

Research Article

Analysis and Enhancement of IEEE 802.15.4e DSME Beacon Scheduling Model

Kwang-il Hwang and Sung-wook Nam

Department of Embedded Systems Engineering, Incheon National University, Incheon 402-772, Republic of Korea

Correspondence should be addressed to Kwang-il Hwang; hkwangil@incheon.ac.kr

Received 23 January 2014; Accepted 6 April 2014; Published 6 May 2014

Academic Editor: Young-Sik Jeong

Copyright © 2014 K.-i. Hwang and S.-w. Nam. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to construct a successful Internet of things (IoT), reliable network construction and maintenance in a sensor domain should be supported. However, IEEE 802.15.4, which is the most representative wireless standard for IoT, still has problems in constructing a large-scale sensor network, such as beacon collision. To overcome some problems in IEEE 802.15.4, the 15.4e task group proposed various different modes of operation. Particularly, the IEEE 802.15.4e deterministic and synchronous multichannel extension (DSME) mode presents a novel scheduling model to solve beacon collision problems. However, the DSME model specified in the 15.4e draft does not present a concrete design model but a conceptual abstract model. Therefore, in this paper we introduce a DSME beacon scheduling model and present a concrete design model. Furthermore, validity and performance of DSME are evaluated through experiments. Based on experiment results, we analyze the problems and limitations of DSME, present solutions step by step, and finally propose an enhanced DSME beacon scheduling model. Through additional experiments, we prove the performance superiority of enhanced DSME.

1. Introduction

Growing concerns about machine-to-machine communications, such as sensor networks and the Internet of things (IoT), have accelerated the development of a low-power, low-rate, and low-cost wireless system. In particular, IEEE 802.15.4 [1] has become a representative standard for low-rate wireless personal area network (LR-WPAN) communications. One of the main advantages in using IEEE 802.15.4 is low-power operation, which is accomplished by a beacon-enabled mode. However, the beacon-enabled mode requires all devices in the network to be synchronized with a pan coordinator (PC) and this PC manages a superframe to maintain active durations and inactive durations within beacon intervals. This requirement also makes it difficult for a WPAN to be extended to multihop networks [2–4]. One of the critical problems is collision among beacon frames transmitted by different devices.

Therefore, beacon scheduling to address the beacon collision problem has been considered one of the significant

challenges in multihop networks comprising IEEE 802.15.4 systems. In order to address the beacon collision problem, various beacon scheduling methods have been studied, so far. Beacon scheduling can be largely categorized into the tree-based approach and the mesh-based approach. The tree-based category includes various beacon scheduling approaches [5–11] based on a tree network topology. The tree network has the advantage of low routing overhead, but it is prone to partial network isolation from link failure in an intermediate node. On the other hand, a mesh network can provide more flexible topology management, and thus some research [12–16] emphasizes the importance of beacon scheduling for mesh networks. In particular, an IEEE 802.15.4 task group (TG4e) realized the need for beacon scheduling in mesh networks, and they evolved an efficient beacon scheduling model utilizing a specific bitmap for neighboring superframe duration slot management in the IEEE 802.15.4e draft [15], more specifically a deterministic and synchronous multichannel extension (DSME) capability. However, the draft provides only a concept, so concrete algorithm details

TABLE 1: Tree-based beacon scheduling.

Title (author)	Characteristics
Cho and An [5]	Proposes a clustering approach based on group management using transmission power control
TG4b [6]	Proposes two approaches: (i) using a dedicated beacon-only period in a superframe (ii) compensating beacon offset between parent and child
Koubâa et al. [7]	Proposes time division beacon scheduling based on the neighbors' location and connection information
Ahn et al. [8]	Proposes a slot allocation method coupled with the Cskip address allocation scheme introduced in ZigBee
Yeh et al. [9]	Utilizes separate uplink and downlink slots
Yen et al. [10]	Proposes stochastic beacon scheduling in order to reuse beacon slots in the network
Chen et al. [11]	Proposes application specific beacon scheduling for sensor networks (i) controls the beacon interval adaptively according to varying the target detection level

TABLE 2: Mesh-based beacon scheduling.

Title (author)	Characteristics	Remarks
Burda and Wietfeld [12]	Proposes a random beacon slot assignment to construct autonomous and distributed mesh networks using a beacon-enabled mode.	(i) Requires a long time delay to complete an integrity check. (ii) Has high collision probability among request packets
IEEE 802.15.5 [13]	Mesh network based on IEEE 802.15.4 (i) characterizes block address assignment (ii) utilizes a connection matrix	Asynchronous energy saving in non-beacon-enabled mode might create unnecessary energy waste and delay
MeshMAC [14]	Distributed beacon scheduling for a mesh network	Does not address how to collect a neighboring beacon slot
IEEE P802.15.4e DSME [15]	Provides a concept level for bitmap-based distributed beacon scheduling	(i) Does not address SD slot selection methods (ii) Correctness of the algorithm is not yet verified via experiments
DFBS [16]	Distributed, fast beacon scheduling for a mesh network (i) bitmap-free beacon scheduling	(i) Proposing a conceptual model (ii) Similar to IEEE 802.15.4e DSME except for using an indicator instead of a bitmap

for implementation as well as validity of the algorithm through experiments have not yet been provided.

In this paper, through experiments, we evaluate the validity of the DSME beacon scheduling model specified in the IEEE 802.15.4e draft and propose a concrete design model. The experiment was conducted using ns-3 [17], a popular open source network simulator, and we found some problems in the pure DSME beacon scheduling algorithm by analyzing the experiment results. Therefore, based on the results, we revised the pure DSME beacon scheduling algorithm step by step and now propose enhanced DSME beacon scheduling, including limited permission notification coupled with a proper superframe duration slot-selection method.

The remainder of this paper is organized as follows. Section 2 discusses various beacon scheduling algorithms in multihop networks. In Section 3, we present an overview of IEEE 802.15.4e (an enhanced draft of IEEE 802.15.4). Experimental study of the DSME beacon scheduling is presented in Section 4. Section 5 introduces the enhanced DSME beacon scheduling, and performance is evaluated in Section 6. Finally, Section 7 provides concluding remarks.

2. Related Work

Over the past few years, several researchers have made numerous attempts to construct scalable multihop WPANs

based on a beacon-enabled mode and having the advantage of energy efficiency. Major challenges in a multihop extension are synchronization and collision avoidance. In order to address these two problems, various beacon scheduling methodologies have been proposed. The beacon scheduling algorithm can also be divided into two categories according to topology: tree-based beacon scheduling and mesh-based beacon scheduling.

Even though research on tree-based beacon scheduling as presented in Table 1 resulted in various attempts to solve the beacon collision problem for multihop-enabled WPANs, all of these methods only focus on the tree topology, and thus it is impossible to apply them to another topology, such as a mesh structure. The entire tree structure might be reconstructed if a communication link failure on the path of the tree occurs. Moreover, interference with other communications might occur because each node manages only its uplink and downlinks. These problems restrict utilization of a multihop low-power WPAN in more applications. Table 2 presents research on mesh-based beacon scheduling.

3. Overview of IEEE 802.15.4e DSME

As an enhanced version of IEEE 802.15.4, IEEE 802.15.4e includes new network structures and functionalities to meet a variety of application requirements in LR-WPANs. To

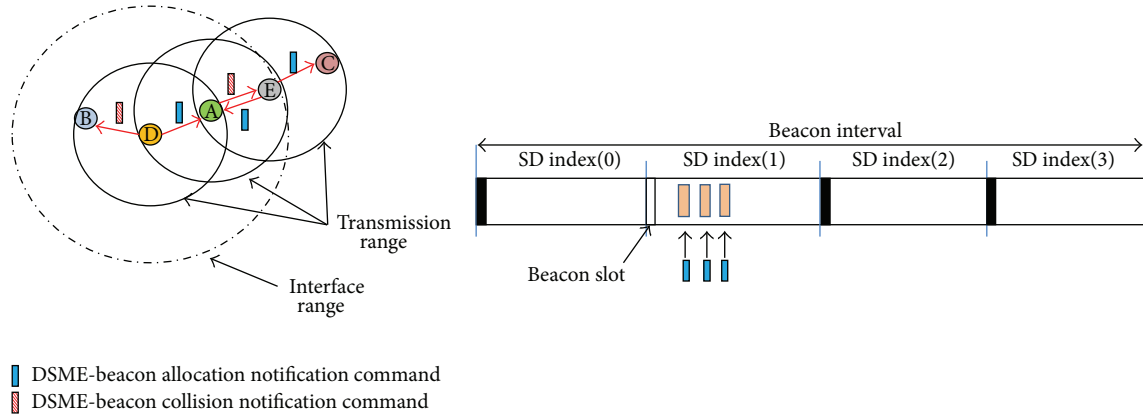


FIGURE 1: An example of collision avoidance during SD slot allocation in IEEE 802.15.4e DSME.

accommodate these requirements, IEEE 802.15.4e provides five different types of mode as follows:

- (i) TSCH: time-slotted channel hopping for high throughput requirements,
- (ii) LLDN: low latency deterministic network for high reliability and low latency,
- (iii) DSME: deterministic and synchronous multichannel extension for deterministic latency and scalability requirements,
- (iv) RFID: radio frequency identification (e.g., Blink) for item and people identification, location, and tracking,
- (v) AMCA: asynchronous multichannel adaptation for infrastructure monitoring networks.

From the above modes, DSME involves a novel beacon scheduling for mesh networks. Even though DSME supports multichannel utilization, the multichannel operation is limited only in a contention-free period (CFP) for guaranteed time slot (GTS) usage. Actual superframe operation is achieved using a single channel, so DSME focuses on avoiding beacon collisions among different WPANs by introducing a beacon scheduling method in which different pan coordinators in a complex mesh network are synchronized by conducting beacon scheduling based on a multisuperframe structure, which allows a number of superframes to coexist in a beacon interval (BI). In DSME beacon scheduling, each prospective device (more specifically, the pan coordinator) first performs a scan procedure over the available channels. Each DSME device has a superframe duration (SD) index table to manage the SD information of neighboring nodes. In addition, the SD index information of a node is represented as a bitmap, included in a macSDBitmap field of the beacon frame, which is transmitted periodically to notify neighbors about current SD index allocation information. If a prospective node receives a beacon of an active node indicating that an SD index is already allocated, the node selects a vacant slot, which is represented as "0" in the received macSDBitmap, sets the corresponding bit to "1", and broadcasts a DSME beacon-allocation notification command frame to its neighbors.

The neighboring nodes that receive the notification command first check if the bit is being used by other neighboring nodes and then they update their SD index table if the slot is available.

However, in the SD index allocation process mentioned above, a collision might occur when more than two devices make an attempt to occupy the same slot. Figure 1 illustrates this beacon collision situation. When nodes D and E, which are neighbors of node A but cannot communicate with each other, receive a beacon from node A, both can select the same slot out of vacant slots in the SDBitmap (the hidden-node problem). In that case, the two nodes have the same SD index, so the beacon transmission slot overlaps. That is, since the beacons of the two nodes collide, node A cannot hear either beacon transmitted from the two nodes. To address this problem, DSME uses an additional frame, a DSME beacon-collision notification command. If the two nodes want to use the same SD index by sending an allocation notification message, node A allows the node arriving first to allocate the SD slot, and if another node then requests the already occupied slot, node A makes the new requester select another slot by sending the collision notification command. Eventually, this procedure can avoid overlapped allocation of SD indexes among neighboring nodes. This method provides a simple but powerful beacon scheduling, which is not solved in IEEE 802.15.4. In particular, it is possible for a superframe duration of two-hop neighboring nodes as well as neighboring nodes to be scheduled in a distributed manner.

4. Experiment with IEEE 802.15.4e DSME Beacon Scheduling

IEEE 802.15.4e DSME beacon scheduling, as described in the previous section, can provide efficient scheduling among neighboring WPANs. However, the standard introduces just an abstract concept without any concrete outline and implementation details. Therefore, in this section we first present a guideline for implementation of DSME beacon scheduling and then evaluate the performance and validity of the algorithm.

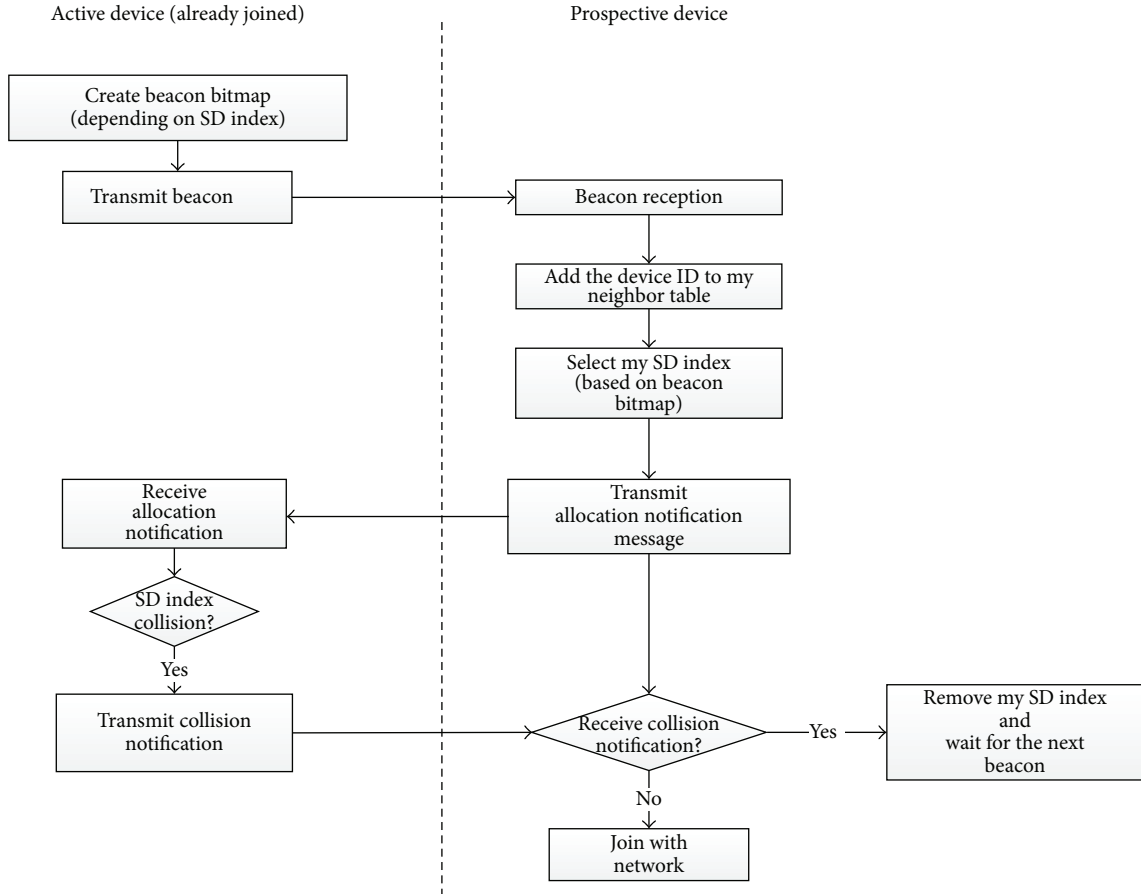


FIGURE 2: Flowchart of the IEEE 802.15.4e DSME beacon scheduling algorithm.

4.1. Experiment Environment. We implemented beacon scheduling for IEEE 802.15.4 DSME based on the LR-WPAN module of ns-3 [17]. For a channel model, a single model spectrum channel was used and a communication channel was constructed based on a long distance propagation loss error model. Specific primitives and frames for DSME are involved in the LR-WPAN media access control (MAC) layer and additional PAN information base (PIB) attributes and constants are defined in the MAC header. In addition, instead of a superframe structure used in the original IEEE 802.15.4, a multisuperframe structure (a specific feature of DSME) is used.

4.2. Pure DSME

4.2.1. DSME Beacon Scheduling Review. This subsection presents the DSME beacon scheduling algorithm in terms of implementation. Figure 2 shows a concrete flow chart of a DSME beacon scheduling algorithm. The algorithm is divided into two stages: the active node stage and the prospective node stage. First, an active node generates a beacon bitmap based on its own SD index and the SD index information of its neighboring nodes. This SDBitmap information is contained in a beacon frame broadcasted periodically. All nodes update their neighboring SD index table whenever

they receive a beacon frame containing an SDBitmap field. In particular, if a prospective node receives a beacon frame, it adds the sender ID to the SD index table entry and updates SD index information based on the SDBitmap contained in the received beacon. Then, a prospective node selects an available (vacant) slot and broadcasts a beacon allocation notification message. All the nodes that receive the allocation notification message check whether the notified slot is already used. If the slot is available, the SD index table is updated by adding the designated slot. However, if the slot is already used, the node that receives the notification message sends a collision notification message to the originator to prevent overlapped SD slot usage. As soon as the prospective node, which sent notification of its new SD index, receives a collision notification message, it clears the SD index and then waits until the next beacon reception. After broadcasting the allocation notification message, if a collision notification message is not received by the end of the contention access period (CAP) of the present superframe duration, the prospective node becomes an active node with the allocated SD index, and it can send its periodic beacon in the allocated SD index slot.

4.2.2. IEEE 802.15.4e DSME SD Slot Selection. If a prospective node receives a beacon, a superframe slot of the node can be allocated by selecting a vacant slot. The allocated slots are

(Variables)**A[]:** BITMAP ARRAY**sA:** Size of bitmap data type(eg., unsigned char A[BITMAP_ARRAY_SIZE],
in that case we have sA = 8)**index:** indicator of array index**position:** bit position in a A[x],
where $x = 0, 1, 2, \dots, \text{BITMAP_ARRAY_SIZE}$ **LAB (A)**

```

(1) index ← 0
(2) for i ← 0 to BITMAP_ARRAY_SIZE
(3)   for j ← 0 to sA - 1
(4)     if (A[i] & (0 × 01 << j)) == 0
(5)       SD_index ← j + (sA * i)
(6)       return SD_index
(7)     j ← j + 1
(8)   i ← i + 1
(9) return NO_AVAILABLE_SD

```

MAB (A)

```

(1) index ← (BITMAP_ARRAY_SIZE - 1)/sA
(2) position ← (BITMAP_ARRAY_SIZE - 1) mod sA
(3) if (A[index] & (0 × 01 << position)) == 1
(4)   return NO_AVAILABLE_SD
(5) for i ← index to 0
(6)   for j ← 0 to sA - 1
(7)     if (A[i] & ((0 × 01 << sA - 1) >> j)) == 0
(8)       SD_index ← sA - j + (j * sA)
(9)       return SD_index
(10)    j ← j + 1
(11)  i ← i - 1
(12) return NO_AVAILABLE_SD

```

RAND_SD (A)

```

(1) do
(2)   r_sd ← random(1.MAX_SD)
(3)   index ← r_sd/sA
(4)   position ← r_sd mod sA
(5) while (A[index] & (0 × 01 << position)) == 1
(6) SD_index ← position + index * sA
(7) return SD_index

```

ALGORITHM 1: Possible SD slot selection algorithms.

represented as “1” in the SDBitmap so that the node chooses a slot from the “0” bits. As a matter of fact, since the allocation method is distributive and the SDBitmap only represents the SD index of the neighboring sender, slot allocation distribution in the received SDBitmap at any time varies according to network topology. Therefore, to make a rule for selecting an SD index from vacant slots in a bitmap, we consider three different selection methods: least available bit (LAB), most available bit (MAB), and random. However, we also presume that the beacon scheduling performance might vary according to the selection method used.

Algorithm 1 shows each of the three possible SD slot selection algorithms: LAB, MAB, and random. LAB searches the first “0” bit for the received bitmap from the least significant bit. The first “0” bit finally becomes its own beacon slot number. On the other hand, MAB searches the first “1”

bit for the received bitmap from the most significant bit. The “0” bit followed by the first “1” becomes its own beacon slot number. Random method randomly chooses one number and then the selected number is used if the corresponding bit is clear in the bitmap. Otherwise, the random selection process is repeated until the selected bit is clear.

The LAB selection method is to choose the lowest slot number out of the vacant slots. This method might increase the reuse ratio of the superframe duration among the nodes that are separated by more than two hops. However, there is the possibility that collisions will occur during the SD index selection process.

The MAB selection method is to choose the vacant bit that immediately follows the largest value of the allocated slot numbers. This method may provide a lower reuse ratio of the SD index than LAB, so the possibility of collision might be reduced. In addition, with more hops, there is a greater possibility of allocating an SD index in order.

The random method chooses a vacant slot at random. Random SD index selection might increase unnecessary network traffic to avoid collisions because of the possibility that different nodes select the same slot.

4.2.3. Experiment Results. In this subsection, through the experiments we verify the validity of DSME and evaluate its performance with respect to different SD index slot selection methods. For the experiments, we considered two representative topologies: sparse and dense models, consisting of 3×3 nodes as shown in Figure 3.

To evaluate the performance of DSME beacon scheduling, we first observed the successful SD slot allocation ratio with respect to different SD slot selection methods (LAB, MAB, and random) in sparse and dense topology models, as shown in Figure 4. The result shows that the MAB selection method is superior to LAB and random. In particular, the LAB selection method shows the worst performance. That is because LAB caused a number of collisions among the nodes that chose the same SD slot, as we expected. However, the result also reveals that even MAB, which shows the best performance among them, shows an allocation failure ratio of more than 20 percent in the dense topology. This results from a beacon collision problem that is not yet completely resolved.

Through experiments, we also found limitations of pure DSME in some specific situations. The problem is mainly caused by collisions of command messages transmitted from different nodes at the same time. In particular, the more complex the topology and the more the devices, the higher the collision possibility. Figure 5 illustrates an example of collision among allocation notification messages during an SD allocation phase. As shown in Figure 5(a), nodes 1, 2, 4, and 5 have completed SD index allocation and, as a result, slots 3, 4, and 5 are allocated for nodes 2, 4, and 5, respectively. After finishing the superframe duration of node 1, node 5 (which has slot 2) transmits its beacon, as shown in Figure 5(b). Already activated nodes 1, 2, and 4 just update their neighbor SD index table. On the other hand, the remaining nodes where the SD index is not yet allocated select an available SD index based on the received

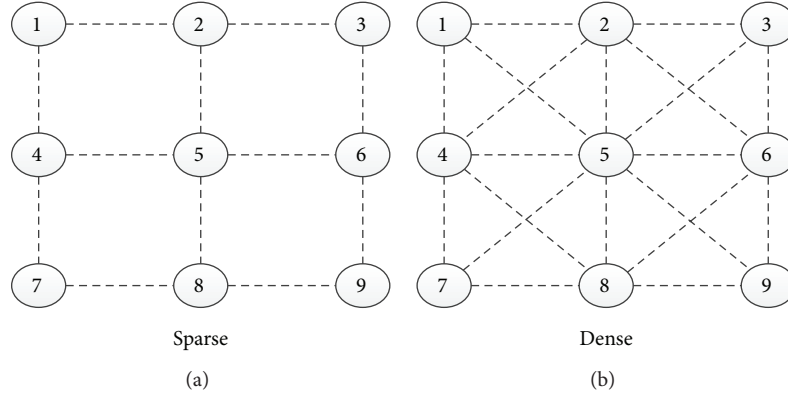


FIGURE 3: Experiment topologies: sparse and dense models.

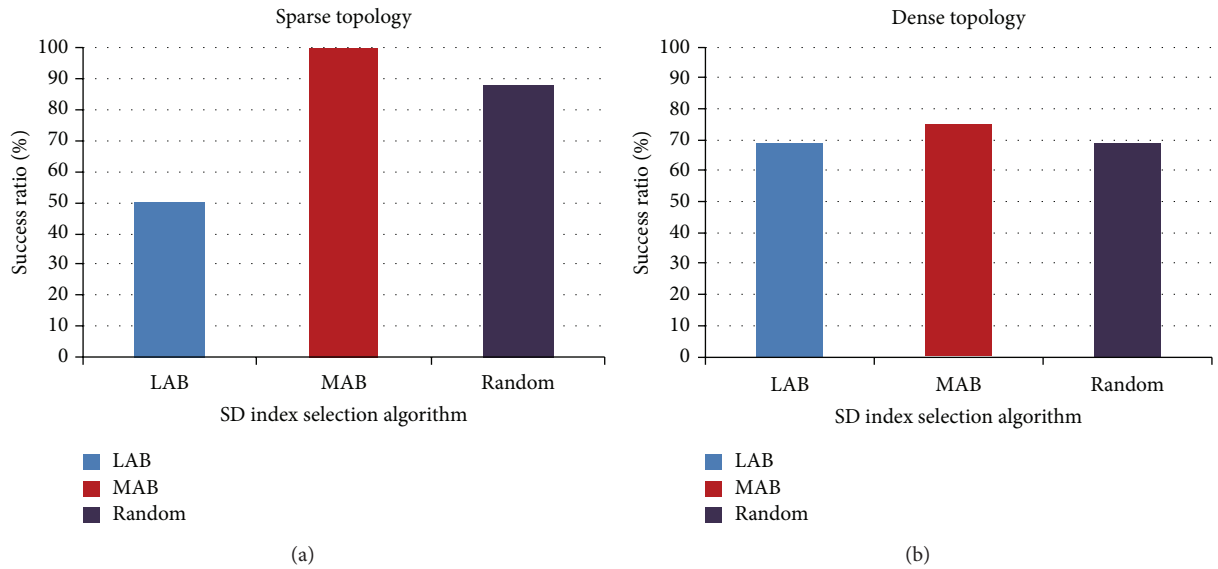


FIGURE 4: Experiment results of pure DSME beacon scheduling.

SDBitmap. Since the nodes receive the same SDBitmap and use the same SD selection algorithm, they choose the same vacant slot and make an attempt to transmit their own allocation notification message using carrier sense multiple access/collision avoidance (CSMA/CA). Even though they transmit their own messages using CSMA/CA, collisions still occur, as shown in Figure 5(c), and thus the nodes that are not acknowledged by node 5 retry sending the message after waiting for random back-off delay. However, since the random back-off duration is lengthened more whenever collision occurs, eventually the present superframe duration being maintained by node 5 is not completed and some nodes remain unallocated. That is, when the topology is complex and the number of nodes increases, the collision possibility for command messages increases. This results in SD index allocation failure in some nodes. Furthermore, if acknowledgement of each command frame is disabled, after transmitting its notification message, each node is convinced that its selected SD slot is available and transmits its beacon

on the allocated SD slot. In that case, the beacon frames of the nodes might collide with each other, as shown in Figure 5(d).

The major reason for these collisions is the hidden node problem, in which each node cannot identify the presence of other nodes. Therefore, even though every node performs CSMA before a transmission, allocation notification messages of others might not be detected. Furthermore, the back-off effect with respect to the same beacon frame results in an increased collision possibility among messages. To observe the effect of hidden node problems in a realistic environment, we conducted an experiment on successful data ratios with respect to varying the number of hidden nodes, as shown in Figure 6. The result shows that a 100% success ratio is not guaranteed, even among two hidden nodes, and performance deteriorates drastically as the number of hidden nodes increases.

4.3. Distributed Permission Notification. The experiment results revealed that the DSME beacon scheduling algorithm

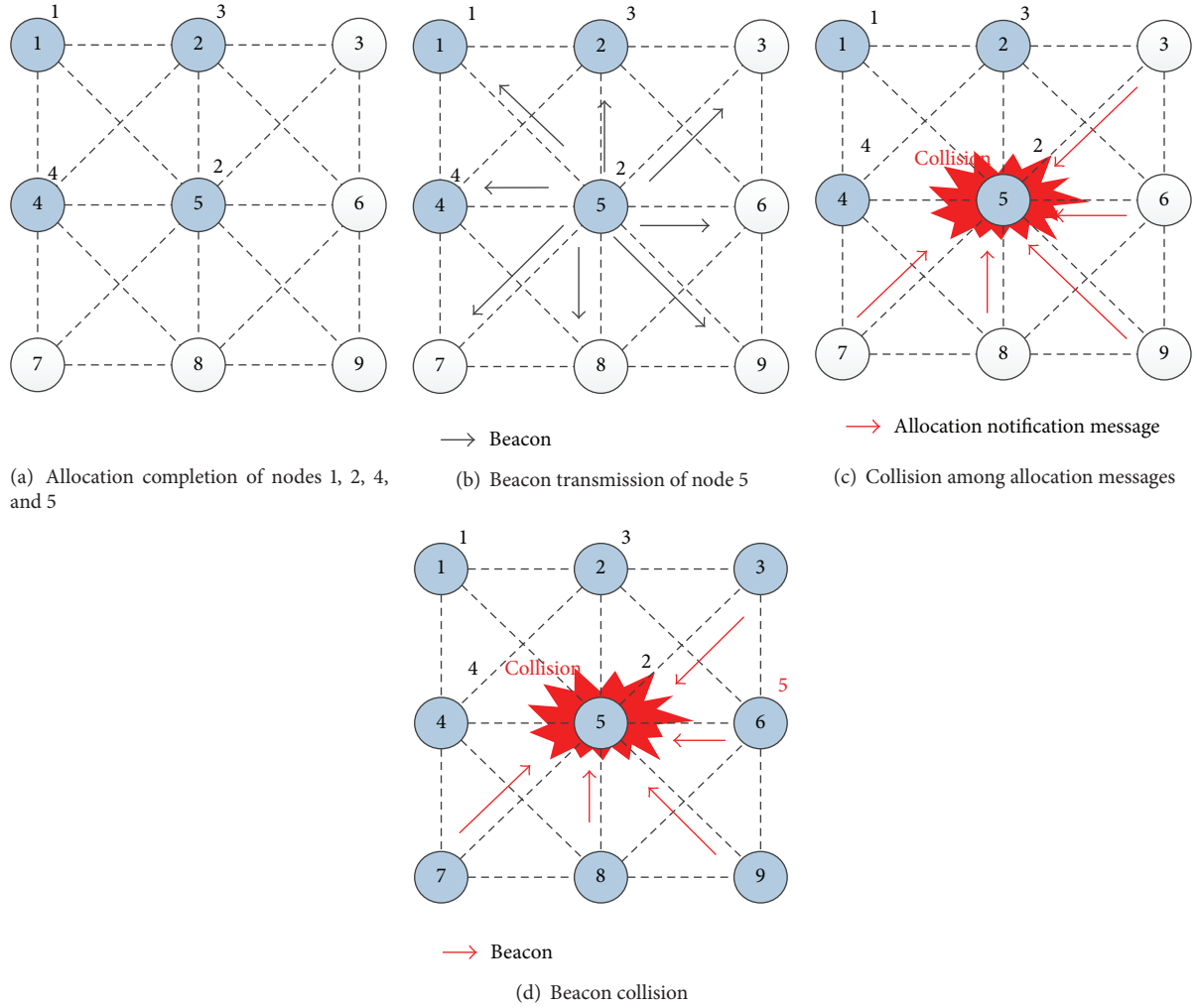


FIGURE 5: An example of collision among allocation notification messages.

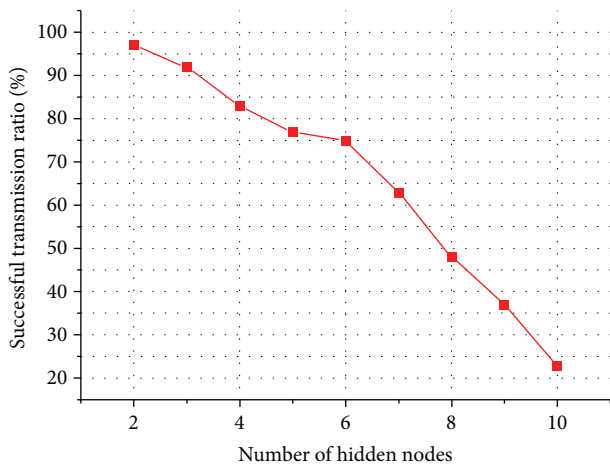


FIGURE 6: Number of hidden nodes versus packet success ratio.

has some critical problems. Therefore, it is necessary to rework the pure DSME beacon scheduling algorithm by

resolving collisions. One of the significant problems in pure DSME is collisions between command frames, such as allocation notification and collision notification, and another is beacon collision that is caused by using overlapped SD slots. This is because a node allocates its SD index slot for itself based on the received SDBitmap information.

So, we first employ a distributed permission notification to enhance collision notification specified in pure DSME. While pure DSME uses a negative allocation by sending a collision notification only when the newly allocated SD slot overlaps with another neighbor's, the distributed permission method uses positive allocation by allowing only the node that receives a permission notification message after sending an allocation notification message to complete the SD index allocation. A prospective node that sent an allocation notification message waits until permission notification is received, and the neighboring active node that receives allocation notification from the prospective node checks whether the requested SD index is available or not, and if the slot is available, it broadcasts a permission notification. The neighbors of the active node that broadcast permission

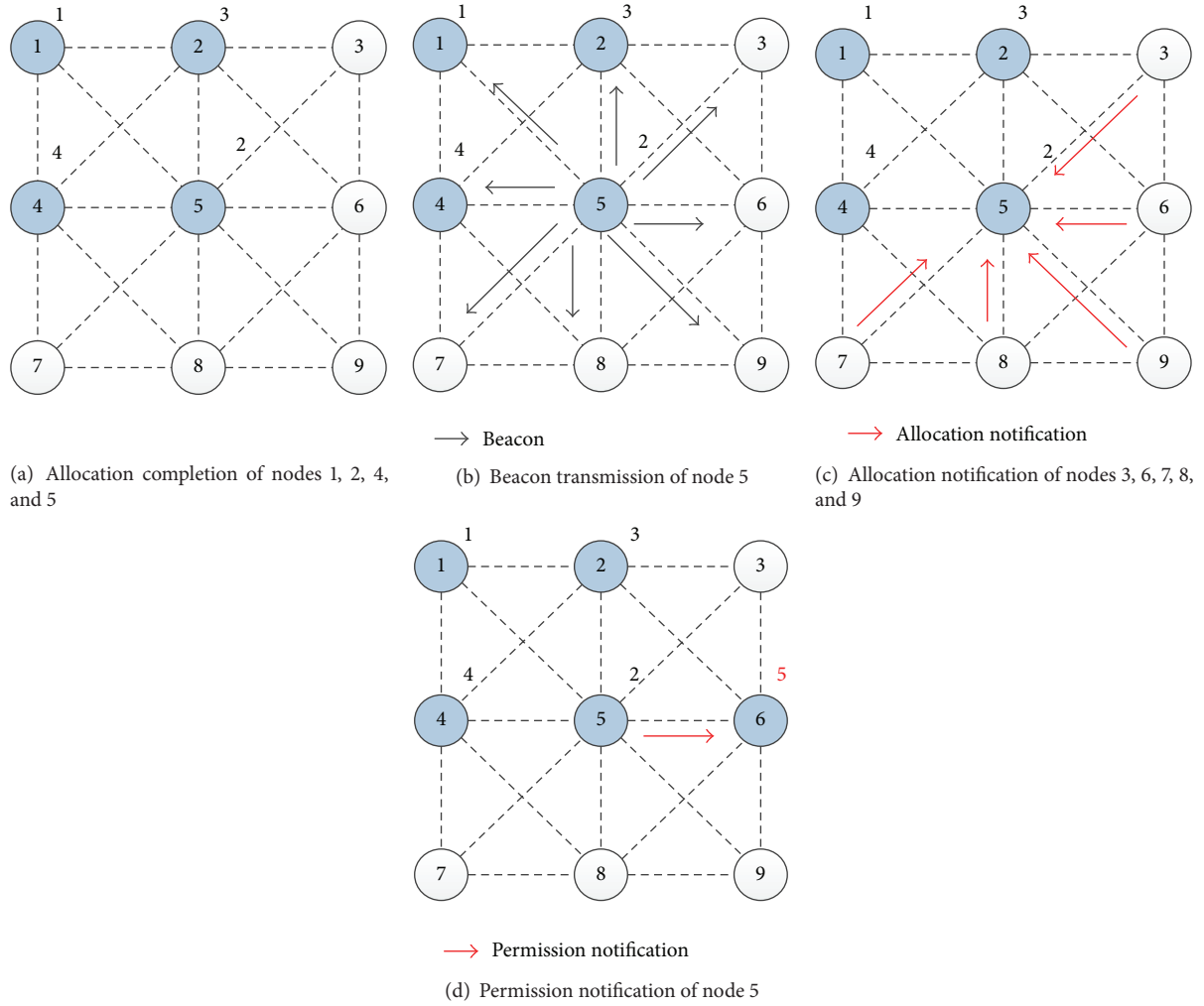


FIGURE 7: An example of distributed permission notification method.

notifications also come to know the information of the SD index of a new node separated by two hops by hearing this permission notification. This might result in the reduction of collision. Furthermore, it is even expected to cope well with a complex topology. Figure 7 shows an example of a distributed permission method. As in the previous example, the nodes that receive the beacon from node 5 send their allocation notification message to node 5. However, unlike pure DSME, where there have been collisions among the allocation notification messages, the permission method can avoid the collisions by only permitting the allocation of node 6, which is selected by node 5.

This enhancement is expected to significantly improve performance of DSME. However, contrary to our expectations, the experiment results were worse than pure DSME. As shown in Figure 8, the successful beacon scheduling ratio shows lower values in both sparse and dense models.

Figure 9 illustrates the main reason for the performance degradation. The distributed permission method can avoid collisions by learning the SD index information of two-hop nodes through permission notification, but all the nodes

that receive an allocation notification message have the right to send a permission notification, and thus, as shown in Figure 9, the nodes that do not have to be allocated also complete SD allocation. Therefore, they allocate the same SD slot, and eventually beacon collision occurs. The collisions also occur regardless of the SD index selection method used.

In addition to that reason, there is the possibility that the permission notification messages of active nodes might collide with the allocation notifications of the prospective nodes, so that the neighboring active nodes often miss the SD index information of new neighboring nodes.

5. Enhanced DSME Beacon Scheduling

5.1. DSME Experiment Results Review. IEEE 802.15.4e DSME beacon scheduling presents a method that can minimize beacon collisions efficiently using a multiple superframe structure in mesh-based multihop networks. However, experiment results demonstrated that the DSME beacon scheduling model is still in its conceptual stage and needs to be enhanced in order to apply it to various topology models

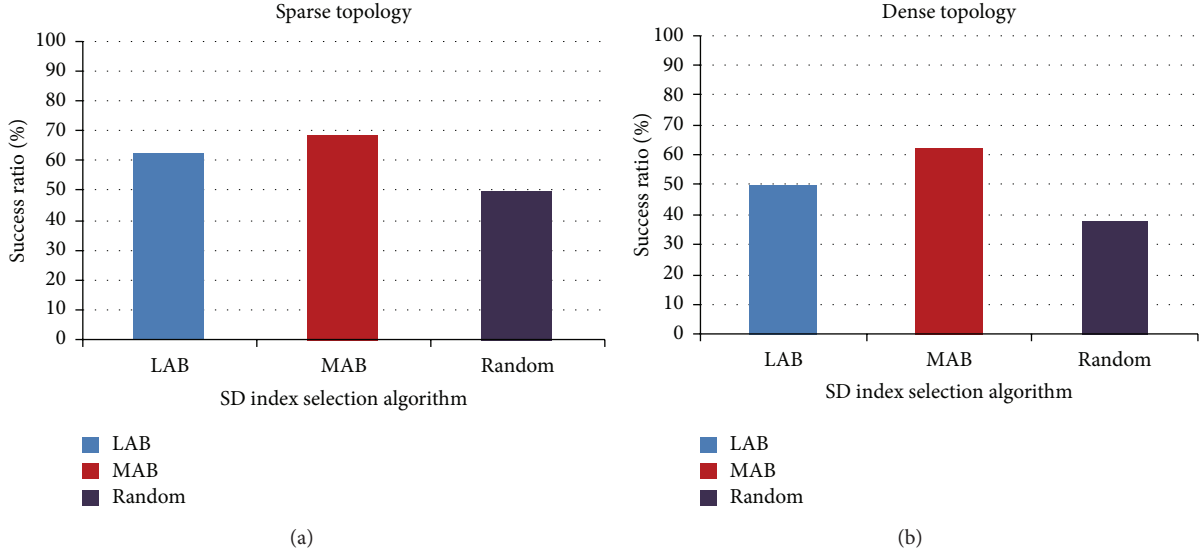


FIGURE 8: Experiment results of the distributed permission method.

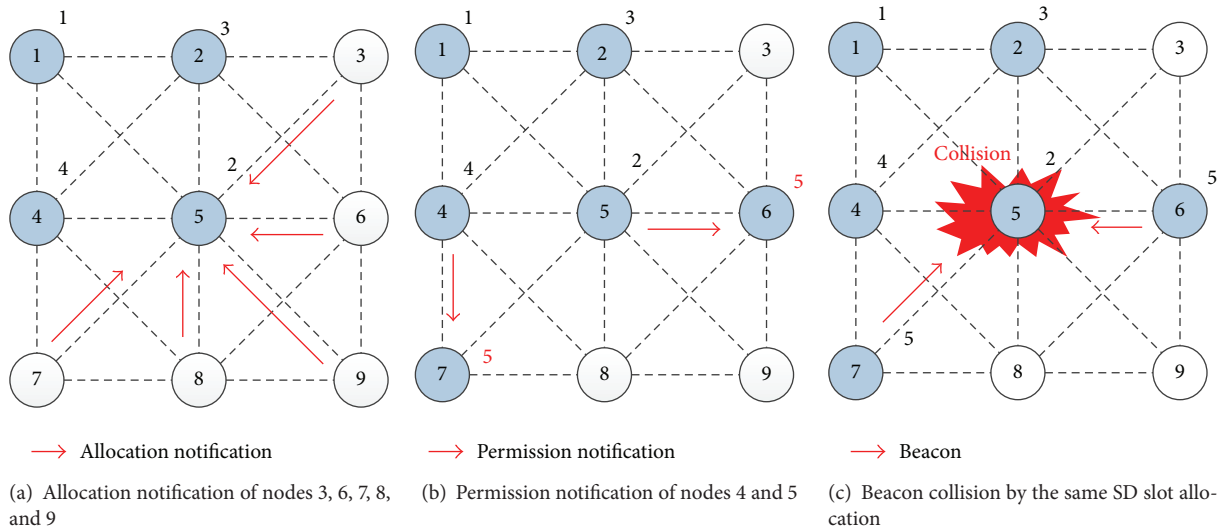


FIGURE 9: Main reason for performance degradation: distributed permission method.

and environments. In the previous section, we analyzed the DSME beacon scheduling model based on experiment results and proposed the first revision in which we tried to resolve command frame collision problems caused by absence of information about nodes separated by two hops. Even though our first revision can solve the problems caused in pure DSME, overall performance was not improved because it has another collision problem. Therefore, in this section, we present a new revision of DSME beacon scheduling to cope well with several problems that are not solved so far and we verify the algorithm's correctness and performance.

5.2. Limited Permission Method. The distinguished enhancements of enhanced DSME (E-DSME) are the use of limited permissions and a new superframe structure suitable for

distributed beacon scheduling. First, a permission right is only limited to the originator of the latest beacon, as shown in Figure 10. Restricting the node that can send permission to the sender of the beacon can avoid abuse of SD allocation caused by unnecessary permission notifications from neighboring active nodes. In addition to this, the superframe is restructured, as shown in Figure 11. A superframe duration is composed of a number of SD allocation durations (SAD), which also consist of an allocation contention period (ACP) and a permission notification period (PNP). ACP is a period in which prospective nodes that receive a beacon of a parent PC assign their own candidate SD index and transmit an allocation notification to the parent PC through contention. A permission notification of a parent PC is allowed only during the PNP. This is to separate transmission timings

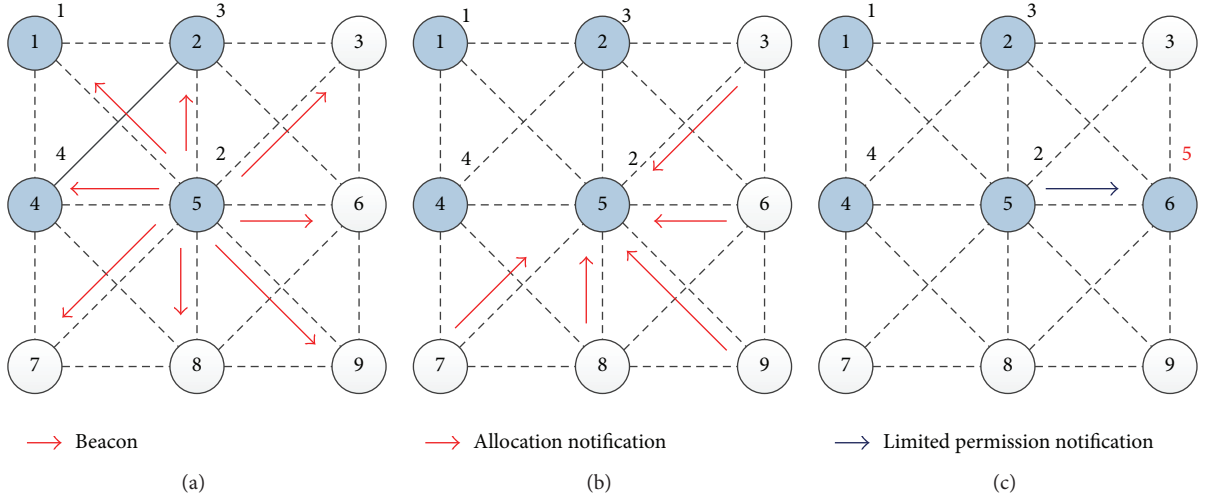


FIGURE 10: Limited permission notification.

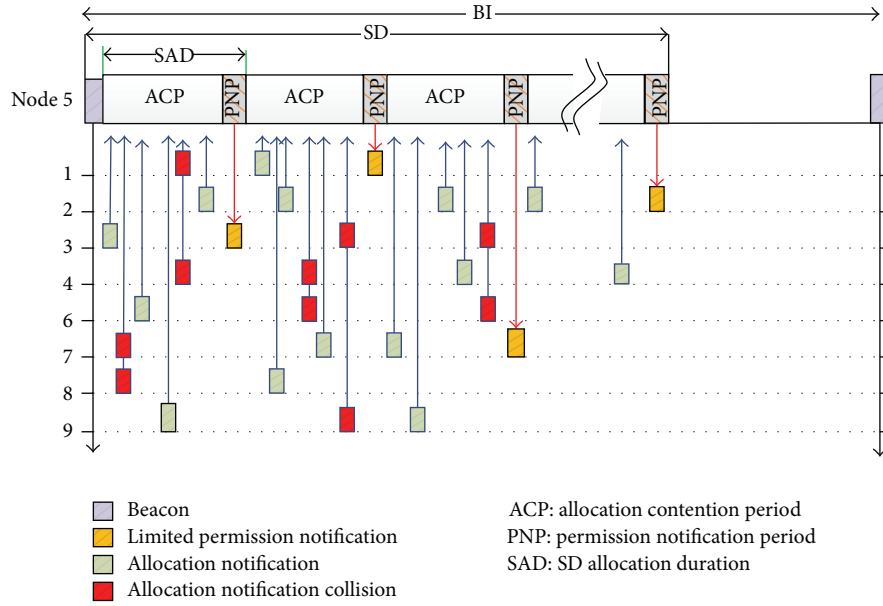


FIGURE 11: E-DSME superframe structure.

between allocation notification and permission notification messages. The repetitive architecture of SAD within a superframe duration enables a node that did not have permission during the first SAD to update its candidate SD index by itself and retry allocation notification with the updated SD index. Therefore, it is possible for more prospective nodes to join the network within a superframe duration.

5.3. E-DSME Superframe Structure. As mentioned in the previous subsection, a superframe duration is made up of a number of SADs. A SAD is also composed of ACPs and PNPs:

$$\text{SAD} = T_{\text{ACP}} + T_{\text{PNP}}. \quad (1)$$

Here, each duration of T_{ACP} and T_{PNP} is as follows:

$$\begin{aligned}
 T_{\text{ACP}} &= \left[\left\{ \left(\sum_{i=0}^{\text{macMaxBE}} 2^{\text{maxMinBE}+i} \right) \times \text{aUnitBackoffPeriod} \right\} \right. \\
 &\quad \left. + \text{aBaseSlotDuration} \right] \times (\text{Symbol rate})^{-1}, \quad (2) \\
 T_{\text{PNP}} &= \frac{\text{aBaseSlotDuration}}{\text{Symbol rate}}.
 \end{aligned}$$

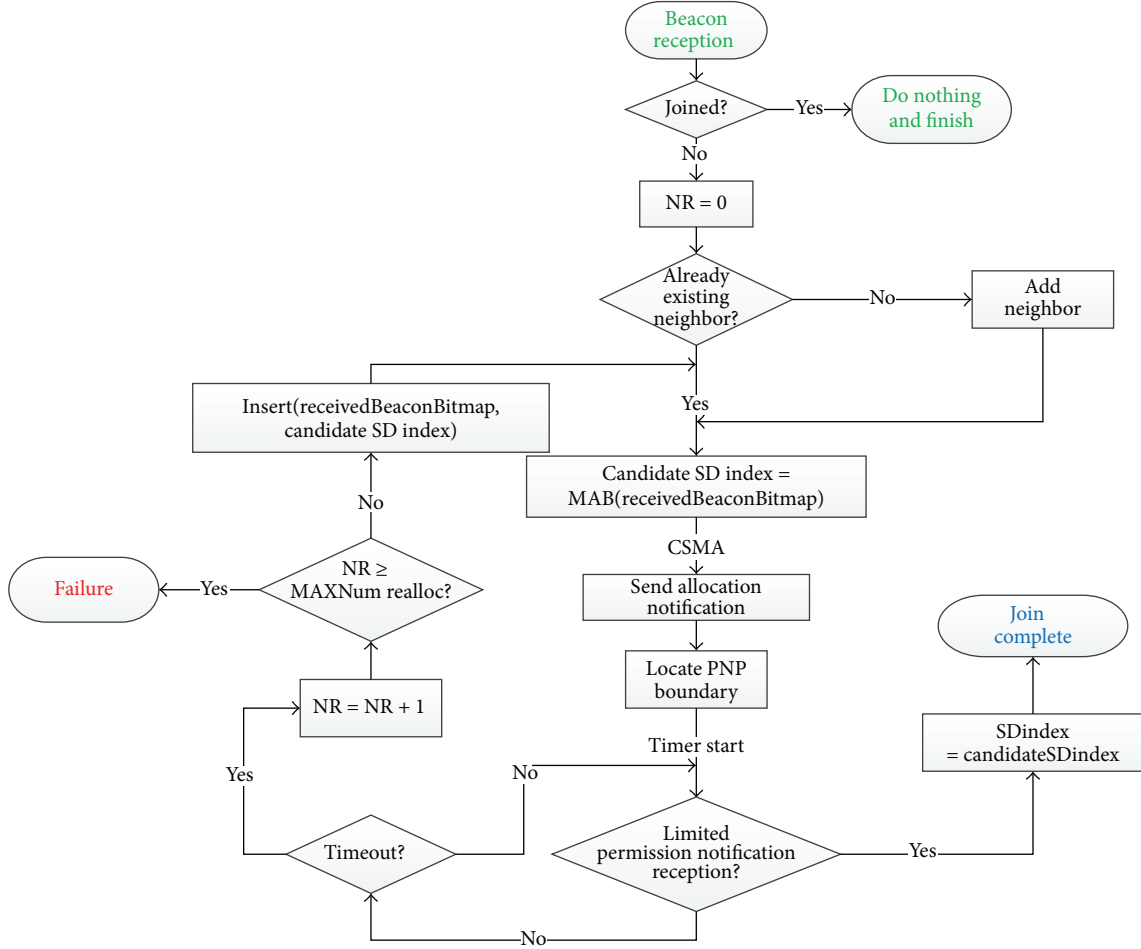


FIGURE 12: E-DSME beacon scheduling.

In addition, the maximum number of SADs in a superframe is obtained as follows:

MaxNumSAD

$$= \min \left\{ \frac{SD}{SAD}, \text{macMaxBeaconBitmapSize} - \text{CandidateSDindex} \right\}, \quad (3)$$

where $SD = \text{aBaseSuperframeDuration} \times 2^{SO}$, $0 \leq SO \leq 14$ and $\text{aBaseSuperframeDuration} = \text{aBaseSlotDuration} + \text{aNumSuperframesSlots}$.

5.4. E-DSME Beacon Scheduling. Figure 12 shows a beacon scheduling flowchart for E-DSME. Upon the reception of a beacon frame from a neighbor, a prospective device first checks whether it has already joined or not. In addition, the originator of the received beacon is not registered in neighbor table, the information is updated, and then the prospective node selects a candidate SD index from the received bitmap using MAB slot selection method. To avoid collisions, the node waits for the upcoming PNP after transmitting allocation notification request. If, during the PNP, there is no

permission, the node retries allocation notification request at the next ACP. If a permission notification is received from the originator, the node registers current candidate SD index slot for its own SD index slot and then completes the join procedure. The outstanding feature of E-DSME beacon scheduling is that, according to success or failure, a prospective node can update its own candidate SD index by itself and perform an allocation notification procedure repeatedly.

5.5. Algorithm Verification. Testing enhanced DSME was conducted in the same environment as previous experiments. As shown in Figure 13, enhanced DSME shows a 100% allocation success ratio when the MAB SD index selection method is applied in both the sparse and the dense models. Compared to the previous method (pure DSME and distributed permission notification), performance improvement with enhanced DSME beacon scheduling might result from utilizing a limited permission notification and repetitive SAD structure. Furthermore, the result demonstrates that the MAB SD index selection algorithm is the most suitable for DSME beacon scheduling compared to the other SD index selection methods, LAB and random.

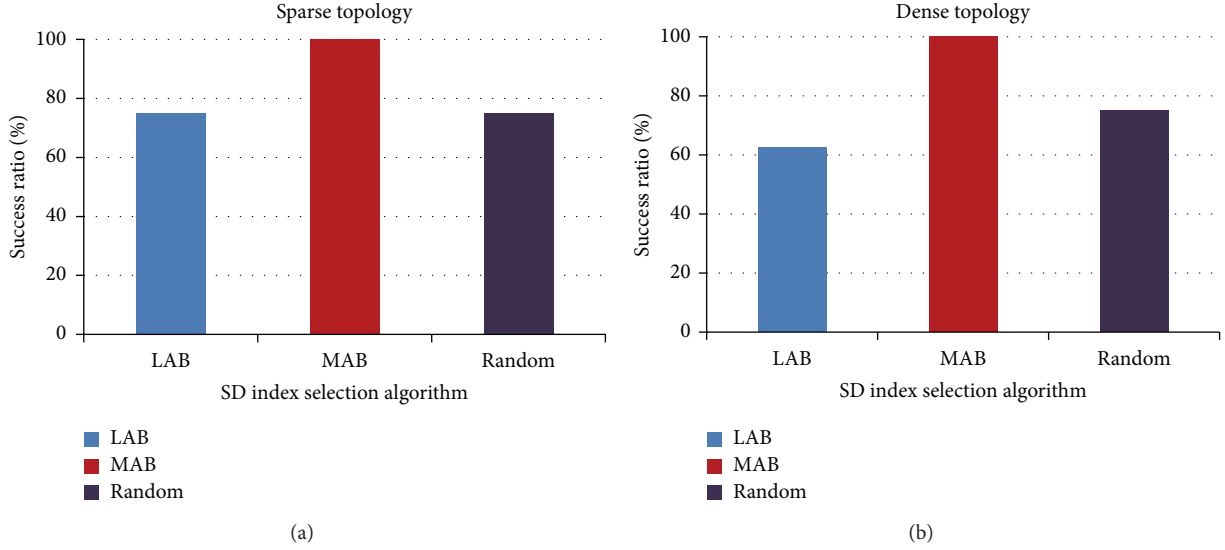


FIGURE 13: Experiment results of E-DSME beacon scheduling.

TABLE 3: Key parameters used in the experiments.

Parameter	Value
<i>Bitrate</i>	250 kb/s
<i>Symbol rate</i>	62.5 ksymbol/s
<i>aMaxPHYPacketSize</i>	127
<i>phyCurrentChannel</i>	0
<i>phyCCAMode</i>	1
<i>aBaseSlotDuration</i>	60 symbols
<i>aBaseSuperframeDuration</i>	$aBaseSlotDuration * aNumSuperframeSlot$
<i>aMaxPermissionOnlyPeriodDuration</i>	$aBaseSuperframeDuration/2$
<i>aNumSuperframeSlots</i>	16
<i>aUnitBackoffPeriod</i>	20 symbols
<i>macBeaconOrder</i>	0–15 (14)
<i>macMaxBE</i>	3–8 (5)
<i>macMaxCSMABackoffs</i>	0–5 (4)
<i>macMaxFrameRetries</i>	0–7 (3)
<i>macMinBE</i>	0– <i>macMaxBE</i> (3)
<i>macSuperframeOrder</i>	0–15 (5~7)
<i>macDSMEenabled</i>	(TRUE)
<i>macMultisuperframeorder</i>	0–15
<i>macBeaconSlotLength</i>	128

6. Performance Evaluation of the Enhanced DSME

6.1. Experiment Environments. In the previous section, we revised the pure DSME beacon scheduling step by step by analyzing experiment results. The final revision, enhanced DSME, showed a satisfactory performance in both the sparse and the dense models. However, the two topology models used in the previous experiments are so specific that we need to verify algorithm correctness and evaluate various performances of enhanced DSME via additional experiments

in which more general environments are applied. For the experiments, the number of nodes randomly deployed was also varied between 10 and 40, as shown in Figure 14. Table 3 shows the key parameters used in the experiments.

6.2. Successful Association Ratio. Figure 15 shows the results for successful allocation ratio with respect to varying the number of devices between 10 and 40. For comparative evaluation, we conducted enhanced DSME beacon scheduling by applying the MAB and LAB SD index selections, respectively. The result shows that, with LAB, as the number of devices

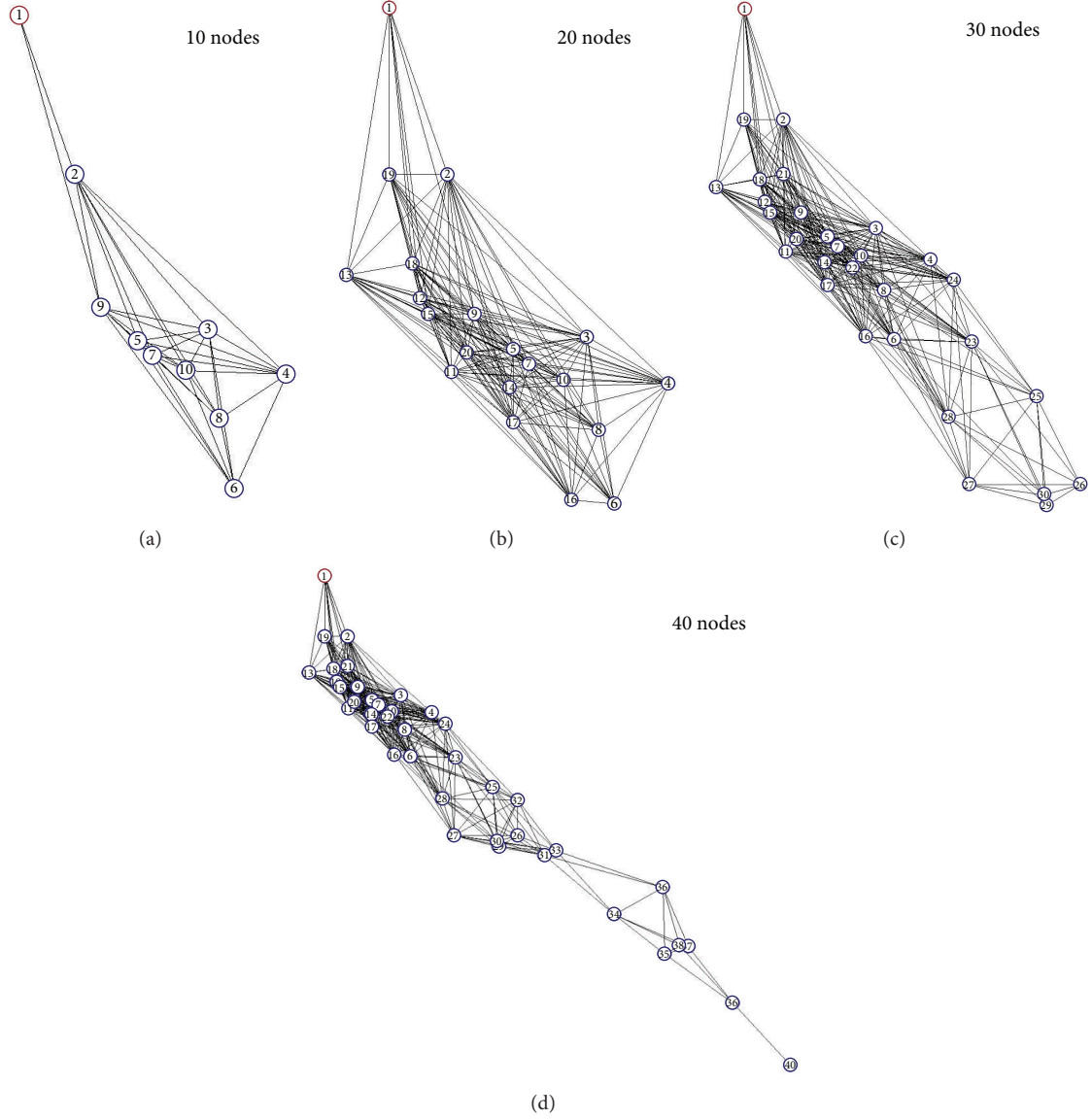


FIGURE 14: Experiment topologies.

increases, success ratio degrades. On the other hand, the result demonstrates that MAB maintains a high success ratio regardless of the number of devices. The performance degradation with LAB is because of collisions caused by the reuse of SD index slots as the topology becomes complicated. However, since MAB always selects a candidate SD index that follows the largest value in the already occupied SD index, collisions found in LAB are avoided. Therefore, note that the enhanced DSME beacon scheduling utilizing limited permissions, coupled with the MAB SD index selection scheme, shows an optimal combination.

6.3. Allocation Completion Time. We also observed the allocation completion time of E-DSME beacon scheduling. Allocation completion time represents the total time required to complete SD index allocation for all nodes in the network.

We observed the allocation completion time by varying the number of nodes in a different superframe order (SO): 5, 6, and 7. As shown in Figure 16, the result shows that the completion time is longer as SO size becomes larger. This is because a larger SO can generate more SD slots within a beacon duration. Since the duration of a superframe is represented by $\text{aBaseSuperframeDuration} \times 2^{\text{SO}}$, $0 \leq \text{SO} \leq 14$, a short superframe duration can accommodate only a small number of allocation requests, and eventually the nodes not allocated in that round have to wait until the next neighbor's beacon is received.

The number of multisuperframe slots is equal to the beacon bitmap length, and the length of a bitmap is represented by $\text{SDBitmaplength} = 2^{(\text{BO} - \text{SO})}$. So, enhanced DSME beacon scheduling can apply the result after calculating the expected number of slots in advance of network formation

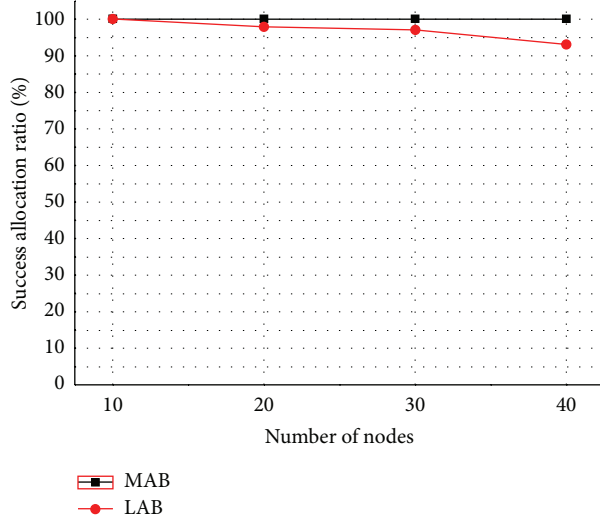


FIGURE 15: Successful allocation ratio.

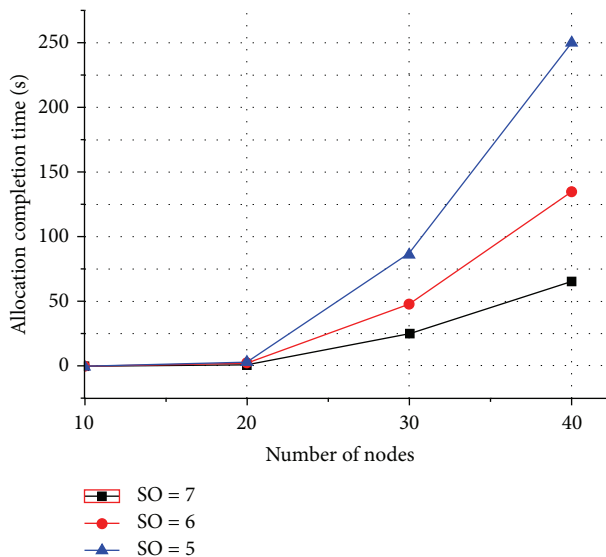


FIGURE 16: Allocation completion time.

according to the network size. Moreover, even though the network size is extended during run time, it is possible to dynamically change the expected number of slots in the process of network formation since each beacon contains BO and SO information.

7. Conclusion

In this paper, we introduced IEEE 802.15.4e DSME beacon scheduling, evaluated its validity and performance, and proposed a concrete design model. Through experiments, we found some problems in the pure DSME beacon scheduling algorithm by analyzing results. Therefore, based on the results, we revised the pure DSME beacon scheduling algorithm step by step and proposed an enhanced DSME beacon

scheduling including new features: limited permission notification and a repetitive SAD architecture.

The proposed E-DSME model is expected to contribute to design and modeling of beacon scheduling for large-scale sensor networks or IoT sensor domains.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is extended and improved from a paper accepted at the KCIC-2013 conference. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2041271).

References

- [1] "Part 15.4: wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," IEEE Standard for Information Technology IEEE Std 802.15.4, 2011.
- [2] J. Liu and S. H. Chung, "An efficient load balancing scheme for multi-gateways in wireless mesh networks," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 365–378, 2013.
- [3] A. Sinha and D. K. Lobiyal, "Performance evaluation of data aggregation for cluster-based wireless sensor network," *Human-centric Computing and Information Sciences*, vol. 3, no. 13, pp. 1–17, 2013.
- [4] M. Yoon, Y. K. Kim, and J. W. Chang, "An energy-efficient routing protocol using message success rate in wireless sensor networks," *Future Technology Research Association International*, vol. 4, no. 1, pp. 15–22, 2013.
- [5] J. J. Cho and S. S. An, "An adaptive beacon scheduling mechanism using power control in cluster-tree WPANs," *Wireless Personal Communications*, vol. 50, no. 2, pp. 143–160, 2009.
- [6] M. Lee, J. Zheng, Y. Liu et al., "Combined beacon scheduling proposal to IEEE 802.15.4b," IEEE 802.15-04-0536-00-004b, September 2004.
- [7] A. Koubâa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for IEEE 802.15.4/zigbee cluster-tree wireless sensor networks," in *Proceedings of the 19th Euromicro Conference on Real-Time Systems (ECRTS '07)*, pp. 125–135, July 2007.
- [8] S. Ahn, J. Cho, and S. An, "Slotted beacon scheduling using ZigBee Cskip mechanism," in *Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM '08)*, pp. 103–108, August 2008.
- [9] L. W. Yeh, M. S. Pan, and Y. C. Tseng, "Two-way beacon scheduling in ZigBee tree-based wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, pp. 130–137, June 2008.
- [10] L. H. Yen, Y. W. Law, and M. Palaniswami, "Risk-aware beacon scheduling for tree-based ZigBee/IEEE 802.15.4 wireless networks," in *Proceedings of the 4th Annual International Conference on Wireless Internet (WICON '08)*, November 2008.

- [11] S. Chen, L. Almeida, and Z. Wang, "Analysis and experiments for dual-rate beacon scheduling in ZigBee/IEEE 802.15.4," in *Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM '11)*, pp. 756–761, April 2011.
- [12] R. Burda and C. Wietfeld, "A distributed and autonomous beacon scheduling algorithm for IEEE 802.15.4/ZigBee networks," in *Proceedings of the 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, October 2007.
- [13] M. J. Lee, R. Zhang, J. Zheng et al., "IEEE 802.15.5 WPAN mesh standard-low rate part: meshing the wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 973–983, 2010.
- [14] P. Muthukumaran, R. de Paz, R. Spinar et al., "MeshMAC: enabling Mesh Networking over IEEE 802.15.4 through distributed beacon scheduling," in *Proceedings of the the 1st International Conference on Ad Hoc Networks (AdHocNets '09)*, Niagara Falls, Canada, September 2009.
- [15] "Amendment 5: amendment to the MAC sub-layer for low-rate Wireless Personal Area Networks (WPANs) amendment 5," IEEE Standard for Information Technology IEEE P802.15.4e draft, 2011.
- [16] W. Y. Lee, K. I. Hwang, Y. A. Jeon, and S. Choi, "Distributed fast beacon scheduling for mesh networks," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '11)*, pp. 727–732, October 2011.
- [17] ns-3, <http://www.nsnam.org/>.