

Channel-quality-aware multihop broadcast for asynchronous multi-channel wireless sensor networks

Ingook Jang¹ · Dohoo Pyeon¹ · Hyunsoo Yoon¹ · Dongwook Kim²

Published online: 13 October 2015
© Springer Science+Business Media New York 2015

Abstract In this paper, we propose *Multi-channel EMBA (M-EMBA)*, efficient multihop broadcast for asynchronous multi-channel wireless sensor networks. Our scheme employs two channel-quality-aware forwarding policies of *improved forwarder's guidance* and *fast forwarding* to improve multihop broadcast performance. The improved forwarder's guidance allows forwarders to transmit broadcast messages with *guidance* to their receivers through channels with good quality. The guidance indicates how each receiver should forward the broadcast message to its neighbor nodes. The improved forwarder's guidance tremendously reduces redundant transmissions and collisions. Fast forwarding allows adjacent forwarders to send their broadcast messages simultaneously through different channels that have good quality, which helps to reduce multihop broadcast latency and improve multi-channel broadcast utility. In this work, we evaluate the multihop broadcast performance of M-EMBA through theoretical analysis of the system design and empirical simulation-

based analysis. We implement M-EMBA in *ns-2* and compare it with the broadcast schemes of ARM, EM-MAC, and MuchMAC. The performance results show that M-EMBA outperforms these protocols in both light and heavy network traffic. M-EMBA reduces message cost in terms of goodput, total bytes transmitted, as well as broadcast redundancy and collision. M-EMBA also achieves a high broadcast success ratio and low multihop broadcast latency. Finally, M-EMBA significantly improves energy efficiency by reducing average duty cycle.

Keywords Wireless sensor networks · Multi-channel · Multihop broadcast · Asynchronous duty-cycling

1 Introduction

Multihop broadcast in wireless sensor networks (WSNs) is a fundamental service for high-level operations in both single- and multi-channel environments. Its purpose is to convey a broadcast message to all nodes reliably and energy-efficiently in a network, and it is utilized for crucial applications, such as query propagation [1], route discovery [2], and network configuration and reprogramming [3, 4]. In multihop broadcast, a source node produces burst traffic in a short time period, and these broadcast packets are spread across the whole network (its pattern is similar to that of an infection). This inherent broadcast mechanism causes serious redundancy and collision, called the *broadcast storm problem* [5], and it can hinder transmission of sensing data for data gathering which is the intrinsic goal of WSNs. Therefore, it is essential that multihop broadcast should be performed with the minimum number of transmissions and collisions.

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

Ingook Jang
ikjang@nslab.kaist.ac.kr

Dohoo Pyeon
dhpyeon@nslab.kaist.ac.kr

Hyunsoo Yoon
hyoon@nslab.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, Yuseong-gu, Suwon-si, Gyeonggi-do 443-742, Korea

Multihop broadcast is relatively simple to be designed on top of synchronous medium access control (MAC) protocols [6–8]. Synchronization allows nodes to easily exchange messages in common active states. In addition, a broadcast message transmitted by a sender reaches multiple receivers. However, it is laborious to support multihop broadcast in asynchronous MAC protocols because sensor nodes independently wake up. Moreover, nodes wake up according to their own wake-up schedules on different channels. To support multihop broadcast in asynchronous multi-channel MAC protocols, a node should transmit a broadcast message to each of its neighbor nodes one by one. It is problematic to support multihop broadcast efficiently due to collisions and redundant transmissions of the same broadcast messages. These limitations highlight the importance of designing an efficient multihop broadcast protocol for asynchronous multi-channel WSNs.

Recently proposed asynchronous multi-channel MAC protocols [9–14] support simple 1-hop broadcast, but most of them cannot efficiently support multihop broadcast due to some limitations. Sensor multi-channel MAC (SMC MAC) [9] and asynchronous receiver-initiated multi-channel MAC (ARM) [10] produce broadcast channel saturation because they use a dedicated broadcast channel. A sender in interference-aware multi-channel MAC (IMMAC) [11] transmits a broadcast message on every available channel, so it incurs high broadcast cost in multihop delivery. Efficient multi-channel MAC (EM-MAC) [13] is a receiver-initiated MAC protocol that sends a broadcast message to neighbor nodes one by one. Multi-channel MAC (MuchMAC) [14] repeatedly transmits the same broadcast messages for a certain time as long as the sleep duration of a receiver. These two protocols suffer from a number of collisions and redundant transmissions. To the best of our knowledge, there has not been a multihop broadcast protocol yet that works separately from MAC protocols for asynchronous multi-channel WSNs.

Many researchers have been exploring multihop broadcast in asynchronous single-channel WSNs. Asynchronous duty-cycle broadcasting (ADB) [15] and CEN / DIS [16] reduce broadcast redundancy by deterministically selecting broadcasting nodes to transmit broadcast messages. Wang et al. [17] proposed an algorithm to determine broadcast sequences by transforming the broadcast problem to the shortest path problem. However, these protocols focus only on multihop broadcast in single-channel environments and does not provide any multi-channel support.

We recently proposed an efficient multihop broadcast protocol for asynchronous duty-cycled WSNs (EMBA) [18]. In EMBA, each *forwarder* that wants to send a broadcast message gives a guideline to neighbor nodes about how to forward the broadcast message (*forwarder's guidance*). This mechanism can significantly reduce the

number of redundant transmissions and collisions; however, its design philosophy does not consider multi-channel characteristics. To support multihop broadcast for asynchronous multi-channel WSNs, the following key features should be taken into account: channel assignment and rendezvous, how to reduce broadcast redundancy and collision, and how to improve transmission concurrency by exploiting the characteristics of multi-channel communications. These aspects will be thoroughly discussed in Sect. 2.

In this paper, we propose channel-quality-aware multihop broadcast for asynchronous multi-channel WSNs (*multi-channel EMBA*, *M-EMBA*), which operates on top of asynchronous multi-channel MAC protocols. M-EMBA is an extension of EMBA for multi-channel asynchronous WSNs, which employs *improved forwarder's guidance* and *fast forwarding*. M-EMBA informs MAC protocols of the best channel to transmit broadcast messages. This cross-layer solution enables multihop broadcast to be conducted through channels with good quality. M-EMBA significantly reduces broadcast redundancy and collision through the improved forwarder's guidance, which incurs less overhead than the forwarder's guidance of EMBA. M-EMBA also improves broadcast utility through fast forwarding which exploits the characteristics of multi-channel communications.

The main contribution of this paper is the proposal and thorough evaluation of M-EMBA through theoretical analysis of the system design and empirical simulation-based analysis. Our theoretical analysis of M-EMBA gives performance upper bounds that demonstrate the practicality and feasibility of the system design. We evaluate the multihop broadcast performance of M-EMBA by conducting a thorough simulation study. To evaluate the performance of multihop broadcast, we provide representative metrics: (1) the number of collisions, (2) broadcast redundancy, (3) total bytes transmitted, (4) goodput, (5) broadcast success ratio, (6) multihop broadcast latency, and (7) duty cycle. Our simulation results show that M-EMBA improves multihop broadcast performance in terms of all these metrics compared to existing protocols.

The rest of this paper is organized as follows. Section 2 presents the design of M-EMBA, and we give analyses of M-EMBA in Sect. 3. Performance results of M-EMBA are provided in Sect. 4 and finally, Sect. 5 concludes this paper.

2 M-EMBA

In this section, we discuss design issues of M-EMBA: advertisement procedure, channel assignment and rendezvous, the improved forwarder's guidance, and fast forwarding.

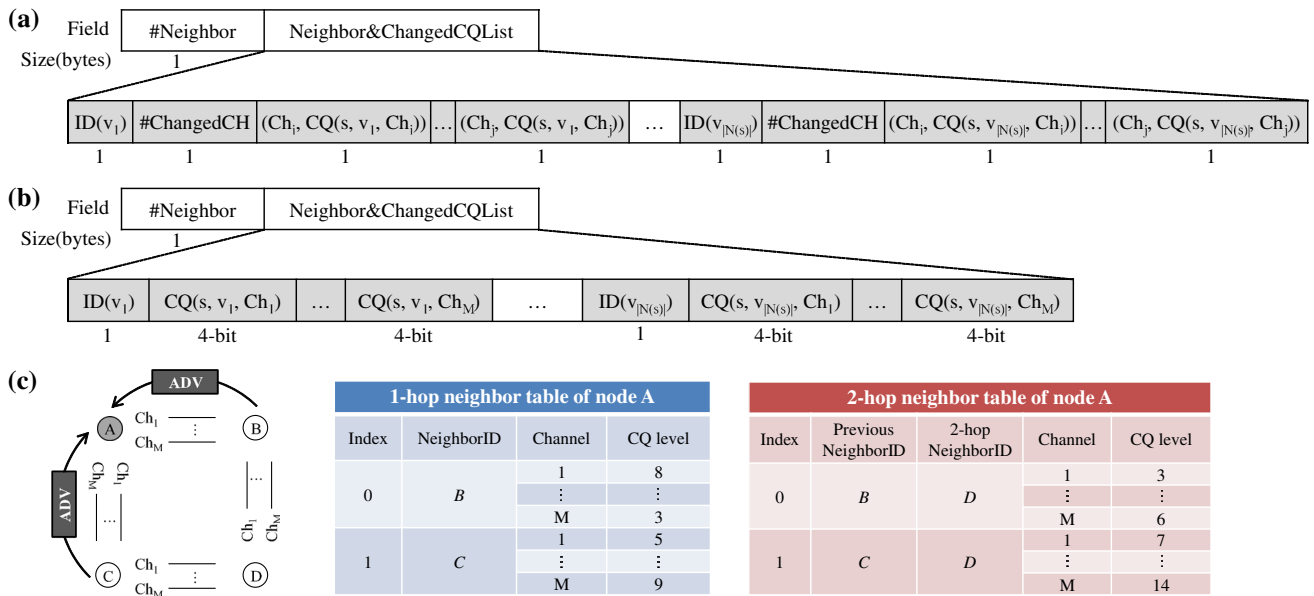


Fig. 1 Advertisement procedure for M-EMBA. **a** Structure of an ADV message in the advertisement method 1. **b** Structure of an ADV message in the advertisement method 2. **c** 1-hop and 2-hop neighbor tables of node A after receiving advertisement messages from nodes B and C

2.1 Advertisement procedure for M-EMBA

In M-EMBA, each node maintains a 1-hop neighbor table that stores the quality of available channels from itself to each of its neighbor nodes. Channel quality can be measured by existing channel estimation algorithms, such as [19] and [20]. Nodes periodically measure and store the quality of the channels they operate on. We denote the quality of channel *i* from node *s* to node *v* as $CQ(s, v, Ch_i)$, and it is represented by 16 priority levels (0–15). Each node periodically exchanges its 1-hop neighbor information with its neighbor nodes (called an *advertisement procedure* in M-EMBA).

Each node *s* transmits an advertisement (ADV) message to its neighbor nodes, which plays double roles: (1) notifying its neighbor node list and (2) advertising the changed channel status. Figures 1(a) and (b) show the structure of an ADV message composed of two fields: *#Neighbor* and *Neighbor&ChangedCQList*. The *#Neighbor* field contains the number of neighbor nodes of node *s*.

At this point, we face an important issue that must be discussed. M-EMBA should determine how ADV messages are exchanged between nodes. We introduce the following two feasible advertisement methods.

2.1.1 Advertisement method 1

An ADV message includes channel information only whose channel quality values are changed, as shown in Fig. 1(a). For one neighbor node *v_n*, node *s* creates one or

more elements in the same quantity as the number of channels whose quality values are changed. The *Neighbor&ChangedCQList* field is a list that includes neighbor node IDs and changed channel information between node *s* and its neighbor nodes. When the number of available channels is *M*, for a neighbor node *v_n*, each element of the *Neighbor&ChangedCQList* field generated by node *s* is presented as follows:

$$[v_n][\#CCH][Ch_i, CQ(s, v_n, Ch_i)] \dots [Ch_j, CQ(s, v_n, Ch_j)], \tag{1}$$

$$1 \leq n \leq |N(s)|, 1 \leq i, j \leq M \tag{2}$$

where $|N(s)|$ is the number of neighbor nodes of node *s*, *v_n* is the ID of node *s*' neighbor node *v_n* (1 byte), and *#CCH* is the number of channels with changed quality (1 byte). Here, $[Ch_i, CQ(s, v_n, Ch_i)]$ represents the information of the changed channel *i* (1 byte). When the number of available channels is 16, *Ch_i* and $CQ(s, v_n, Ch_i)$ are encoded with 4 bits, respectively.

If a node *s* has not found any channel quality change, it assigns 0 to *#CCH*. For example in Fig. 1(c), if the quality of *Ch₁* between nodes *B* and *D* is only changed for a certain period of time, *B* assigns $\{[[A][0]], [[D][1][1, 7]]\}$ to the *Neighbor&ChangedCQList* field in its ADV message.

If half of the channels are changed among 16 available channels within a certain period of time, about 10 bytes are required per neighbor node. If no channels are changed compared to the previous advertisement procedure, about 2 bytes are needed per neighbor.

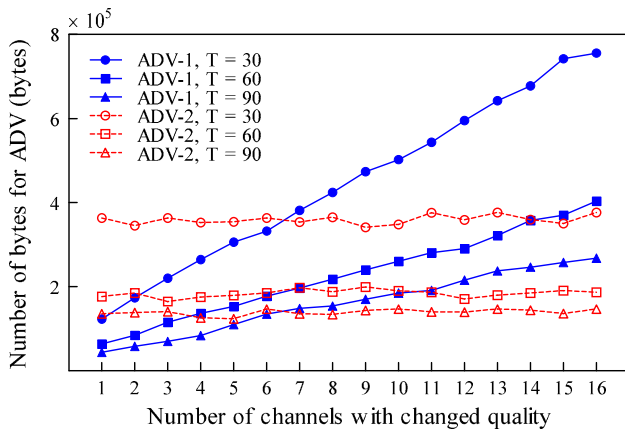


Fig. 2 Total number of bytes of six variants of advertisement methods

2.1.2 Advertisement method 2

An ADV message contains a whole 1-hop neighbor table, as shown in Fig. 1(b). The *Neighbor&ChangedCQList* field contains the neighbor node IDs and quality values of all channels. For a neighbor node v_n , each element of the *Neighbor&ChangedChQList* field generated by node s is presented as follows:

$$[v_n][CQ(s, v_n, Ch_1)] \dots [CQ(s, v_n, Ch_M)], \tag{3}$$

$$1 \leq n \leq |N(s)| \tag{4}$$

where v_n is the ID of node s ' neighbor node v_n (1 byte), and $[CQ(s, v_n, Ch_k)] (1 \leq k \leq M)$ represents the quality value of channel k . Each value of $CQ(s, v_n, Ch_k)$ is encoded with 4 bits (16 priority levels). When the number of available channels is 16, an ADV message in advertisement method 2 is always 8 bytes per neighbor node (16×4 bits).

If ADV message loss occurs due to collision or other network failures, M-EMBA follows the resolution mechanism of the underlying MAC protocol (e.g., retransmission) to maintain up-to-date channel information. By exchanging ADV messages, multihop broadcast can be conducted through channels with good quality. M-EMBA significantly reduces broadcast redundancy and collision in multi-channel environments by using 2-hop neighbor information.

There is a trade-off between advertisement methods 1 and 2 in relation to the frequency of channel quality changes. Figure 2 shows the simulation results of six variants of advertisement methods during 1000 s in a $1500 \text{ m} \times 1500 \text{ m}$ network with 16 channels and 50 nodes, where T is the advertisement procedure interval (i.e., 30, 60, and 90 s). We vary the number of channels whose qualities are changed within time period of T . In *ADV-1*, the number of bytes for ADV shows a proportional behavior with a constant slope as the number of channels changed increases, and its slope becomes steeper as the

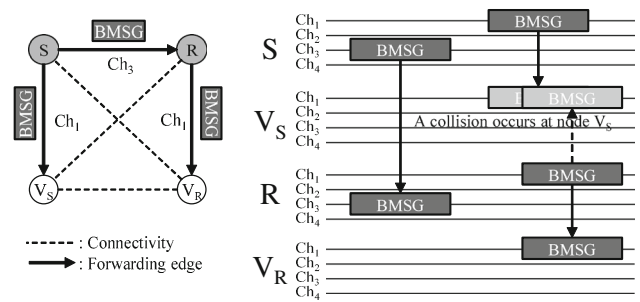


Fig. 3 Multi-channel hidden terminal problem

advertisement period of T decreases. On the other hand, *ADV-2* shows an almost constant number of bytes regardless of the increased number of channels changed. Therefore, *ADV-1* has an advantage in static networks where channel quality changes infrequently. Otherwise, *ADV-2* is favorable.

2.2 Cross-layer channel assignment for multihop broadcast

Cross-layer design is a technique that can be used to improve network performance in wireless networks by exploiting interactions between two or more layers. In M-EMBA, channel information to be used for multihop broadcast is provided to MAC protocols. M-EMBA maintains 2-hop neighbor information through the advertisement procedure. When a forwarder has a broadcast message to transmit to a receiver, M-EMBA informs the underlying MAC protocol of the best channel between the forwarder and receiver. Then, the forwarder will rendezvous with the receiver on that channel even though it is different from the channel initially scheduled by the MAC protocol. This cross-layer solution helps the delivery of broadcast messages over reliable channels with good quality, so it can reduce the number of transmissions, collisions, and retransmissions.

In multi-channel environments, channel assignment schemes should take into account how to reduce the communication interference caused by the *multi-channel hidden terminal problem* [23], which can degrade throughput performance. Figure 3 shows an example of the multi-channel hidden terminal problem in multihop broadcast with 4 channels. Node S sends a broadcast message (*BMSG*) to node V_S through Ch_1 after sending it to node R . As a new forwarder, node R also starts to transmit the *BMSG* to node V_R through Ch_1 , which is chosen as the best channel between R and V_R . In this situation, the *BMSG* sent by node R arrives at not only node V_R but also node V_S , so a collision occurs at V_S .

To mitigate this problem, we employ a simple cross-layer channel assignment scheme in our M-EMBA, in

which a forwarder S chooses the best channel among available ones except the channel to be used by the new forwarder R 's next transmission. Through this solution, the multi-channel hidden terminal problem can be easily solved; thus, it can prevent throughput degradation.

2.3 Channel rendezvous for various asynchronous multi-channel MAC protocols

A forwarder s looks into its neighbor table to search the best channel from itself to a receiver r . To actually transmit a broadcast message, node s informs node r of the selected channel; thus, they can rendezvous on that channel.

The challenge, however, is that existing asynchronous multi-channel MAC protocols are classified into multiple types of approaches: (1) *control-channel-based*, (2) *semi-dynamic channel-assignment-based*, and (3) *dynamic channel-assignment-based* [21, 22]. In control-channel-based approaches, such as SMC MAC [9] and ARM [10], a pair of a forwarder and a receiver negotiates for a channel to be used for a data transmission via the common control channel. In semi-dynamic channel-assignment-based approaches, such as IMMAC [11], CMAC [12], and EM-MAC [13], channel assignment is periodically executed, and a sender switches its radio to a receiver's channel for communication. It is possible for nodes to switch their operating channel to another one due to some situations, such as events, channel overload, and network failure. In dynamic channel-assignment-based approaches, such as MuchMAC [14] and EM-MAC [13] (EM-MAC may be also dynamic), every node dynamically switches its radio from one channel to another at every wake-up time, and a forwarder can predict the receiver's wake-up channel and time. The differences between these approaches lead to subtle differences in channel rendezvous. We briefly explain how channel rendezvous is designed according to each approach.

- Control-channel-based: A forwarder informs a receiver of the best channel via the common control channel by piggy-backing it in control messages. They wake up on the channel chosen by the forwarder and transmit/receive a broadcast message.
- Semi-dynamic channel-assignment-based: Although each node uses a constant channel or a pseudo-randomly selected channel, multihop broadcast should be conducted through channels with good quality. A forwarder and a receiver wake up according to the receiver's schedule determined by the underlying MAC protocol, and then the forwarder announces the best channel to the receiver. After that, both of them wake up on the best channel and transmit/receive a broadcast message.
- Dynamic channel-assignment-based: Similarly in semi-dynamic channel-assignment-based approaches, the

forwarder and receiver promise to rendezvous on the best channel. The receiver does not select the next wake-up channel dynamically, and it wakes up on the best channel announced by the forwarder to rendezvous and receive a broadcast message.

Since M-EMBA operates on top of asynchronous multi-channel MAC protocols, M-EMBA can be easily implemented for other approaches not mentioned above if they provide proper schemes of channel rendezvous.

2.4 Improved forwarder's guidance

In M-EMBA, a forwarder provides guidance to each of its neighbor nodes by using 2-hop neighbor information. The guidance indicates how each node should propagate a broadcast message to reduce broadcast redundancy and collision. The guidance is generated by each forwarder and is transmitted together with a broadcast message through a channel with good quality. Each node that receives the guidance enclosed in the broadcast message works as a new forwarder to deliver the broadcast message to its neighbor nodes according to the guidance.

To offer guidance to neighbor nodes, a forwarder s creates a *guidance list* (GL) for each neighbor node r (referred to here as GL_r^s). If forwarder s should transmit the broadcast message to three neighbor nodes r_1, r_2 , and r_3 , s creates three different guidance lists ($GL_{r_1}^s, GL_{r_2}^s$, and $GL_{r_3}^s$) for each of them. The sequence to transmit the broadcast message to multiple neighbor nodes is determined based on the sorted order of quality values of the best channel between a forwarder and each of its neighbor nodes. For example, node s covers node r_2 first if the quality values of the best channel from s to r_1, r_2 , and r_3 are 9, 13, and 8, respectively (*covering* means transmitting the broadcast message).

When forwarder s makes the GL for a neighbor node r that has n neighbor nodes, GL_r^s is composed of n elements ($GL_r^s[v_1], \dots, GL_r^s[v_n]$) and each element $GL_r^s[v_i] (1 \leq i \leq n)$ is assigned one of two states of guidance as follows.

- **REACHED**: This means that node v_i has already been reached by the broadcast message or will eventually receive it from another node that has a better channel to v_i than node r . If node r receives this guidance, it does not do anything for neighbor node v_i .
- **OBLIGATED**: Node r has an obligation to transmit the broadcast message to node v_i because the link from r to v_i is the best one.

The improved forwarder's guidance of M-EMBA uses only two states for multihop broadcast, whereas three states (**COVERED**, **DELEGATED**, and **OBLIGATED**) are used in the forwarder's guidance of EMBA. **COVERED** and

Table 1 Sets maintained by a forwarder s for the i -th broadcast message ($BMSG^i$)

Set	Description	Initial set
$N(s)$	Neighbor nodes' IDs of a forwarder s	
$RCH^i(s)$	Neighbor nodes' IDs that have already been <i>reached</i> by $BMSG^i$, or will be <i>reached</i> by $BMSG^i$ from other nodes with better channels	ϕ
$OBLG^i(s)$	Neighbor nodes' IDs to which a forwarder s is <i>obligated</i> to forward $BMSG^i$	$N(s)$

DELEGATED states in EMBA allow receivers to follow the same behavior (not to do anything for neighbor nodes marked as either *COVERED* or *DELEGATED*). Therefore, the improved forwarder's guidance in M-EMBA utilizes a new state *REACHED* combined with *COVERED* and *DELEGATED*. The improved forwarder's guidance makes multihop broadcast simpler and reduces the overhead of guidance lists piggy-backed in broadcast messages.

To support multihop broadcast, each new forwarder in M-EMBA initializes the sets shown in Table 1. These sets are used both to keep track of the broadcast status and to create guidance lists. A pair of a forwarder and a receiver in M-EMBA carries out the following steps:

Algorithm 1 Procedure CREATE_GL()

```

1: //  $s$ : forwarder,  $r$ : receiver,  $i$ : broadcast message index
2: for each node  $v \in N(r)$  do
3:   if  $v = s$  then
4:      $GL_r^s[v] \leftarrow REACHED$ 
5:   else if  $v \in N(s)$  then
6:     if  $v \in RCH^i(s)$  then
7:        $GL_r^s[v] \leftarrow REACHED$ 
8:     else
9:       find  $BEST\_CQ(s, v)$  // from  $s$ 's 1-hop neighbor table
10:      find  $BEST\_CQ(r, v)$  // from  $s$ 's 2-hop neighbor table
11:      if  $BEST\_CQ(s, v) \leq BEST\_CQ(r, v)$  then
12:         $RCH^i(s) \leftarrow RCH^i(s) \cup \{v\}$ 
13:         $OBLG^i(s) \leftarrow OBLG^i(s) - \{v\}$ 
14:         $GL_r^s[v] \leftarrow OBLIGATED$ 
15:      else
16:         $GL_r^s[v] \leftarrow REACHED$ 
17:      end if
18:    end if
19:   else // if  $v \notin N(s)$ 
20:     find node  $t$  which has the best quality channel to  $v$  among  $s$ 's neighbor nodes' channels
21:     if  $r = t$  then
22:        $GL_r^s[v] \leftarrow OBLIGATED$ 
23:     else
24:        $GL_r^s[v] \leftarrow REACHED$ 
25:     end if
26:   end if
27: end for

```

Step 1. Generating GL Algorithm 1 represents the generation of GL_r^s before a forwarder s transmits the i -th broadcast message ($BMSG^i$) to a receiver r . Forwarder s makes a guidance list whose size is identical to the number

of neighbor nodes of receiver r . If a node $v \in N(r)$ is forwarder s or has already received $BMSG^i$, s sets $GL_r^s[v]$ to *REACHED* (lines 3–7). If node v is not covered yet and can communicate with both nodes s and r ($v \in N(r) \cap N(s)$), s decides which node will cover v after looking into its 1-hop and 2-hop neighbor tables (lines 8–17). Forwarder s finds the best channel among available channels from s and r to v . If forwarder s has the best channel to node v , s sets $GL_r^s[v]$ to *REACHED*. Otherwise, $GL_r^s[v]$ is set to *OBLIGATED*. Then, the one that has the best channel will only attempt to cover node v through the best channel.

If a neighbor node v of receiver r ($v \in N(r)$) is not a neighbor node of forwarder s ($v \in N(r) - N(s)$), only one node t that has the best channel is chosen among nodes that are reachable to v in 1-hop. Forwarder s sets only $GL_r^s[v]$ to *OBLIGATED* and assigns others to *REACHED* (lines 19–26). Then, node v will be covered only by node t that has the best channel without any redundant transmissions from multiple forwarders.

Step 2. Channel-assigning and rendezvousing Before transmitting $BMSG^i$ to receiver r , forwarder s selects the best channel as the next operating channel by using the cross-layer solution. Node s informs node r of the selected channel, and then they rendezvous on that channel for broadcast transmission, as described in Sects. 2.2 and 2.3.

Step 3. Transmitting and acknowledging When node r receives $BMSG^i$ with guidance GL_r^s from forwarder s , it analyzes the GL_r^s to update its broadcast status, as shown in Algorithm 2.

Algorithm 2 Procedure UPDATE_BROADCAST_STAT()

```

1: //  $s$ : forwarder,  $r$ : receiver,  $i$ : broadcast message index
2: // Initialize the sets for  $BMSG^i$ 
3:  $RCH^i(r) \leftarrow \phi$ 
4:  $OBLG^i(r) \leftarrow N(s)$ 
5: // Eliminate  $s$  from  $OBLG^i(r)$ 
6:  $RCH^i(r) \leftarrow RCH^i(r) \cup \{s\}$ 
7:  $OBLG^i(r) \leftarrow OBLG^i(r) - \{s\}$ 
8: for each node  $v \in N(r)$  do
9:   if  $GL_r^s[v] = REACHED$  then
10:     $RCH^i(r) \leftarrow RCH^i(r) \cup \{v\}$ 
11:     $OBLG^i(r) \leftarrow OBLG^i(r) - \{v\}$ 
12:   else if  $GL_r^s[v] = OBLIGATED$  then
13:     prepare to work as a new forwarder
14:   end if
15: end for

```

After that, node r acknowledges reception of them by sending a broadcast ACK message i ($BACK^i$) through the same channel. Upon receiving $BACK^i$, forwarder s inserts the ID of node r into $RCH^i(s)$ and removes that from its $OBLG^i(s)$. Forwarder s repeats Step 1–Step 3 until $OBLG^i(s)$ becomes empty.

Step 4. Working as a new forwarder Node r works as a new forwarder to deliver $BMSG^i$ to its neighbor nodes according to guidance provided by forwarder s . The new forwarder r creates new guidance lists and transmits $BMSG^i$ with them only to its neighbor nodes marked as *OBLIGATED*. The set $OBLG^i(r)$ has been determined during the procedure `UPDATE_BROADCAST_STAT()`.

Each guidance list is encoded by a *bitmap* to reduce overhead. Since forwarder s maintains node r 's neighbor information sorted by the node ID ($N(r) = \{v_1, \dots, v_{|N(r)|}\}$), s sequentially encodes only each guidance state of GL_r^s ($GL_r^s[v_1], \dots, GL_r^s[v_{|N(r)|}]$) without specifying neighbor node IDs. Two states are used in the improved forwarder's guidance of M-EMBA (*REACHED* and *OBLIGATED*), so each state is encoded by using only a bit, and the size of GL_r^s is only $|N(r)|$ bits. If a node in a network maintains 8–16 neighbor nodes on average, only 1–2 bytes are required per GL . On the other hand, the forwarder's guidance of EMBA uses two bits to encode three states (*COVERED*, *DELEGATED*, and *OBLIGATED*), so 2–4

bytes are needed per GL . Therefore, the improved forwarder's guidance reduces the size of encoded guidance lists piggy-backed in broadcast messages by half.

Figure 4 shows a simple example of the improved forwarder's guidance. When node S has a broadcast message, node R wakes up and sends a beacon message (B) according to its own schedule. Node S responds to node R with a reply beacon (β) to announce the best channel to R . When node S transmits a $BMSG$ to node R , S obliges R to cover node V if the quality of the best channel between R and V is better than that between S and V . Otherwise, node S assigns $GL_r^s[V]$ to *REACHED* to cover node V itself. If there is no improved forwarder's guidance, nodes S and R simultaneously send $BMSG$ s to node V as soon as they receive a beacon message from V . This causes a collision at node V , and two retransmissions of the same $BMSG$ from S and R . Therefore, the improved forwarder's guidance reduces the number of redundant transmissions and collisions.

Figure 5 shows the operation of M-EMBA in a more complex topology. Node S sends a broadcast message only to nodes $R1$ and $R2$ through Ch_3 and Ch_2 , respectively. Node $R3$ is expected to receive the broadcast message from node $R2$ which has the best channel to $R3$. If nodes $R1$ and $R2$ receive the broadcast message without guidance, two identical copies of the broadcast message will be transmitted to node V by both of them. By the improved

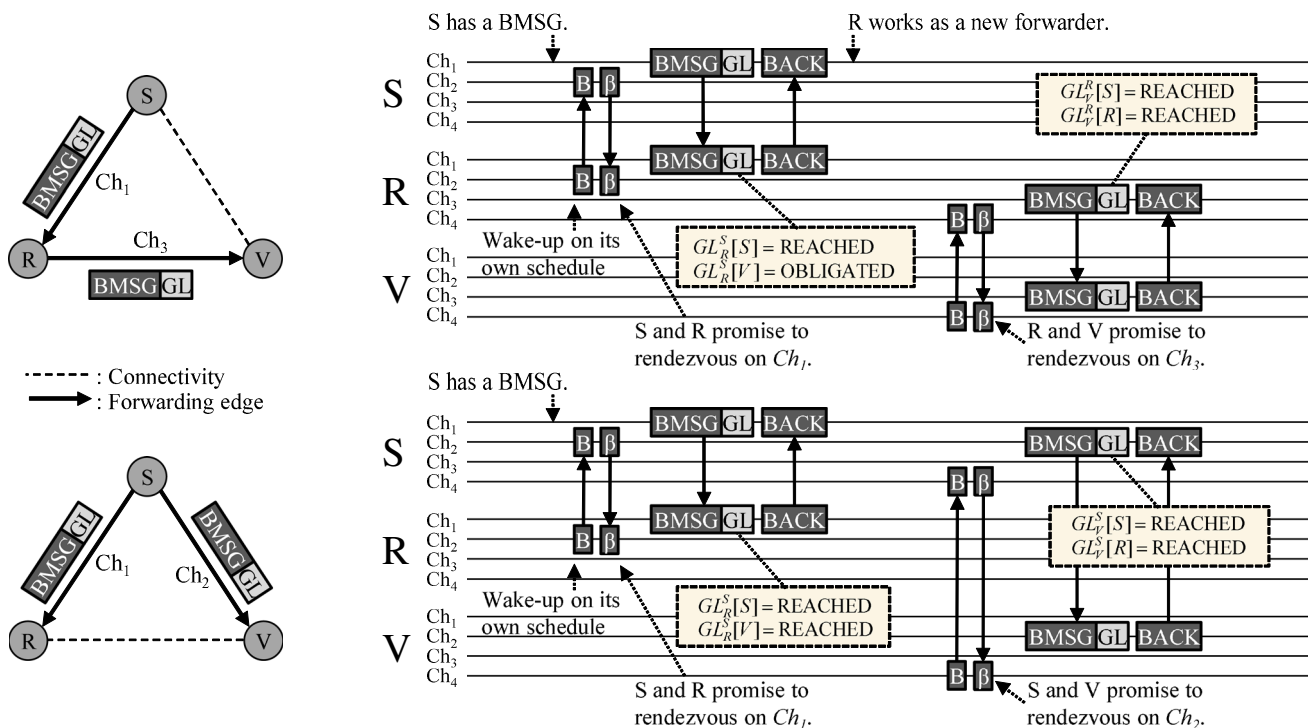


Fig. 4 Operation of the improved forwarder's guidance in a simple network with 4 channels. In this example, M-EMBA works above a receiver-initiated and dynamic channel-assignment-based asynchronous multi-channel MAC protocol

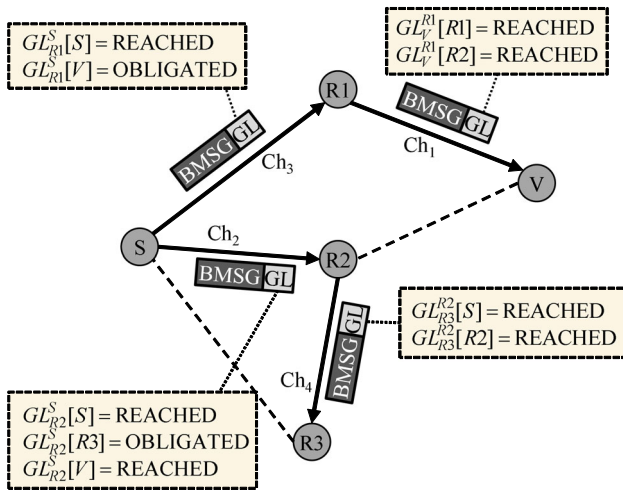


Fig. 5 Operation of the improved forwarder's guidance in a complex network with 4 channels

forwarder's guidance, the broadcast message is transmitted to node V only by node R1 because node S provides $GL_{R1}^S[V] = \text{OBLIGATED}$ only to R1 which has the best channel to V. As a result, multihop broadcast in this topology will be done with the minimum number of transmissions without redundancy and collision.

M-EMBA makes the best effort to transmit broadcast messages on good channels, but messages can be lost either due to collision or network failure. To reliably support multihop broadcast, M-EMBA provides a simple mechanism for resolving network failure. If a BMSG is lost, the forwarder retransmits the BMSG on the same channel after a certain timeout (the forwarder repeats the process again from Step 3). If a BACK message is lost, the forwarder tries to rendezvous with the receiver again (the forwarder repeats the process again from Step 2) because the receiver has returned to its own schedule or prepares to work as a new forwarder.

2.5 Fast forwarding

Fast forwarding is a forwarding strategy which exploits one of the characteristics of multi-channel communications; adjacent nodes can concurrently transmit/receive packets on different channels. This approach enables adjacent forwarders to simultaneously send broadcast messages to their target nodes without any collision. It provides high concurrency of broadcast transmissions that can be exploited to increase spatial reuse. Therefore, fast forwarding helps to increase the multi-channel broadcast utility. In addition, it enables multihop broadcast to finish earlier by reducing multihop broadcast latency, which is the time it takes for one broadcast message to be delivered from a source node to every node in a network.

Algorithm 3 Procedure FAST_FORWARDING()

```

1: // s: forwarder, r: receiver
2: // BMSGi: the i-th broadcast message
3: // vs: next target node of s, vr: next target node of r
4: // ThFF: the threshold value for fast forwarding
5: send BMSGi with GLrs to r
6: // find the best channel by looking into the s' 1-hop table
7: s.best_ch ← ID of the best channel from s to vs
8: // find the best channel by looking into the s' 2-hop table
9: r.best_ch ← ID of the best channel from r to vr
10: if s.best_ch ≠ r.best_ch then
11:   send BMSGi / GLvss to vs on the channel s.best_ch
12: else
13:   s.2nd_best_ch ← ID of the 2nd best channel from s to vs
14:   if CQ(s, vs, s.best_ch) - CQ(s, vs, s.2nd_best_ch) ≤ ThFF
   then
15:     send BMSGi / GLvss to vs on the channel s.2nd_best_ch
16:   else
17:     backoff()
18:     send BMSGi / GLvss to vs on the channel s.best_ch
19:   end if
20: end if
    
```

Algorithm 3 shows the operation of fast forwarding of M-EMBA. Before a new forwarder delivers a broadcast message to its neighbor node, it can predict possible broadcast transmissions within 1-hop and 2-hop distance by examining its neighbor tables. After a forwarder s transmits BMSGⁱ with GL_r^s to a receiver r (line 5), s prepares the next transmission for other neighbor node v_s and r also prepares the transmission for node v_r as a new forwarder. Node s searches for the best channel from itself to node v_s (s.best_ch) and the best channel from node r to node v_r (r.best_ch) by examining its 1-hop and 2-hop neighbor tables (lines 6–9). If nodes s and r choose different channels, they can simultaneously transmit BMSGⁱ to v_s and v_r, respectively (lines 10–11). Otherwise, the probability of collision increases because nodes s and r are adjacent. To solve this problem, forwarder s selects one of the two available methods. If the difference in quality values between the best channel (s.best_ch) and the second-best channel (s.2nd_best_ch) from node s to node v_s is smaller than a threshold value (Th_{FF}), s transmits BMSGⁱ with GL_{v_s}^s to v_s on the second-best channel (lines 13–15). Otherwise, node s covers node v_s after backoff to strategically avoid a collision (lines 17–18).

Figure 6 shows constructed time-coverage graphs of M-EMBA operation with and without fast forwarding. In Fig. 6(a), node B becomes a new forwarder after receiving a BMSG from node A. Node A looks into its 2-hop neighbor table to predict which channel will be used for transmission from node B to node D. If the best channel between nodes A and C is different from that between nodes B and D, A and B can simultaneously send their broadcast messages to C and D, respectively. Multihop broadcast without fast forwarding

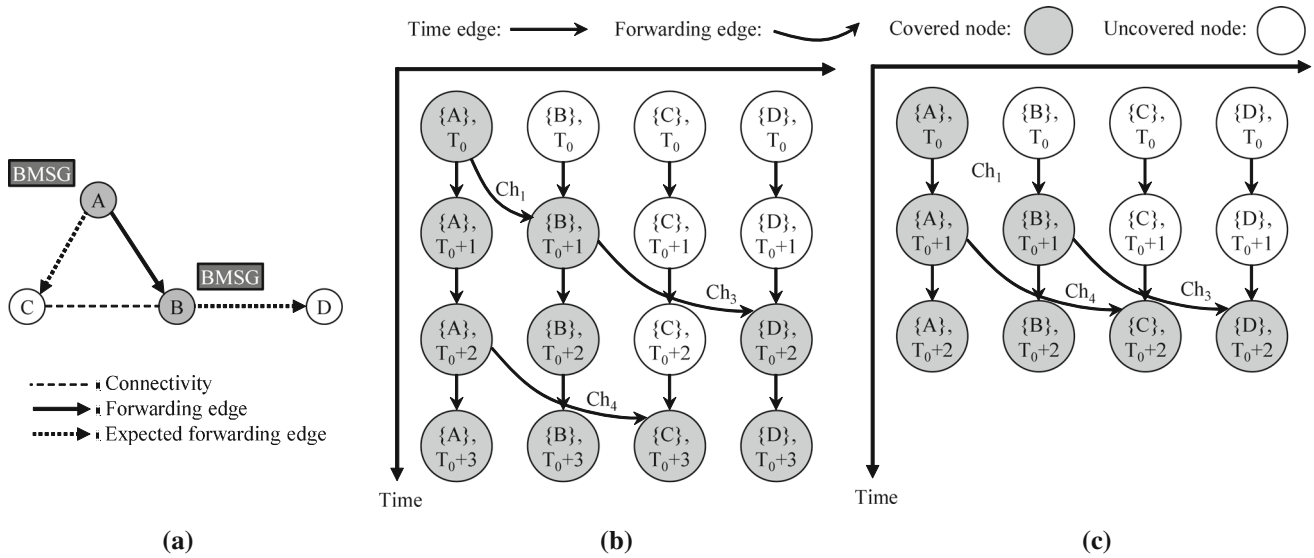


Fig. 6 Comparison of M-EMBA with and without fast forwarding in a simple network with 4 channels. **a** A simple network with 4 nodes. **b** A constructed time-coverage graph without fast forwarding. **c** A constructed time-coverage graph with fast forwarding

finishes in 3 operational cycles (i.e., $T_0 + 3$), as shown in Fig. 6(b). On the other hand, multihop broadcast with fast forwarding is done in 2 operational cycles (i.e., $T_0 + 2$), as shown in Fig. 6(c). Although the difference is only one cycle in this example, it can be larger at the point of view of an entire network that consists of hundreds of nodes.

Fast forwarding reduces multihop broadcast latency and improves multi-channel broadcast utility. Moreover, fast forwarding slightly improves energy efficiency by enabling nodes to complete their deliveries earlier.

3 Discussion of system design

We analyze the bounds of ADV message size, required memory space for channel quality information, guidance list size, and multihop broadcast latency. This gives insight into the practicality and feasibility of M-EMBA in practical asynchronous multi-channel WSNs.

3.1 Analysis of ADV message complexity

Lemma 1 *In a random network with N nodes, the size of an ADV message transmitted by a node is bounded by $\mathcal{O}(N)$.*

Proof We assume that M-EMBA adopts the advertisement method 1 for the advertisement procedure. Each node s periodically transmits an ADV message to its neighbor nodes. The size of an ADV message can be varied according to the frequency of channel quality changes. When M is the number of available channels, the size of each ADV message Θ is calculated as:

$$1 + 2|N(s)| \leq \Theta \leq 1 + (2 + M)|N(s)| \tag{5}$$

where $|N(s)|$ is the number of neighbor nodes of node s and always satisfies $|N(s)| \leq N$. Therefore, the size of each ADV message transmitted by each node is bounded by $\mathcal{O}(N)$ in M-EMBA. \square

Lemma 2 *The size of channel quality information stored in each node is bounded by a constant.*

Proof We suppose that the depth of a network is obtained by counting the number of maximum layers through breadth-first-search (BFS), and this depth is Δ . Each node maintains one- and two-hop neighbor information; thus, each node stores at most $M\Delta^2$ channel quality information when the number of available channels is M . Therefore, the size of channel quality information stored in each node is bounded by a constant. \square

If all nodes in a network are connected with each other, they form a fully connected network. This situation is the worst case of this analysis. Therefore, channel quality information stored by each node scales in $\mathcal{O}(N^2)$ in the worst case.

3.2 Analysis of guidance list complexity

Lemma 3 *In a random network with N nodes, the size of a guidance list piggy-backed in a broadcast message is bounded by $\mathcal{O}(N)$.*

Proof A forwarder s sends a broadcast message with a guidance list that contains an identical number of guidance elements as the number of neighbor nodes of a receiver r . The number of guidance elements is identical with $|N(r)|$

satisfying $|N(r)| \leq N$. Therefore, the size of a guidance list piggy-backed in a broadcast message is bounded by $\mathcal{O}(N)$ in M-EMBA. \square

3.3 Analysis of multihop broadcast latency

We analyze the multihop broadcast latency of M-EMBA in a simplified network model with dynamic channel assignment. We assume this network model is error-free and collision-free. This simple analysis gives an estimation of the upper bound of the multihop broadcast latency in multi-channel environments. In addition, this analysis can be a great help to determine the time interval between successive propagation of broadcast messages. Successive multihop broadcast with a short time interval makes the performance of multihop broadcast worse. To reliably support consecutive multihop broadcast, it is recommended for the next propagation to begin after the previous one is finished.

By Lemmas 1 and 2 in EMBA [18], the upper bound of the multihop broadcast latency in a single-channel network with N nodes is $\max_{0 \leq s < N} \{|OBLG^i(s)|(T_{slp} + T_{tran})\Delta\}$, where $\max_{0 \leq s < N} |OBLG^i(s)|$ is the size of the largest $OBLG^i(s)$ in the network, T_{slp} is the sleep interval, T_{tran} is the maximum time that a broadcast message takes to be transmitted to a neighbor node, and Δ is the network depth which is the maximum layers by breadth-first-search (BFS).

In multi-channel communications, channel switching time cannot be negligible. In dynamic channel assignment approaches, each node switches its channel at every wake-up, so it takes T_{ch} which is the channel switching time. Therefore, when a broadcast message originating from a source node is transmitted to every node in a multi-channel network, the upper bound of the multihop broadcast latency is $\max_{0 \leq s < N} \{|OBLG^i(s)|(T_{slp} + T_{ch} + T_{tran})\Delta\}$.

Lemma 4 *In a random network with N nodes, the multihop broadcast latency is bounded by $\mathcal{O}(N\Delta)$.*

Proof The upper bound of the multihop broadcast latency is at most $\max_{0 \leq s < N} \{|OBLG^i(s)|(T_{slp} + T_{ch} + T_{tran})\Delta\}$. The maximum number of transmissions of $BMSG^i$ for 1-hop broadcast is $\max_{0 \leq s < N} |OBLG^i(s)|$ and always satisfies the following condition:

$$\max_{0 \leq s < N} |OBLG^i(s)| \leq N \quad (6)$$

Therefore, the multihop broadcast latency is bounded by $\mathcal{O}(N\Delta)$. \square

The above mathematical analysis of multihop broadcast latency provides good parameters of the time interval between successive propagation for real deployment or simulation.

4 Performance evaluation

4.1 Simulation environments

To evaluate the performance of our proposed schemes, we implement M-EMBA on top of a receiver-initiated and dynamic channel-assignment-based asynchronous multi-channel MAC protocol by using the *ns-2* simulator. We compare M-EMBA with broadcast schemes of ARM [10], EM-MAC [13], and MuchMAC [14]. In the simulation results, we refer to broadcast schemes of these protocols as *ARM*, *EM-MAC*, and *MuchMAC*, respectively.

ARM is a control-channel-based asynchronous multi-channel MAC protocol, and it provides broadcast support by using a dedicated broadcast channel. To transmit and receive broadcast messages, nodes periodically switch their radio to the broadcast channel. Each forwarder in ARM repeatedly sends broadcast messages without control messages on the broadcast channel for a certain period of time. Hence, the broadcast channel can be easily saturated in dense or burst-traffic networks. EM-MAC is one of the semi-dynamic channel-assignment-based asynchronous multi-channel MAC protocols. In EM-MAC, a source node transmits a broadcast message to its neighbor nodes one-by-one. Similarly, each receiver also relays it to its neighbor nodes except for the sender. MuchMAC adopts the dynamic channel-assignment approach, so each node frequently switches its operating channel. To broadcast, a forwarder repeatedly transmits the same broadcast messages for a certain time as long as the sleep duration of a receiver. EM-MAC and MuchMAC face a large number of redundant transmissions of the same broadcast messages and collisions. ARM, EM-MAC, and MuchMAC broadcast serve as a good baselines to evaluate the performance of M-EMBA in terms of energy consumption and multihop broadcast latency.

We evaluate two variants of M-EMBA: (1) M-EMBA with fast forwarding (referred to here as *M-EMBA*) and (2) M-EMBA without fast forwarding (referred to here as *M-EMBA w/o FF*). In our simulations, we assume that the threshold value Th_{FF} in fast forwarding is 4 levels. A performance comparison of M-EMBA and M-EMBA w/o FF offers a good insight into how efficient the improved forwarder's guidance and fast forwarding can be.

To compare M-EMBA with ARM, EM-MAC, and MuchMAC broadcast, we use network parameters based on the data sheet of CC2420 [24], which is widely used in MICAz and TelosB motes. The transmission range is 250 m as the default channel model in *ns-2*, and the number of available channels is 16. M-EMBA and EM-MAC use the receiver-initiated control messages (beacon), and the sizes of a wake-up beacon and an ACK beacon are 8 bytes and 10 bytes, respectively. If an ACK beacon is a response to a

broadcast message (*BACK*), 2 additional bytes are required to include the broadcast message number. For unicast in ARM, the RTS/CTS mechanism is used, and nodes use two dedicated channels: a control channel and a broadcast channel. MuchMAC adopts a preamble-based approach for unicast. However, both ARM and MuchMAC do not use any control messages for broadcast.

The quality values of channels are represented by 16 priority levels (0–15) in our simulations. The initial quality values of channels are uniformly distributed between 0 and 15 levels at the start of each simulation. To change channel quality over time, we use the shadowing channel model where packet loss occurs. On average, the quality values of half of the available channels are changed every 30 s, so the advertisement procedure in M-EMBA is executed every 30 s. We choose advertisement method 1 to dynamically reflect the changes in channel quality in the performance results. In all protocols, each node stays active for 10 s after booting to build its 1-hop neighbor table.

We perform simulations with various broadcast message generation intervals. The purpose of varying the broadcast message generation interval is to show how efficiently M-EMBA supports multihop broadcast in multi-channel environments with both light and heavy network traffic. In every simulation, 50 nodes are randomly deployed in a network with 250 Kbps bandwidth, and a sink node works as a source node of broadcast messages. The sink node generates 100 broadcast messages periodically with various broadcast message generation intervals of 1, 2, 4, 6, 8, 10, 15, and 20 s. Short intervals (e.g., 1, and 2 s) produce extremely heavy broadcast traffic, and longer ones (e.g., 15 and 20 s) generate light broadcast traffic. Thus, it is meaningful for performance comparisons to vary the broadcast message generation interval. The size of data payload is 28 bytes, and each node sleeps for a uniformly distributed duration between 1 and 2 s. All protocols use a queue to rapidly deal with broadcast messages transmitted by neighbor nodes. Each node has a broadcast message queue, and the maximum size of the queues is 8. Incoming broadcast messages can be dropped when a queue is full.

4.2 Performance metrics for multihop broadcast

To evaluate the performance of multihop broadcast, the following metrics should be considered.

- The number of collisions: The number of collisions should be minimized due to retransmissions caused by collisions of broadcast messages.
- Broadcast redundancy: High broadcast redundancy causes a serious broadcast storm problem. The number of transmissions of duplicate broadcast messages should be reduced.

- Total bytes transmitted: To efficiently support multihop broadcast, nodes need to reduce an amount of the total bytes transmitted. Broadcast redundancy and collision are major factors in increasing the total bytes transmitted.
- Goodput: Throughput is not a good metric because it includes duplicate broadcast messages transmitted by nodes. Therefore, we consider the goodput which is defined by the rate of the total number of unique broadcast messages successfully transmitted to all nodes.
- Broadcast success ratio: To evaluate the reliability of multihop broadcast, the broadcast success ratio should be considered, which means the proportion of broadcast messages successfully delivered to all nodes in a network.
- Multihop broadcast latency: Multihop broadcast latency means the time it takes for one broadcast message to be delivered from a source node to every node in a network.
- Duty cycle: Energy efficiency is the most important design consideration in WSNs. The duty cycle is widely used for evaluation of energy efficiency.

Since multihop broadcast can easily burden a network, it is necessary to evaluate the performance of multihop broadcast in various aspects. Therefore, we perform a thorough simulation study according to these performance metrics.

4.3 Evaluation on collision and broadcast redundancy

Figure 7 shows the average number of collisions when five protocols are used. M-EMBA tremendously reduces the average number of collisions compared to ARM, EM-MAC, and MuchMAC. The number of collisions in M-EMBA is only 20 % of that in ARM, 26 % of that in EM-MAC, and 22 % of that in MuchMAC. The results of

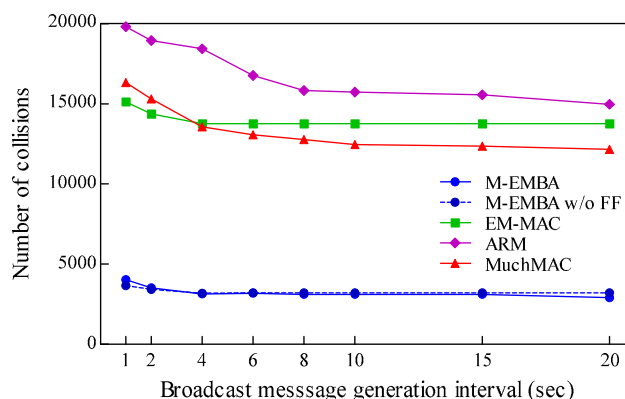


Fig. 7 Average number of collisions

the five protocols slightly decrease as the network traffic becomes lighter. In the broadcast message generation intervals from 4 to 20 s, broadcast messages are periodically delivered to a network through 16 channels, so the network traffic does not strongly impact these results. However, heavy traffic produces a large number of collisions, as shown in the results of the broadcast message generation intervals of 1 and 2 s. In ARM, the number of collisions significantly increases because the broadcast channel used in ARM can be easily saturated by heavy broadcast traffic.

Figure 8 shows the average number of redundant receptions of identical broadcast messages with the five protocols to show their broadcast redundancy performance. Broadcast redundancy with ARM and MuchMAC decreases as the network traffic becomes lighter. In EM-MAC, each forwarder sends broadcast messages to its neighbor nodes one-by-one, and the number of redundant receptions significantly increases in heavy broadcast traffic. The number of redundant receptions in M-EMBA is only 28 % of that in ARM, 20 % of that in EM-MAC, and 26 % of that in MuchMAC.

M-EMBA significantly reduces the average number of redundant transmissions of the same broadcast messages and the number of collisions in both light and heavy network traffic and it leads to higher energy efficiency. However, the difference between M-EMBA and M-EMBA w/o FF is negligible in the results shown in both Figs. 7 and 8. Therefore, fast forwarding does not considerably affect the number of collisions or the number of redundant transmissions of the same broadcast messages.

4.4 Evaluation on Message Cost

Figure 9 shows the average number of bytes transmitted by M-EMBA, M-EMBA w/o FF, ARM, EM-MAC, and MuchMAC. The metric of the average number of bytes transmitted indicates the byte-usage to deliver 100 broadcast messages to all nodes, so the use of fewer bytes indicates better performance. These results of the five protocols include all types of messages, such as advertisement messages, broadcast messages, and control messages. ARM and MuchMAC show better performance than EM-MAC because they use no control packet during broadcast. M-EMBA improves the number of bytes by about 15–42 % compared to ARM, by about 25–51 % compared to EM-MAC, by about 13–41 % compared to MuchMAC, and by about 2–7 % compared to M-EMBA w/o FF. The average number of bytes transmitted by M-EMBA and M-EMBA w/o FF increases with an almost constant slope. In our simulations, every node frequently executes an advertisement procedure (every 30 s) to reflect

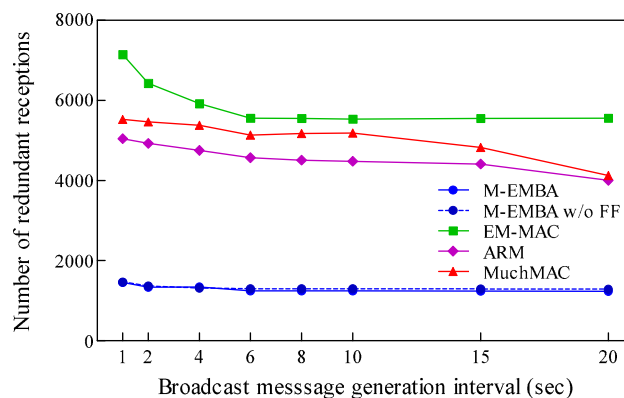


Fig. 8 Average number of redundant receptions of the same broadcast messages

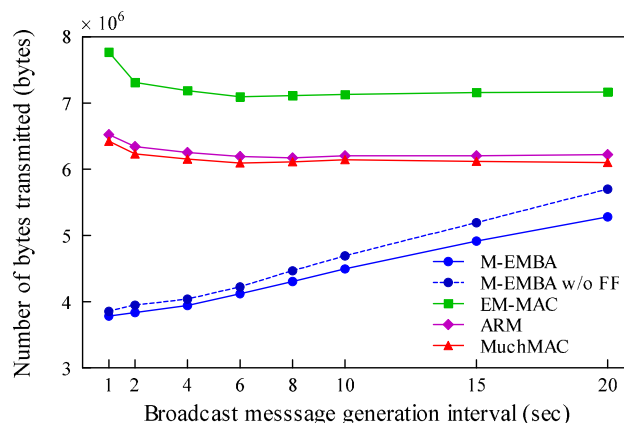


Fig. 9 Average number of bytes transmitted

frequent changes in topology or channel quality. If a network is static, the advertisement rate can be adjusted (e.g., 60 or 90 s).

Figure 10 shows the goodput of the five protocols. M-EMBA significantly improves the goodput by about 88–25 % in comparison to ARM, by about 39–45 % in comparison to EM-MAC, and by about 88–97 % in comparison to MuchMAC. As shown in the graph, fast forwarding improves the goodput of M-EMBA by about 13–32 %. Consequently, M-EMBA achieves high multi-channel broadcast utility by using fast forwarding and it helps to improve the goodput.

4.5 Evaluation on reliability

Figure 11 shows the results of the broadcast success ratio. If the n -th broadcast message does not arrive at one or more nodes in a network, we regard the n -th broadcast as a failure. The broadcast success ratio is defined by the following equation:

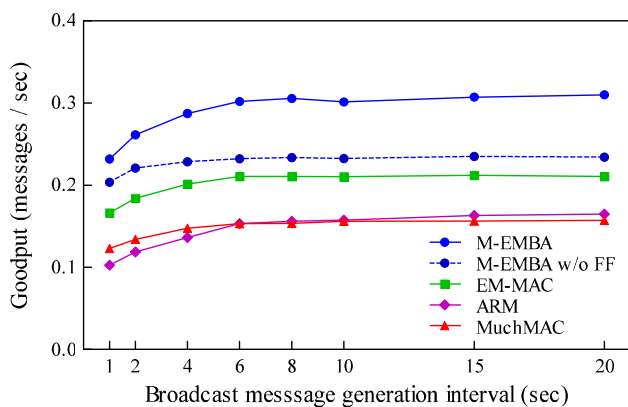


Fig. 10 Goodput

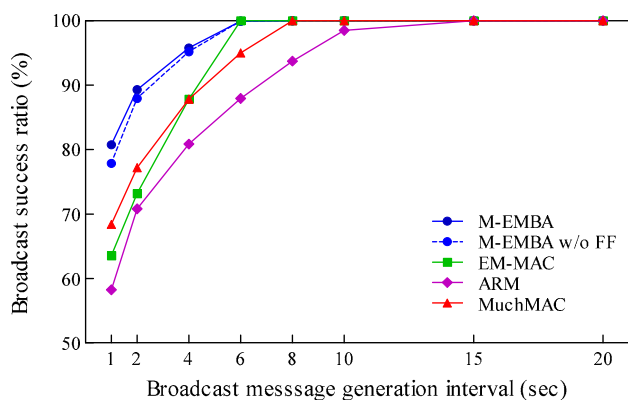


Fig. 11 Broadcast success ratio

$$1 - \frac{\text{Failed messages}}{\text{Messages generated by sink}} \tag{7}$$

The results of the five protocols show almost 100 % from broadcast message generation interval 10–20. However, the broadcast success ratio of each protocol significantly decreases in extremely heavy broadcast traffic (i.e., broadcast message generation intervals of 1 and 2). These results may be influenced by the size of the broadcast message queue, which is set to 8 in our simulations. If the size of a queue increases, the broadcast success ratio of each protocol will increase somewhat.

ARM shows the worst performance in terms of the broadcast success ratio. The broadcast channel in ARM can be easily saturated during multihop broadcast because adjacent forwarders compete for the broadcast channel to transmit broadcast messages. M-EMBA and M-EMBA w/o FF show better performance than ARM, EM-MAC, and MuchMAC. M-EMBA improves the broadcast success ratio by about 22 % compared to ARM, by about 17 % compared to EM-MAC, and by about 12 % compared to MuchMAC. The difference between M-EMBA and M-EMBA w/o FF is slight. Consequently, M-EMBA

achieves a higher broadcast success ratio compared to other protocols in heavy network traffic.

4.6 Evaluation on multihop broadcast latency

The multihop broadcast latency is obtained by the following equation:

$$\frac{\sum \text{Multihop broadcast latency of message}}{\text{Messages generated by source}} \tag{8}$$

The results of the multihop broadcast latency do not include the broadcast message generation interval. Figure 12 shows the average multihop broadcast latency of M-EMBA, M-EMBA w/o FF, ARM, EM-MAC, and MuchMAC. The results of all protocols do not considerably increase or decrease with the broadcast message generation intervals from 6 to 20 because a broadcast message generated from a source node is delivered according to the duty cycle schedule of each node. To propagate a broadcast message in asynchronous multi-channel networks, each forwarder must wait on a certain channel until an intended receiver wakes up. This process is repeatedly executed in every hop due to the different wake-up schedules of individual node. On the other hand, the multihop broadcast latency of each protocol significantly increases in heavy broadcast traffic because a large number of collisions occur in the broadcast message generation intervals of 1, 2, and 4 s.

Among M-EMBA and ARM, EM-MAC, and MuchMAC, M-EMBA achieves better performance. The multihop broadcast latency in M-EMBA is reduced by about 56 % compared to that in ARM, by about 28 % compared to that in EM-MAC, and by about 47 % compared to that in MuchMAC. Comparing M-EMBA and to M-EMBA w/o FF, M-EMBA reduces the multihop broadcast latency by about 23 % compared to M-EMBA w/o FF. Fast forwarding increases the multi-channel broadcast utility and has a positive impact on performance in terms of multihop broadcast latency in multi-channel communications.

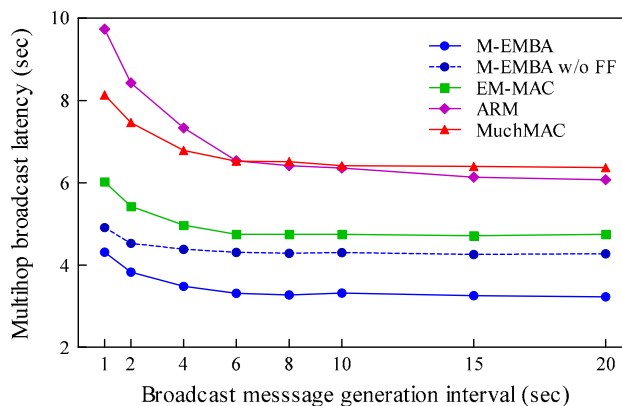


Fig. 12 Average multihop broadcast latency

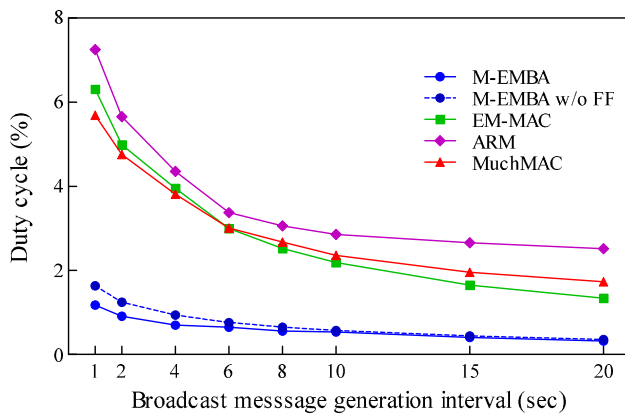


Fig. 13 Average duty cycle

4.7 Evaluation on energy efficiency

Figure 13 shows the average duty cycle per node in M-EMBA, M-EMBA w/o FF, ARM, EM-MAC, and MuchMAC. These results include all energy-consuming activities, such as broadcast message transmission, control message exchange, and advertisement procedures. The results of all protocols decrease as the network traffic becomes lighter. M-EMBA reduces the average duty cycle by about 84–87 % compared to ARM, by about 76–81 % compared to EM-MAC, and by about 79–82 % compared to MuchMAC in light and heavy broadcast traffic. Unlike other protocols, the energy efficiency of M-EMBA is less influenced by network traffic because M-EMBA significantly reduces the number of redundant transmissions and collisions. In addition, fast forwarding slightly improves the average duty cycle by about 10–28 % by exploiting multi-channel broadcast utility. Since fast forwarding helps nodes to finish their delivery earlier, nodes are able to save power. Consequently, we conclude that M-EMBA outperforms the existing protocols in multihop broadcast in terms of energy efficiency represented by the average duty cycle.

5 Conclusions

In this paper, we have proposed an efficient multihop broadcast protocol for asynchronous multi-channel WSNs (M-EMBA). M-EMBA employs two channel-quality-aware techniques of improved forwarder's guidance and fast forwarding to efficiently support multihop broadcast. The improved forwarder's guidance substantially reduces the number of redundant transmissions of identical broadcast messages and the number of collisions in both light and heavy broadcast traffic, which significantly improves energy efficiency. Fast forwarding helps nodes finish their delivery of broadcast messages earlier by exploiting multi-

channel broadcast utility, so this simple technique reduces multihop broadcast latency and improves goodput. Through the theoretical analysis of the system design, we have analyzed our M-EMBA from the viewpoint of practical system design, and the analysis have shown that our scheme can be implemented with acceptable demands in terms of memory and processing power. We have also performed a thorough simulation-based analysis with various performance metrics of multihop broadcast. We have compared the performance of our M-EMBA with that of existing protocols of broadcast schemes of ARM, EM-MAC, MuchMAC in the *ns-2* simulator. The result have shown that our scheme achieves higher energy efficiency than other protocols in terms of the duty cycle and improves the broadcast success ratio and multihop broadcast latency. It also improves message cost in terms of the average number of redundant transmissions and collisions, the average number of bytes transmitted, and the goodput. Therefore, we have concluded that our M-EMBA is the most reliable and energy-efficient protocol for multihop broadcast in asynchronous multi-channel environments and it can be practically implemented while considering the limited battery and computing power of sensor nodes in WSNs.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2014R1A2A2-A01006957) and IT R&D program of MKE/KEIT [10041244, SmartTV 2.0 Software Platform].

References

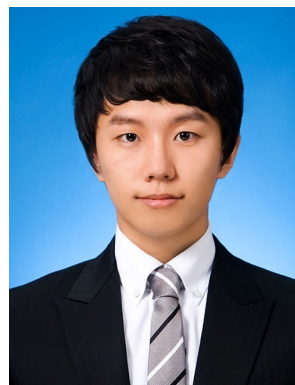
- Sadagopan, N., Krishnamachari, B., & Helmy, A. (2005). Active query forwarding in sensor networks. *Ad Hoc Networks*, 3(1), 91–113.
- Ma, M., Yang, Y., & Ma, C. (2006). Single-path flooding chain routing in mobile wireless networks. *International Journal of Sensor Networks*, 1(1), 11–19.
- Hui, J. W., & Culler, D. (2004). The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems*, (pp. 81–94).
- Kulkarni, S. S., & Wang, L. (2005). MNP: Multihop network reprogramming service for sensor networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, ICDCS 2005*, (pp. 7–16).
- Tseng, Y. C., Ni, S. Y., Chen, Y. S., & Sheu, J. P. (2002). The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2–3), 153–167.
- Zhou, G., Huang, C., Yan, T., He, T., Stankovic, J. A., & Abdelzaher, T. F. (2006). MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In *Proceedings of IEEE INFOCOM*, (PP. 1–13).
- Incel, O. D., van 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.
- Kim, Y., Shin, H., & Cha, H. (2008). Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks.

In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks, IEEE Computer Society*, (pp. 53–63).

9. Ramakrishnan, M., & Ranjan, P. V. (2009). Multi channel MAC for wireless sensor networks. *International Journal of Computer Networks & Communications (IJCNC)*, 1(2), 47–54.
10. Li, J., Zhang, D., Guo, L., Ji, S., & Li, Y. (2010). ARM: An asynchronous receiver-initiated multichannel MAC protocol with duty cycling for WSNs. In *Performance Computing and Communications Conference (IPCCC), 2010 IEEE 29th International* (pp. 114–121).
11. Yuanyuan, Z., Xiong, N., Park, J. H., & Yang, L. T. (2012). An interference-aware multichannel media access control protocol for wireless sensor networks. *The Journal of Supercomputing*, 60(3), 437–460.
12. Chowdhury, K. R., Nandiraju, N., Cavalcanti, D., & Agrawal, D. P. (2006). CMAC-A multi-channel energy efficient MAC for wireless sensor networks. In *Wireless Communications and Networking Conference, IEEE WCNC*, (pp. 1172–1177).
13. Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011). EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 20th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, (p. 23).
14. Borms, J., Steenhaut, K., & Lemmens, B. (2010). Low-overhead dynamic multi-channel mac for wireless sensor networks. In J. S. Silva, B. Krishnamachari & F. Boavida (Eds.), *Wireless Sensor Networks* (pp. 81–96). Berlin Heidelberg: Springer.
15. Sun, Y., Gurewitz, O., Du, S., Tang, L., & Johnson, D. B. (2009). ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, (pp. 43–56).
16. Zhao, D., Chin, K. W., & Raad, R. (2014). Minimizing broadcast latency and redundancy in asynchronous wireless sensor networks. *Wireless networks*, 20(3), 345–360.
17. Wang, F., & Liu, J. (2009). Duty-cycle-aware broadcast in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, (pp. 468–476).
18. 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.
19. Xing, G., Sha, M., Huang, J., Zhou, G., Wang, X., & Liu, S. (2009). Multi-channel interference measurement and modeling in low-power wireless networks. In *30th IEEE Real-Time Systems Symposium. RTSS*, (pp. 248–257).
20. Toscano, E., & Lo Bello, L. (2008). Cross-channel interference in IEEE 802.15. 4 networks. In *IEEE International Workshop on Factory Communication Systems, WFCS*, (pp. 139–148).
21. Incel, O. D. (2011). A survey on multi-channel communication in wireless sensor networks. *Computer Networks*, 55(13), 3081–3099.
22. Soua, R., & Minet, P. (2015). Multichannel assignment protocols in wireless sensor networks: A comprehensive survey. *Pervasive and Mobile Computing*, 16, 2–21.
23. IEEE 802.11. (1999). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. *ANSI/IEEE Std. 802.11*.
24. Chipcon Inc. CC2420 data sheet. <http://www.chipcon.com/>.



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.



broadcast protocols, for energy efficiency in wireless sensor networks and for efficient spectrum utilization in cognitive radio networks.



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, Indian Hill, IL. Since 1989, he has been a professor with the department of computer science in Korea Advanced Institute of Science and Technology. His research interests include mobile ad hoc networks, wireless networks, and network security.

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

Dohoo Pyeon received his B.S. degree in the college of information and communications at Hanyang University, Seoul, Korea, in 2011, and his M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2013. He is currently towards Ph.D. degree in computer science from KAIST. His research interests include the communication protocols, such as medium access control, routing, and

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.



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.

design and optimization of LTE (Long Term Evolution)-advanced