay nodes between nodes S and D, every future data packet can be transmitted within $(n + 1)\mathcal{T}$.

3.2 RM-algorithm

A channel and time of a reservation schedule are determined by the *RM-algorithm*. First, in a periodic data stream, all nodes along a route use the same channel to reduce the cost of channel switching. A source node chooses Ch_i when transmitting the first data packet of the periodic data stream, and Ch_i is used for all reservations of the periodic data stream. Each relay node assigns Ch_i to $ch^{\mathcal{P}}$ when it makes $\mathcal{P}$, and each receiver decides Ch_i for

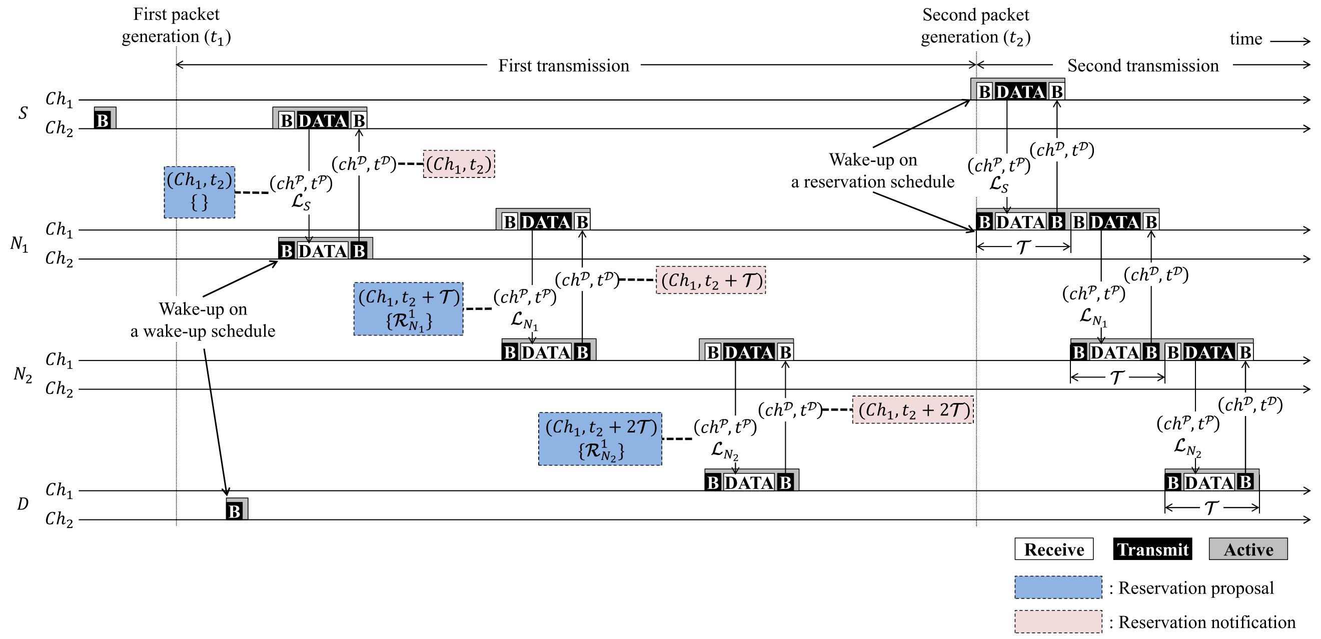

First packet generation (t_1)
Second packet generation (t_2)
time
First transmission
Second transmission
S
N1
N2
D
Ch1
Ch2
B
DATA
(Ch_1, t_2) { }
($ch^{\mathcal{P}}, t^{\mathcal{P}}$)
$\mathcal{L}_S$
($ch^{\mathcal{D}}, t^{\mathcal{D}}$)
(Ch_1, t_2)
Wake-up on a reservation schedule
Wake-up on a wake-up schedule
($Ch_1, t_2+\mathcal{T}$) {$\mathcal{R}^1_{N_1}$}
$\mathcal{L}_{N_1}$
($Ch_1, t_2+\mathcal{T}$)
($Ch_1, t_2+2\mathcal{T}$) {$\mathcal{R}^1_{N_2}$}
$\mathcal{L}_{N_2}$
($Ch_1, t_2+2\mathcal{T}$)
$\mathcal{T}$
Receive
Transmit
Active
: Reservation proposal
: Reservation notification

Fig. 2 The reservation mechanism of RM-MAC

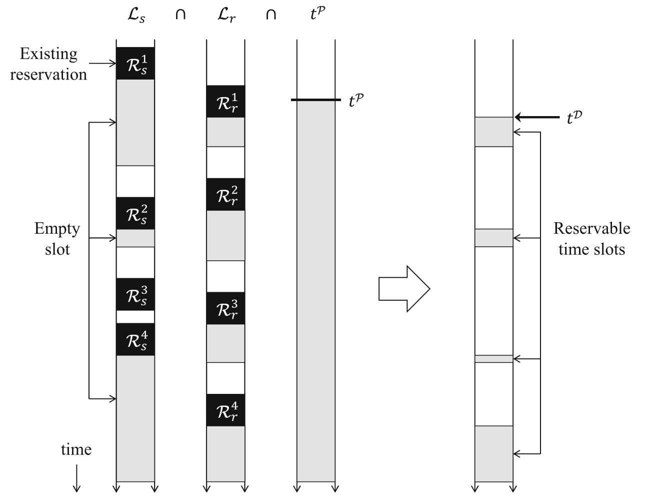


Fig. 3 The RM-algorithm for $t^{\mathcal{D}}$

$ch^{\mathcal{D}}$. For example, as shown in Fig. 2, a source node S chooses Ch_1 for $ch^{\mathcal{P}}$, and relay nodes N_1 and N_2 also assign Ch_1 to $ch^{\mathcal{P}}$. Nodes N_1, N_2, and D choose Ch_1 for $ch^{\mathcal{D}}$ when they make the decisions. Consequently, the second data packet is transmitted on Ch_1 from node S to node D without channel switching. The reservation channel can be changed when the channel has a bad condition (described in Sect. 3.4).

The purpose of determining the reservation time is to reduce delivery latency and to avoid overlapping reservations. A sender finds the earliest time to determine $t^{\mathcal{P}}$ and then transmits $t^{\mathcal{P}}$ and $\mathcal{L}_s$ to a receiver. The receiver makes three sets by using $t^{\mathcal{P}}$, $\mathcal{L}_s$, and its list $\mathcal{L}_r$ as shown in Fig. 3. The receiver finds the *reservable time slots*, and then the earliest time of the reservable time slots is chosen for $t^{\mathcal{D}}$. As the result, the decided reservation time is the earliest time without overlapping reservations.

3.3 Operation of RM-MAC in a cross topology

In a cross topology, the reservation mechanism helps to mitigate the multiple-senders-single-receiver problem. Figure 4 shows how to operate on a simple cross network with a cross node R and two senders S_1 and S_2. We assume that nodes S_1 and S_2 simultaneously generate periodic data streams with the same time interval. The three nodes have no reservations, so their $\mathcal{L}$ are empty.

Nodes S_1 and S_2 simultaneously transmit the first data packets after receiving a wake-up beacon from node R. The first data packets collide on node R, and node R transmits the wake-up beacon, which includes a back-off window size. Then, nodes S_1 and S_2 retransmit the first data packets after the random back-off time. In this example, we assume that R first receives the data packet from node S_2. Nodes R and S_2 reserve the second data packet transmission to (Ch_1, t_2). After that, node R receives the first data packet from node S_1 with the proposal (Ch_2, t_2). However, node R cannot reserve the second data packet transmission at t_2 because it already has a reservation schedule at that time. To avoid overlapping reservations, node R chooses $(t_2 + \mathcal{T})$ as the reservation time of node S_1, which is the end of

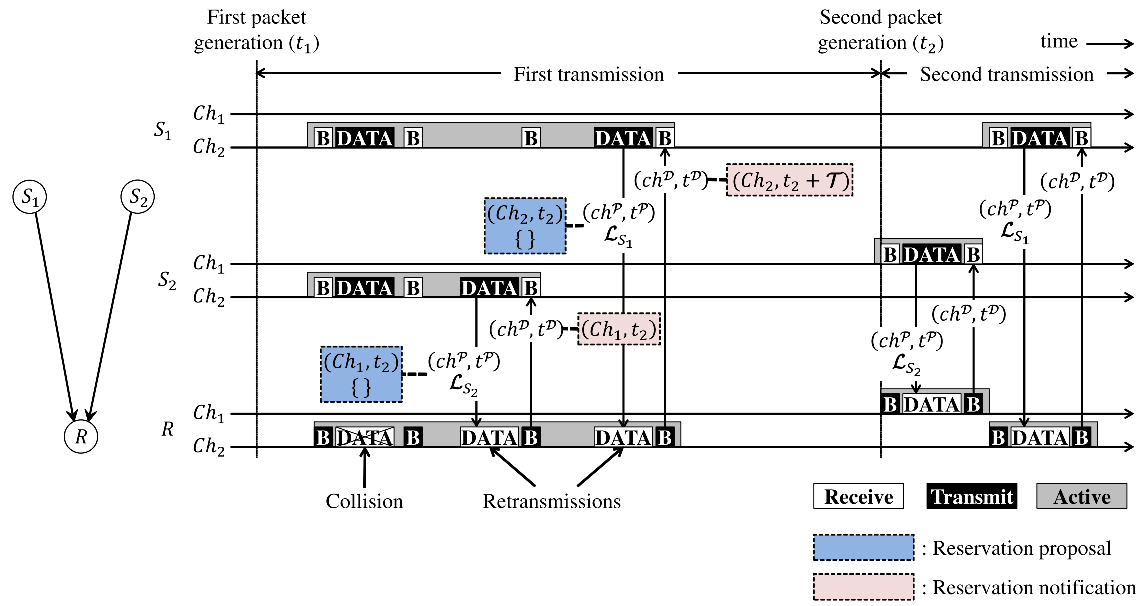


Fig. 4 The operation of RM-MAC in the cross topology

the reservation schedule of node S_2. Thus, node S_1 wakes up at $(t_2 + \mathcal{T})$ although the second data packet is generated at t_2. Node R also wakes up at $(t_2 + \mathcal{T})$ to receive the second data packet from node S_1 after the second transmission of node S_2. Therefore, the cross node R receives the second data packets without a collision. All data packets in the periodic data streams can be also transmitted without any collisions by the same token. In cross topologies with m senders, only $(m - 1)$ or fewer collisions will occur for each periodic data stream. Consequently, the multiple-senders-single-receiver problem is mitigated by the reservation mechanism.

3.4 Consideration of channel conditions

RM-MAC considers channel conditions to avoid reserving bad channels because transmissions on the bad channels cause a number of retransmission with additional energy dissipation [22]. Each node maintains a *channel state value (CSV)* for each channel, which is the transmission success rate. The *CSV* is statistically calculated by an exponential moving average in every transmission as

$$CSV_n = p \cdot S_n + (1 - p) \cdot CSV_{n-1} \tag{1}$$

where S_n is the success or failure of the nth transmission (0 or 1), and p is a weight value. If a channel has a *CSV* that is lower than the threshold CSV_{thresh}, the channel is labeled as a bad channel, and the channel ID is added to the *unusable channel list (UCL)*. The bad channel ID is removed from *UCL* after a certain time to prevent all channels being included in the *UCL*.

A sender and a receiver use their *UCL* to avoid a reservation channel with a bad condition. If the sender proposes Ch_i which is in the receiver's *UCL*, the receiver chooses the best channel Ch_j among channels which are not in the *UCL* of both the sender and the receiver. For this mechanism, the sender has to transmit its *UCL* with $\mathcal{P}$. If there is no available channel for the reservation, the receiver randomly chooses a channel which is not in the receiver's *UCL*. Therefore, the sender can transmit the next data packet to the receiver on Ch_j (or the randomly chosen channel), not on Ch_i.

When the receiver transmits the data packet to the next hop node, the receiver assigns Ch_j (or the randomly chosen channel) to $ch^{\mathcal{P}}$. The rest of the relay nodes from the receiver to a destination node also assign Ch_j to $ch^{\mathcal{P}}$ if Ch_j is not in their *UCL*. Future generated data packets are also reserved on Ch_j, so the nodes can avoid transmitting the periodic data stream on a bad channel.

CSV is also used to determine the time for a reserved transmission $\mathcal{T}$. If $\mathcal{T}$ becomes larger, a sender can retransmit a data packet when collisions occur during the reservation mechanism, but the delivery latency may increase. Otherwise, the delivery latency decreases, but the dropped data packet cannot be retransmitted in the reservation. To balance the trade-off between delivery latency and retransmissions, the receiver flexibly adjusts $\mathcal{T}$ by using *CSV*. We define k as the number of retransmissions for achieving the target packet delivery ratio α:

$$(1 - CSV)^{k+1} \leq (1 - \alpha) \tag{2}$$

The receiver finds the minimum value of k satisfying Eq. (2), which is used for the limited number of retransmissions l. Then, the receiver calculates $\mathcal{T}$ by using the limited number l as follows:

$$\mathcal{T} = t_{tran} + \sum_{i=1}^{l} (t_{back}^{i} + t_{tran}) \qquad (3)$$

where t_{tran} is the total time to transmit a wake-up beacon, a data packet, and an ACK beacon, and t_{back}^{i} is the maximum back-off time of the ith retransmission in the receiver-initiated approach [21]. It helps to efficiently handle packet collisions and packet drop in the reservation mechanism.

If the sender cannot transmit a data packet successfully within $\mathcal{T}$ due to frequent collisions, the sender goes to sleep and then wakes up on the earliest wake-up schedule of the receiver to transmit the data packet again. We refer to this retry transmission as an *additional retransmission*. The sender also does the additional retransmission when the sender does not receive the wake-up beacon from the receiver. After two more failed additional retransmissions, the sender considers the receiver as a failure node and then notifies it to an upper layer protocol to find a new route.

3.5 Analysis of communication overhead

Each node in RM-MAC transmits the information such as $\mathcal{P}$, $\mathcal{L}$, $\mathcal{D}$, and *UCL* by using a *RM-MAC header*. Figure 5a, b show the formats of the RM-MAC headers in a data packet and an ACK beacon, respectively. The RM-MAC headers commonly contain a frame control field (*FCF*), a source address field (*Src*), a destination address field (*Dst*), and a frame check sequence (*FCS*). In the data packet, the RM-MAC header includes $\mathcal{P}$ and $\mathcal{L}_s$ with n reservations. A *Res* field in $\mathcal{P}$ is reserved for future use. Each entry in $\mathcal{L}_s$ consists of a reservation ID, reservation time, and a k value. The k value indicates the number of retransmissions in Eq. (2), which is used to calculate $\mathcal{T}$ of each reservation. A sender includes its *UCL* in the data packet when its *UCL* is changed during each periodic data stream transmission. The number of channels in the CC2420 radio is 16 [23], so the *UCL* is encoded in 2 bytes and the size of $ch^{\mathcal{P}}$ is 4 bits. A *BW* field in the ACK beacon is used for the back-off window size if a receiver has detected collisions when receiving a data packet. A $k^{\mathcal{D}}$ field in $\mathcal{D}$ is used to inform the sender of $\mathcal{T}$.

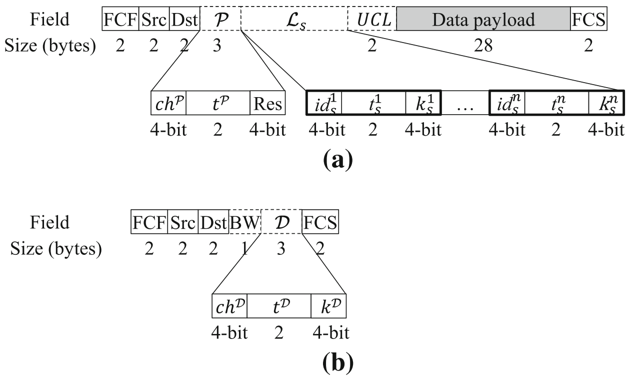


Fig. 5 **a** RM-MAC header in the data packet; **b** RM-MAC header in the ACK beacon

$\mathcal{P}$, $\mathcal{D}$, and $\mathcal{L}_s$ do not need to be transmitted with every transmission because they are predictable. First, if a receiver has $\mathcal{P}$ of the previous data packet transmission, the receiver can predict current $\mathcal{P}$ when deciding $\mathcal{D}$. However, $\mathcal{P}$ that is actually transmitted from the sender may differ from the predicted $\mathcal{P}$ in some cases such as when $ch^{\mathcal{P}}$ should be changed by a bad condition or when $t^{\mathcal{P}}$ is moved forward or backward by other reservations. In these cases, the sender should transmit $\mathcal{P}$ because the receiver should use the transmitted $\mathcal{P}$. On the contrary, if the receiver can uses the predicted $\mathcal{P}$, the sender does not need to transmit $\mathcal{P}$.

Similarly, $\mathcal{D}$ and $\mathcal{L}_s$ can also be predicted by those of the previous data packet transmission. Thus, the receiver transmits $\mathcal{D}$ only when the predicted $\mathcal{D}$ is different from the $\mathcal{D}$ chosen by the receiver, and the sender transmits only some entries of $\mathcal{L}_s$ when some reservations are newly made or removed. Although the sender and the receiver do not receive $\mathcal{D}$ and $\mathcal{L}_s$, they can make a reservation schedule by using the predicted $\mathcal{D}$ and $\mathcal{L}_s$.

At this point, we can analyze the size of $\mathcal{P}$, $\mathcal{L}_s$, and $\mathcal{D}$. This analysis is simplified by two assumptions: (i) channel conditions are not changed and (ii) $\mathcal{P}$ and $\mathcal{L}_s$ are transmitted together. This analysis provides the expected size of the piggybacked information in each periodic data stream.

Lemma 1 *Suppose that k periodic data streams are generated in time t which is the time duration of each periodic data stream, and k follows a Poisson distribution* $P(k) = \frac{\lambda^k e^{-\lambda}}{k!}$ *with a mean rate* $\lambda = \mathcal{M}$. *Then, each node has an average of* $2\mathcal{M}$ *reservations.*

Proof Assume that K periodic data streams are generated for a long time T. Then, the average number of periodic data streams that are being transmitted simultaneously $E(N)$ is calculated as the sum of the time durations of the K periodic data streams divided by the time T as follows:

$$\begin{aligned} E(N) &= E\left(\frac{\sum_{i=1}^{K} t}{T}\right) = E\left(\frac{t \times K}{T}\right) \\ &= \frac{t \times E(K)}{T} = \frac{t \times \left(\frac{\mathcal{M}}{t} \times T\right)}{T} = \mathcal{M} \end{aligned} \qquad (4)$$

Each node uses only one reservation when transmitting a periodic data stream because the next data packet transmission is reserved when a current data packet transmission is finished. Similarly, the node also uses a reservation to receive the periodic data stream. Therefore, each node (except source and destination nodes) has $2\mathcal{M}$ reservations for receiving/transmitting the $\mathcal{M}$ periodic data streams. □

Lemma 2 *When a sender transmits the first data packet of a periodic data stream, the size of $\mathcal{P}$ and $\mathcal{L}_s$ that are included in the first data packet is $(6\mathcal{M}+3)$ bytes.*

Proof When the first data packet is transmitted, $\mathcal{L}_s$ includes all reservations of the sender. Since the size of each entry of $\mathcal{L}_s$ is 3 bytes and the sender has $2\mathcal{M}$ reservations by Lemma 1, the size of $\mathcal{L}_s$ is $6\mathcal{M}$ bytes. The size of $\mathcal{P}$ is 3 bytes, so the total size of $\mathcal{P}$ and $\mathcal{L}_s$ in the first data packet is $(6\mathcal{M}+3)$ bytes. □

Lemma 3 *The total size of $\mathcal{P}$ and $\mathcal{L}_s$ included in other data packets except the first data packet is $18\mathcal{M}$ bytes.*

Proof The sender should transmit $\mathcal{P}$ and $\mathcal{L}_s$ with the data packets when new reservations are made by new periodic data streams or when existing reservations are removed by finishing periodic data streams. Two reservations are made or removed for each periodic data stream, so the size of $\mathcal{P}$ and $\mathcal{L}_s$ included in each data packet is (3+6) bytes. Since the number of the new periodic data streams generated in T is $\mathcal{M}$ by the Poisson distribution, their total size transmitted by the new reservations is $9\mathcal{M}$ bytes. Similarly, $\mathcal{M}$ existing periodic data streams are also finished in T, so the size of $\mathcal{P}$ and $\mathcal{L}_s$ transmitted by the removed reservations is $9\mathcal{M}$ bytes. Therefore, the total size of $\mathcal{P}$ and $\mathcal{L}_s$ included in the data packets is $18\mathcal{M}$ bytes. □

Lemma 4 *The total size of $\mathcal{D}$ transmitted in a periodic data stream is $(6\mathcal{M}+3)$ bytes.*

Proof The receiver should transmit $\mathcal{D}$ in the first data packet transmission of the periodic data stream. In addition, $\mathcal{D}$ will also be transmitted when other periodic data streams are generated or finished. Since the total number of periodic data streams generated or finished in T is $2\mathcal{M}$ by the Poisson distribution, $\mathcal{D}$ is transmitted $(2\mathcal{M}+1)$ times in each periodic data stream. The size of $\mathcal{D}$ is fixed at 3 bytes, so the total size of $\mathcal{D}$ transmitted in the periodic data stream is $3\cdot(2\mathcal{M}+1)$ bytes. □

Lemma 5 *The expected size of all information transmitted in a periodic data stream is $(30\mathcal{M}+6)$ bytes.*

Proof According to Lemmas 2 and 3, the size of $\mathcal{P}$ and $\mathcal{L}_s$ transmitted in a periodic data stream is $(24\mathcal{M}+3)$ bytes. The size of $\mathcal{D}$ transmitted in a periodic data stream is $(6\mathcal{M}+3)$ bytes by Lemma 4. Therefore, their total size is $(30\mathcal{M}+6)$ bytes. □

We also analyze the reservation mechanism of MRMAC to compare the communication overhead of RM-MAC with that of MRMAC. As in RM-MAC, a data packet in MRMAC also includes a list of reservations and the next data packet information called medium reservation information *(MRI)* and next data packet arrival time *(NPAT)*, respectively. In contrast with RM-MAC, MRI and NPAT are composed of only time values due to single-channel communications, and their size is fixed. Then, MRI and NPAT are transmitted with every data packet. The following Lemma shows the expected size of MRI and NPAT in MRMAC.

Lemma 6 *In MRMAC, the size of MRI and NPAT transmitted in a periodic data stream is $n(2\mathcal{S}+2)$ bytes, where $\mathcal{S}$ is the maximum number of reservations that each node can have, and n is the number of data packets generated in each periodic data stream.*

Proof In the reservation mechanism of MRMAC, a sender has to transmit its all reservations by using the fixed-sized MRI, so the number of entries of MRI should be equal to the maximum number of reservations $\mathcal{S}$. Then, each data packet includes $(\mathcal{S}+1)$ time values, $\mathcal{S}$ time values for MRI and a time value for NPAT. If MRMAC uses a two-byte floating format for the time values, the size of MRI and NPAT in each data packet is $(2\mathcal{S}+2)$ bytes. Since n data packets are transmitted in a periodic data stream, the total size of MRI and NPAT transmitted in a periodic data stream is $n(2\mathcal{S}+2)$ bytes. □

Through Lemmas 5 and 6, we can estimate and compare the upper bounds of the communication overhead of RM-MAC and MRMAC. For the estimation, we suppose that $\mathcal{M}$ satisfies the following condition because two reservations are required for each periodic data stream:

$$2\mathcal{M}\leq\mathcal{S} \tag{5}$$

Then, the upper bound of the communication overhead of RM-MAC is obtained as follows:

$$(30\mathcal{M}+6)\leq(15\mathcal{S}+6)=\mathcal{O}(S) \tag{6}$$

The upper bound of the communication overhead of MRMAC can be easily estimated as $\mathcal{O}(n\cdot\mathcal{S})$. At this point, we can consider $\mathcal{S}$ as a constant because the number of entries of MRI is limited doe to the maximum packet size of CC2420 [23]. Therefore, the upper bound of RM-MAC is bounded by a constant, and the upper bound of MRMAC is bounded by $\mathcal{O}(n)$. This estimation indicates that RM-MAC can transmit periodic data streams with a constant communication overhead regardless of high data rate environments, whereas the communication overhead of MRMAC increases as the data rate increases.

4 Performance evaluation

To verify the performance, we implemented RM-MAC by using an *ns-2* simulator. We used the radio transceiver parameters in the MICAz datasheet [24] as shown in Table 2, which is an IEEE 802.15.4 compliant sensor architecture for low-power and low-cost [25, 26]. IEEE 802.15.4 compliant networks support the lower data rate compared to other wireless networks, and we assumed that the lower data rate is enough for applications of WSNs [27]. We also implemented MRMAC and two multi-channel MAC protocols (MuChMAC [4] and EM-MAC [6]) to compare the performance. We used the parameters of MRMAC, EM-MAC, and MuChMAC from [6, 14], and [4], respectively. The parameters of RM-MAC were set as follows, CSV_{thresh} and CSV_0 were 0.6 and 0.8, respectively, and the weight value p was 0.05. The target packet delivery ratio α is 0.99, and the limited number of retransmissions k was from 0 to 4. Nodes in RM-MAC, MRMAC, and EM-MAC woke up following a uniform distribution between 0.5 and 1.5 seconds, and the nodes in MuChMAC woke up every second.

4.1 Simulation results in simple topologies

First, we simulated the four protocols in chain/cross topologies as shown in Fig. 6. We performed the simulations 20 times and calculated the average in each topology. The simulation time was 500 s, and source nodes *S* generated and transmitted a data packet to a destination node *D* every second. Table 3 shows the simulation results of each topology in terms of the packet delivery latency and the total number of collisions. In the chain topology, no collisions occur, so we did not obtain the number of collisions.

In both the chain topology and the one-hop cross topology, the results of EM-MAC and MuChMAC were higher than those of RM-MAC and MRMAC. The reason is that a sender in EM-MAC and MuChMAC has to wait until a receiver wakes up, so the delivery latency increases. In addition, since the senders *S* in the one-hop cross topology intend to transmit their data packet simultaneously, collisions repeatedly occur on the destination node *D* (the multiple-senders-single-receiver problem). On the other hand, in RM-MAC and MRMAC, the next data packet transmission is reserved by the reservation mechanism, so a data packet is transmitted with lower delay. The reservation mechanism also helps the destination node *D* to coordinate transmissions, so RM-MAC and MRMAC can avoid the repeated collisions caused by the multiple-senders-single-receiver problem. As the results, RM-MAC and MRMAC can reduce not only the delivery latency but also the number of collisions through the reservation mechanism. A comparison of RM-MAC and MRMAC shows that the delivery latency in RM-MAC is higher than that in MRMAC. The difference between these results can be explained by the fact that multi-channel communications requires some extra time for network initialization [1].

In the two-hop cross topology, the results of MRMAC were considerably worse than those of RM-MAC. Since MRMAC operates on single-channel communications, data packets transmitted by adjacent nodes [e.g. the sender nodes *S* in Fig. 6(c)] collide. If the collisions occurred in

Table 2 MICAz parameters

Parameter	Value
Bandwidth	250 Kbps
Number of channels	16
Transmission range	40 m
Power in transmitting	52.2 mW
Power in receiving	59.1 mW
Power in sleeping	3 μW

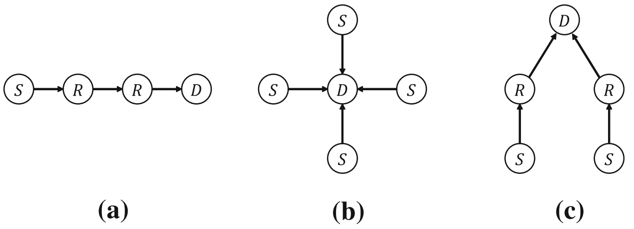


Fig. 6 **a** Chain topology; **b** one-hop cross topology; **c** two-hop cross topology

Table 3 Simulation results in the simple chain/cross topologies

Protocols	Chain topology	One-hop cross topology		Two-hop cross topology	
	Delivery latency (s)	Delivery latency (s)	Number of collisions	Delivery latency (s)	Number of collisions
RM-MAC	0.709	0.163	8.7	0.459	5.3
EM-MAC	2.222	0.720	351.4	1.401	233.4
MuChMAC	2.197	0.776	367.7	1.382	241.2
MRMAC	0.103	0.044	11.6	1.693	492.4

reserved transmissions, the delivery latency significantly increase because MRMAC cannot retransmit the collided data packets in the reservations. In contrast, RM-MAC operates on multi-channel communications that helps to avoid interference and collisions between transmissions of adjacent nodes [28]. In addition, RM-MAC can easily retransmit the data packets even if the collisions occur in the reserved transmissions.

We also performed simulations in a one-hop topology with interferers, which were pairs of senders and receivers. We varied the number of interferers from 0 to 5, which transmitted/received a data packet every 0.1 s. Each interferer in RM-MAC, EM-MAC, and MuChMAC randomly chose a channel for the transmissions, and the interferers in MRMAC used only the first channel. Each simulation was performed 20 times, and the error bars indicate the 95 % confidence intervals.

Figures 7 and 8 show the packet delivery latency and the duty-cycle of each protocol with interferers, respectively. The results of the three multi-channel protocols slightly increased as the number of interferers increased. Through the multi-channel communications, data packets in these protocols are transmitted with less interference. It leads to the reduced number of retransmissions caused by the interference, so the duty-cycle and the delivery latency are also reduced. In contrast, MRMAC operates on single-channel communications, so it is hard to avoid interference. It causes long delivery latency and the high duty-cycle under interference. A comparison of the multi-channel protocols shows that the delivery latency in MuChMAC increased by about 0.123, whereas that in RM-MAC and EM-MAC increased by about 0.083 and 0.094, respectively. The increase of the duty-cycle in MuChMAC (1.11 %) was also more than that in RM-MAC (0.62 %) and EM-MAC (0.55 %). The reason is that nodes in RM-MAC and EM-MAC can avoid communicating on the interfered channels by considering channel conditions. These results indicate that the consideration of channel conditions in RM-MAC and EM-MAC helps to improve the network performance and energy efficiency under interference.

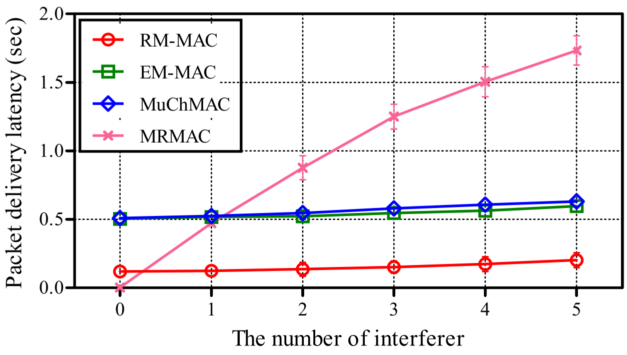


Fig. 7 Packet delivery latency in the one-hop topology with interferers

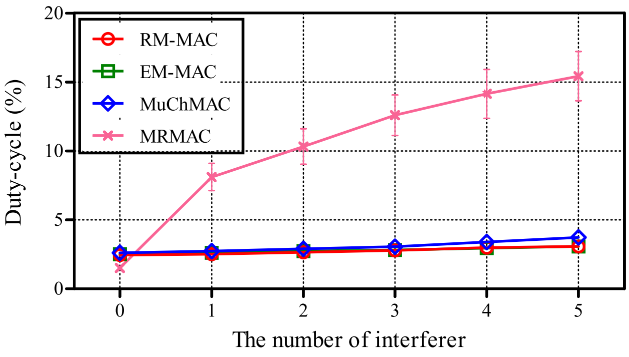


Fig. 8 Duty-cycle in the one-hop topology with interferers

4.2 Simulation results in a random topology

We also simulated the four protocols in 20 random topologies that were similar to real environments. In each simulation, 50 nodes were randomly deployed, and one of them was selected as a sink node. Next, k periodic data streams were generated following a Poisson distribution $P(k) = \frac{\lambda^k e^{-\lambda}}{k!}$ with mean rates $\lambda = 4$ and 8. The duration of each periodic data stream was exponentially determined, and we varied the data packet generation interval in the periodic data stream to 2, 4, 6, 8, and 10 s. Through these parameters, we generated various traffic conditions, e.g., heavy traffic generated by $\lambda = 8$ and the interval of 2 s, and light traffic generated by $\lambda = 4$ and the interval of 10 s. Then, in each traffic condition, we obtained the simulation results with a 95 % confidence interval.

Figure 9 shows the average number of collisions in each protocol. The results of all protocols increased in heavy traffic. Most of all, the number of collisions in MRMAC was significantly higher because MRMAC operates on single-channel communications. Transmissions of adjacent nodes in single-channel communications easily interfere with each other, so it causes a large number of collisions. On the other hand, since other protocols operate on multi-channel communications, fewer collisions occur than MRMAC. A comparison of the multi-channel protocols shows that RM-MAC reduced the number of collisions by about 43.3 and 23.8 % compared to EM-MAC and MuChMAC, respectively. As we mentioned in Sect. 4.1, collisions occur repeatedly on cross nodes in EM-MAC and MuChMAC, but RM-MAC can reduce the number of the repeated collisions through the reservation mechanism.

Figure 10 shows the average duty-cycle in the four protocols. In real applications, a few nodes transmit/receive data packets for each periodic data stream, so the average duty-cycle was lower than that in the simple topologies. Frequent collisions and interference lead to additional active state time such as back-off time and retransmission time, so reducing the number of collisions helps to reduce duty-cycle. Since RM-MAC achieved the least number of collisions among the four protocols, the duty-cycle in RM-MAC was also the lowest, which was 77.9 % of that in EM-MAC, 70.0 % of that in MuChMAC, and 34.1 % of that in MRMAC. These results indicate that the RM-MAC improved energy efficiency by reducing the number of collisions.

Figure 11 shows the average packet delivery latency from source nodes to the sink node. The delivery latency of RM-MAC was 60.0–70.9 % of that in EM-MAC and 59.7–70.9 % of that in MuChMAC. RM-MAC reserves the next data packet transmission through the reservation mechanism, so the data packet in RM-MAC can be delivered with lower delay. In MRMAC which also uses the reservation mechanism, the packet delivery latency was higher than that in other protocols, especially in $\lambda = 8$ and the data packet generation interval of 2 s. This result suggests that MRMAC cannot use the reservation mechanism in heavy traffic due to a large number of collisions. In contrast with MRMAC, the packet delivery latency in RM-MAC did not considerably increase regardless of the data packet generation interval, which was reduced by about 15.6–50.1 % compared to MRMAC. This means that the reservation mechanism of RM-MAC is more stable than MRMAC to support heavy traffic conditions.

Figure 12 shows the total size of the communication overhead incurred due to reservation mechanism. EM-MAC and MuChMAC do not use the reservation mechanism, so we did not obtain any communication overhead of them. MRMAC transmits the piggybacked information with every transmission, so the communication overhead in MRMAC considerably increased as the traffic becomes heavier. On the other hand, the piggybacked information in RM-MAC is transmitted only if necessary, so the communication overhead was significantly reduced by 8.9–15.0 % compared to MRMAC. In addition, it seems that the simulation results can verify our analysis where the

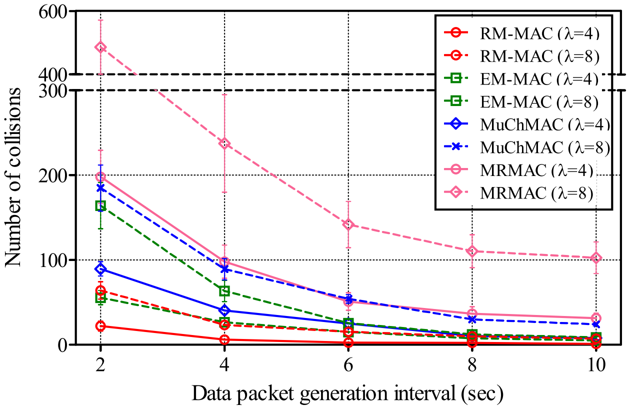


Fig. 9 The number of collisions in the random topologies

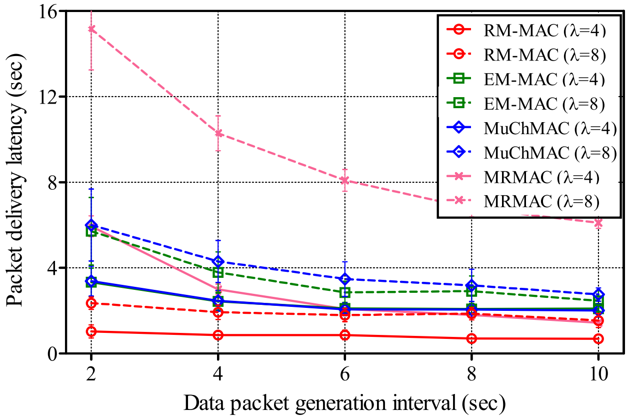


Fig. 11 Packet delivery latency in the random topologies

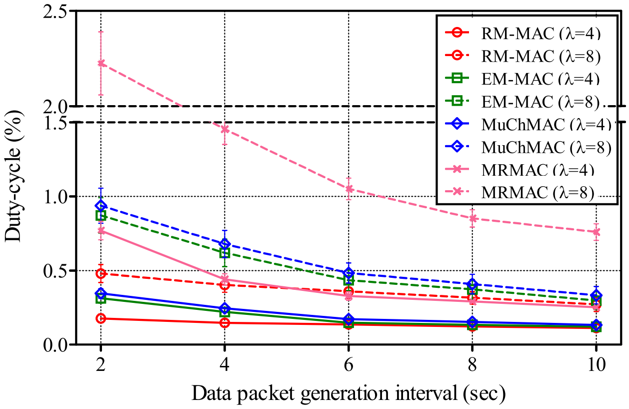


Fig. 10 Duty-cycle in the random topologies

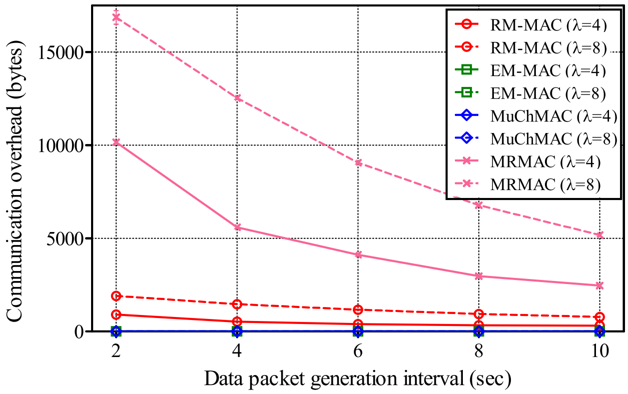


Fig. 12 Communication overhead in the random topologies

communication overhead in RM-MAC and MRMAC is bounded by a constant and $\mathcal{O}(n)$, respectively. It indicates that the RM-MAC can transmit periodic data streams with low communication overhead regardless of heavy traffic.

5 Conclusion

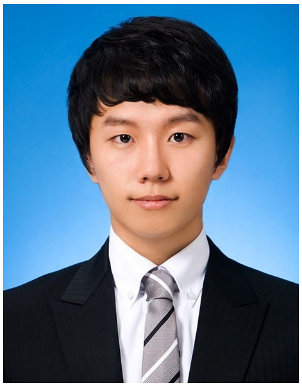
In this paper, we have adopted a reservation mechanism to a multi-channel MAC protocol for periodic convergecast in WSNs (RM-MAC). As the simulation results in the previous section, the reservation mechanism helps RM-MAC to reduce the packet delivery latency by about 62.5–71.4 % compared to other multi-channel MAC protocols (EM-MAC and MuChMAC). In addition, the reservation mechanism can mitigate the multiple-senders-single-receiver problem, so the energy efficiency of RM-MAC is also improved by about 1.2–1.4 times. We have also compared RM-MAC with MRMAC which also uses a similar mechanism in single-channel communications. Since nodes in RM-MAC can avoid a number of collisions and interference through multi-channel communications, RM-MAC improves the packet delivery latency and energy efficiency by about 2–6.4 and 2.9 times, respectively, regardless of heavy traffic. We have also verified that RM-MAC can significantly reduce the communication overhead incurred due to the reservation mechanism.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2014R1A2A2-A01006957) and Institute for Information & Communication Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 10041244, SmartTV 2.0 Software Platform).

References

1. Incel, O. (2011). A survey on multi-channel communication in wireless sensor networks. *Computer Networks*, *55*(13), 3081–3099.
2. Zhou, G., Huang, C., Yan, T., He, T., Stankovic, A., & Abdelzaher, T. (2006). MMSN: Multi-frequency media access control for wireless sensor networks. In *Proceedings of IEEE International conference on computer communications (INFOCOM)* (pp. 1–13).
3. Incel, O., Hoesel, L., Jansen, P., & Havinga, P. (2011). MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Networks*, *9*(1), 73–94.
4. Borms, J., Steenhaut, K., & Lemmens, B. (2010). Low-overhead dynamic multi-channel MAC for wireless sensor networks. In *Proceedings of European conference on wireless sensor networks (EWSN)* (pp. 81–96).
5. Kim, Y., Shin, H., & Cha, H. (2008). Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Proceedings of ACM/IEEE international conference on information processing in sensor networks (IPSN)* (pp. 53–63).
6. Tang, L., Sun, Y., Gurewitz, O., & Johonson, D. (2011). EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of ACM international symposium on mobile ad hoc networking and computing (MobiHoc)*, Article No. 23.
7. Landsiedel, O., Ghadimi, E., Duquennoy, S., & Johansson, M. (2012). Low power, low delay: Opportunistic routing meets duty cycling. In *Proceedings of ACM/IEEE international conference on information processing in sensor networks (IPSN)* (pp. 185–196).
8. Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, *12*(3), 493–506.
9. Ghosh, A., Incel, O., Kumar, V., & Krishnamachari, B. (2011). Multichannel scheduling and spanning trees: Throughput-delay tradeoff for fast data collection in sensor networks. *IEEE/ACM Transactions on Networking*, *19*(6), 1731–1744.
10. Luo, C., Wu, F., Sun, J., & Chen, C. (2009). Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of ACM international conference on mobile computing and networking (MobiCom)* (pp. 145–156).
11. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002). Wireless sensor networks for habitat monitoring, In *Proceedings of ACM international workshop on wireless sensor networks and applications (WSNA)* (pp. 88–97).
12. Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., et al. (2004). A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, *46*(5), 605–634.
13. Yu, L., Wang, N., & Meng, X. (2005). Real-time forest fire detection with wireless sensor networks, In *Proceedings of international conference on wireless communications, networking and mobile computing (WiCom)* (pp. 1214–1217).
14. Hong, J., Jang, I., Lee, H., Yang, S., & Yoon, H. (2010). MRMAC: Medium reservation MAC protocol for reducing end-to-end delay and energy consumption in wireless sensor networks. *IEEE Communications Letters*, *14*(7), 614–616.
15. Buettner, M., Yee, G., Anderson, E., & Han, R. (2006). X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of ACM international conference on embedded networked sensor systems (SenSys)* (pp. 307–320).
16. Ansari, J., Ang, T., & Mahonen, P. (2010). Spectrum agile medium access control protocol for wireless sensor networks. In *Proceedings of IEEE international conference on sensor and mesh and ad hoc communications and networks (SECON)* (pp. 1–9).
17. Cano, C., Bellalta, B., & Oliver, M. (2012). Receiver-initiated vs. short-preamble burst MAC approaches for multi-channel wireless sensor networks. In *Proceedings of EUNICE international conference on information and communications technologies* (pp. 23–32).
18. Borbash, S., Ephremides, A., & McGlynn, M. (2007). An asynchronous neighbor discovery algorithm for wireless sensor networks. *Ad Hoc Networks*, *5*(7), 998–1016.
19. Kandhalu, A., Lakshmanan, K,. & Rajkumar, R. (2010). U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of ACM/IEEE international conference on information processing in sensor networks (IPSN)* (pp. 350–361).
20. Jeon, J., & Ephremides, A. (2012). Neighbor discovery in a wireless sensor network: Multipacket reception capability and physical-layer signal processing. In *Proceedings of Allerton conference on communication, control, and computing (Allerton)* (pp. 310–317).

21. Sun, Y., Gurewitz, O., & Johnson, D. (2008). RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of ACM international conference on embedded networked sensor systems (SenSys)* (pp. 1–14).
22. Jang, I., Yang, S., Yoon, H., & Kim, D. (2013). EMBA: An efficient multihop broadcast protocol for asynchronous duty-cycled wireless sensor networks. *IEEE Transactions on Wireless Communications*, *12*(4), 1640–1650.
23. CC2420 Datasheet. http://www.ti.com.cn/cn/lit/ds/symlink/cc2420
24. Crossbow Technology Inc. MICAz Datasheet. http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet
25. Cuomo, F., Luna, S. D,. Monaco, U,. & Melodia, T. (2007). Routing in ZigBee: Benefit from exploiting the IEEE 802.15.4 association tree. In *IEEE international conference on communications (ICC)* (pp. 3271–3276).
26. Yoo, S., Chong, P., Kim, D., Doh, Y., Pham, M., Choi, E., et al. (2010). Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4. *IEEE Transactions on Industrial Electronics*, *57*(11), 3868–3876.
27. Kredo, K, I. I., & Mohapatra, P. (2007). Medium access control in wireless sensor networks. *Elsevier Computer Networks*, *51*(4), 961–994.
28. Jang, I., Pyeon, D., Yoon, H., & Kim, D. (2015). Channel-quality-aware multihop broadcast for asynchronous multi-channel wireless sensor networks. *Springer Wireless Networks*. http://link.springer.com/content/pdf/10.1007



Dohoo Pyeon received his B.S. degree (*summa cum laude*) in the College of Information and Communications at Hanyang University, Seoul, Korea, in 2011, and his M.S. degree in Computer Science at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2013. He is currently towards Ph.D. degree in Computer Science at KAIST. His research interests include the communication protocols, such as medium access control, routing, and broadcast protocols, for energy efficiency in wireless sensor networks and for efficient spectrum utilization in cognitive radio networks.

Ingook Jang received the B.S. degree (*summa cum laude*) in Computer Science and Engineering from the Chung-Ang University, Seoul, Korea, in 2008. He is currently working toward Ph.D. degree through the Integrated Master's and Doctoral Degree Program in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His current research interests include the design and analysis of energy efficient communication protocols, especially broadcast, medium access control, scheduling of packet transmissions, with applications in wireless networks, Internet of Things (IoT), and broadband access networks.

Hyunsoo Yoon received the B.S. degree in Electronics Engineering from Seoul National University, Seoul, Korea, in 1979, the M.S. degree in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1981, and the Ph.D. degree in Computer and Information Science from The Ohio State University, Columbus, in 1988. He is currently a professor of the Department of Computer Science, KAIST. During 1978–1980, he was with the Tongyang Broadcasting Company, Korea, then Samsung Electronics Company, Seoul, Korea, during 1980–1984. From 1988 to 1989, he was a Member of the Technical Staff with AT&T Bell Labs, Indial Hill, IL. Since 1989, he has been a professor with the Department of Computer Science, Korea Advanced Institute of Science and Technology. His research interests include mobile ad hoc networks, wireless networks, and network security.

Dongwook Kim received his B.S. degree in Information and Computer Engineering from Ajou University, Suwon-si, South Korea, in 2002, and his M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2009, respectively. Since 2009, he is a senior engineer at the Network Business Division, Samsung Electronics. His research interests include the design and optimization of LTE (Long Term Evolution)-advanced systems with smart scheduling technologies, optimal deployment of mobile sensor networks, broadcast techniques and MAC protocol designs for sensor networks.