

Collision-Minimizing Beacon Scheduling Scheme Using RPL in Dense TSCH-Based IoT Environment

Jaeyoung Kim, Sanghwa Chung*, and Hyungteak Shin

Department of Electrical and Computer Engineering, Pusan National University, Busan, Korea
ttled45@gmail.com, shchung@pusan.ac.kr, shin00766@pusan.ac.kr

Abstract

Time slotted channel hopping (TSCH) is one of the techniques defined in the IEEE802.15.4e standard. TSCH operates based on a time-division scheme and uses 16 separate channels for each communication, ensuring high reliability. Since the entire network operates synchronously, the network participation process is indispensable. However, due to the channel change technique of TSCH, the network participation time of a new node that wants to participate in the network becomes longer. In the previous research, channels for beaconing were chosen randomly to reduce network formation time. However, this approach is effective in networks with a smaller number of nodes, e.g., congestion occurs due to collisions of beacon messages in networks with approximately more than 20, which increases the network participation time. In addition, since channels are selected stochastically, beacons can't be broadcast evenly throughout the network. To solve this problem, this paper proposes a collision-minimizing scheduling scheme. The strategy allocates a separate timeslot for each node, eliminating the possibility of beacon message collisions. In this experiment, the average network join time decreased by 15%. The technique applied to this technique can reduce the network formation time without network congestion in a dense industrial environment.

Category: Network and Communications

Keywords: IEEE 802.15.4e; TSCH; Industrial WSN

I. INTRODUCTION

Today, the Internet of Things (IoT) has become common. Especially, a lot of large-scale dense IoT networks in which numbers of sensors form clusters are emerging from small sensor networks in the past. The smart factory is a use case of large dense networks. Sensors are attached to facilities and help to determine the next action by uploading information. In addition, facility managers can monitor the status of the facility and products being manufactured in real-time by connecting to the factory network. In order to meet these demands, there is growing interest in low-power wireless sensor networks that can

reduce wiring costs for connecting the sensors and be installed in a mobility device such as a robot.

IEEE 802.15.4e [1] TSCH MAC and RPL routing technology are typical technologies for low-power wireless sensor networks. Time slotted channel hopping (TSCH) medium access control (MAC) is a technology for low-power sensor networks. It has multi-channel and channel hopping, which is highly reliable in congested situations and well-received in industrial environments requiring stringent standards. IPv6 routing protocol for low-power lossy networks (RPL) [2] is created for low-power sensor networks. Sensors use control messages to maintain the network. It is specialized in a low-power IoT environment

Open Access <http://dx.doi.org/10.5626/JCSE.2021.15.3.97>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 29 October 2020; Revised 22 January 2021; Accepted 03 August 2021

*Corresponding Author

because it can flexibly generate a path with low overhead.

In this paper, we focus on the formation of the TSCH networks. It is a process to inform outside about the network periodically and a starting point for new nodes to join the network. Nodes in the network broadcast enhanced beacon (EB) to neighbor nodes. Nodes not joined yet select one of the 16 channels according to the standard, turn on the radio, and wait for the EB reception. The EB contains information about the network. So, nodes can join the network based on the information. However, since the radio must be turned on before receiving the EB, energy consumption increases in proportion to the waiting time of the EB. It can be a fatal problem for devices with small battery capacity. That is, differences in the network join time can affect performance in terms of network lifetime.

The main contributions of this paper are as follows:

- We solved the uncertainty of the maximum network participation time of the random-based technique.
- We proposed a technique to minimize network congestion by minimizing data collision during network formation.
- EB scheduling was attempted in the TX slot rather than the existing ADV slot, allowing the EB to broadcast throughout the slotframe.
- We implemented our scheme using the OpenMote-CC2538 [3] ported to the OpenWSN [4] project based on 802.15.4e TSCH.

The rest of the paper proceeds as follows: Section II explains the background knowledge for this paper and discusses the features and problems of previous papers. In Section III, we describe the mechanism of EB transmission in a congested environment. In Section IV, we compare and analyze the results with previous strategies. Section V presents conclusions and future work.

A. Background

This paper describes a scheme for shortening the synchronization time of the TSCH network based on the information obtained from the RPL. In this section, we describe IEEE802.15.4e TSCH, TSCH network formation, how to build destination-oriented directed acyclic graph (DODAG) of RPL protocol and DIO (DODAG information object) and DAO (destination advertisement object) message exchange method to maintain it.

1) IEEE 802.15.4e TSCH

The 802.15.4e standard is a revision of the previous 802.15.4 standard to use in an industrial environment. This standard supports three MAC behavior modes. Among them, we focus on the TSCH mode. TSCH is intended for industrial environment automation and is a time-division access technology supporting multi-channel

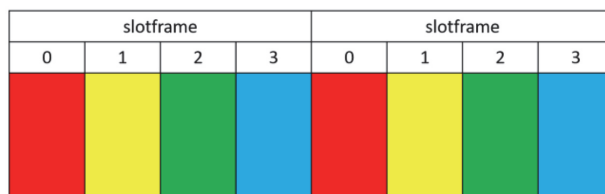


Fig. 1. The TSCH slotframe structure.

and multi-hop communication. The characteristics of the TSCH are that it is divided into 16 channels, and that the channel is changed by changing the frequency periodically. In the TSCH, all the nodes participating in the network are synchronized to the continuously repeated slot frame. Slot frame consists of an N_{slot} timeslot of a certain length. Each timeslot has enough length for each node to send and receive data packets and acknowledgments.

Fig. 1 shows a slot frame consisting of four timeslots. Another feature of the TSCH explicitly divides the channel into 16 channels. Also, it changes periodically and communicates. The channel N_{ch} used for changing the channel is composed of 16 or smaller. The physical frequency f used for actual communication in the channel change is determined by the channel offset set in the timeslot. Eq. (1) is for determining the actual frequency.

$$\text{frequency} = F[\text{ASN} + \text{channelOffset}] \bmod N_c \quad (1)$$

In Eq. (1), the absolute slot number (ASN) is the total number of timeslots that have flowed from the beginning of the network to the present. This value is shared by all the nodes and synchronized to the network. F is implemented as a lookup table, which contains information about the channel sequence. In the TSCH, each link communicates in one slot offset and channel offset, and is called as TSCH cell on {slotOffset, channelOffset} tuple. Because the ASN is changing, physical channel f determined by Eq. (1) is continuously changed.

Fig. 2 shows a simple network configuration and available link scheduling for that network. Four timeslots are assembled into one slot frame, and four-channel offsets are available.

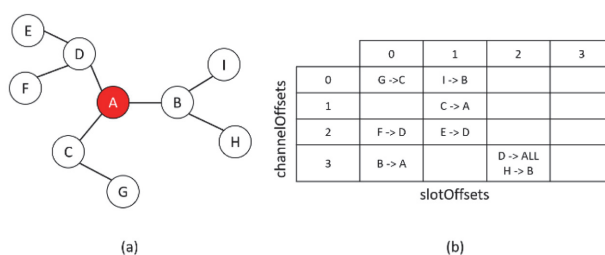


Fig. 2. (a) TSCH network topology and (b) link schedule example.

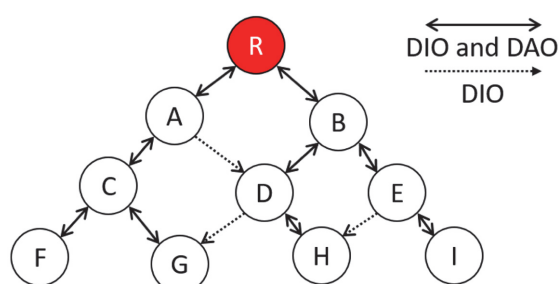


Fig. 3. An example of the RPL.

2) Routing Protocol: RPL

RPL is designed for a network with low-power and very noisy sensors. The RPL is designed to support many-to-one, one-to-many, and one-to-one traffic. The basic idea for the RPL is that nodes joining in the network create DODAGs toward a central root node. These DODAGs are managed with a unique identification. The RPL can configure the DODAG for various purposes such as energy consumption, latency, network lifetime, etc. It is represented by objective function (OF). The OF is an equation including metrics to achieve the goal. The OF may be selected by the network administrator. Each node calculates the RANK value using the OF. An advantage of the RPL is that each device can flexibly determine the uplink path with the RANK value through simple OF operation.

The RPL creates and maintains the upward routing path through the DIO message. The DIO message contains RPL INSTANCE, DODAGID, RANK, and DODAGVersion numbers. Each node collects DIO messages from neighbor nodes before joining the DODAG. Then, according to the policy of the OF, the highest priority neighbor is selected as the parent. Then, it computes its RANK value based on the RANK value received from the parent.

Fig. 3 shows the RPL routing protocol. When each node enters the network from the start node specified as root, it acquires neighbor information through the DIO messages. Among them, the node that best matches the OF strategy is selected as the parent node, and its RANK value is calculated based on the RANK value of the parent node. In the case of node G in Fig. 3, the DIO message is received from node C and node D, but the parent node is selected as node C. The DAO message is delivered through the node selected as the parent node, and finally, the routing path for G is stored in the root. If you want to transfer data from H to G, communication is done by routing the data you want to transfer to the root, then receiving the routing information and moving the data to G.

II. RELATED WORK

The EB broadcasting is an advertisement process to

announce the existence of the network. In low-power wireless networks, it is common to prevent collisions between the EB transmissions to improve energy efficiency. However, in case of frequent re-joining due to dynamic network environment or mobility device, EB broadcasting strategy is needed to improve the EB reception success rate of the device. Therefore, this paper focuses on the EB broadcasting strategy and looks for related studies to reduce network participation time (or synchronization time).

De Guglielmo et al. [5] proposed a simple random-based EB broadcasting strategy. The authors of that paper first considered the EB collision as a factor affecting network participation rate. They therefore proposed a mechanism to reduce the EB collision by controlling the number of EBs broadcasted in a timeslot.

To complement the random-based strategy, Vogli et al. [6] proposed a concept of the multi-slot frame extending slot frame and a random vertical (RV) filling and random horizontal (RH) filling mechanism using multiple channels. The RV mechanism allows nodes that can broadcast the EBs to randomly select one of the available channels to broadcast in the first slot frame of the multi-slot frame. On the other hand, the RH mechanism fixes the EBs broadcasting channel to one and randomly selects a slot frame to broadcast the EBs.

Vallati et al. [7] investigated the performance of the 6TiSCH network formation with the 6TiSCH minimal configuration and proposed a scheme that allocates/deallocates timeslots dynamically. They showed that transmissions of the EBs and RPL control messages could affect the network formation performance.

Duy et al. [8] divided slot frames into advertisement planes and communication planes to avoid the EB collisions in congested situations. They proposed an algorithm that broadcasts the EBs simultaneously on multiple channels while controlling the number of EB messages using fuzzy logic according to the number of neighbor nodes in the advertisement plane consisting of five timeslots. It reduces the congestion of EB messages by adjusting the number of EBs according to changes in the network environment.

Karalis [9] proposed the ATP, which divides an advertisement slot into multiple sub slots. It is possible because the size of the EB is smaller than 40 bytes, and link-layer broadcast is not acknowledged.

III. COLLISION-MINIMIZING BEACON SCHEDULING

We describe a scheme to minimize collision beacon scheduling in this section. In order to apply this technology, we have also been able to broadcast EBs on the Tx slot. In order to prevent the EB collision of nodes, the time slot to broadcast EB is determined by using the result of

the formula consisting of the combination of RANK and random value of the parent node. The information of the time slot determined at each node is transmitted to the root through the DAO message to ensure the independence of the network. In this process, nodes that select the same timeslot are assigned different timeslots. Next, the channel is moved to broadcast EB.

A. Expansion of Slot for Sending EB to General Tx Slot

In the IEEE 802.15.4e TSCH standard, the types of time slots are divided into the following four types: TX slot, RX slot, ADV slot, and sleep slot. Typical TX and RX slots are used for data exchange and are allocated according to a scheduling policy. The ADV slot is used to broadcast the EB and is pre-allocated. The sleep slot is allocated when no data exchange occurs. In standard and previous studies, EB was broadcast only in the ADV slot. Therefore, the ADV slot for broadcasting the EB is preset in the slot frame. As a result, the EB was broadcast in the designated ADV slot, and it was difficult to apply the change to the situation of the network. When the number of nodes is equal to or smaller than the number of channels, the number of collisions of the EB does not increase because of the channel division technique, which is a characteristic of the TSCH. However, when the number of nodes is larger than the number of channels, the number of collisions increases in proportion to the number of nodes broadcasting EB within the communication coverage range. This is fatal for nodes that want to join the network. As the number of collisions increases, the number of EBs that can be received decreases. Therefore, the waiting time of the EB on the node increases. An increase in latency increases the energy consumption of the node, thus shortening the lifetime.

We changed the EB to broadcast in the TX slot. Therefore, EB broadcasts are not guaranteed to be broadcast more than once in a slot frame, unlike broadcasts in an ADV slot. To overcome this problem, we propose a method in which each node in the network allocates a specific time slot among the entire slot frame and broadcasts the EB there. Here, the node must select a slot according to a certain criterion. The timeslot used for general data exchange changes frequently depending on the type of data and the network conditions. In this case, it is advantageous to schedule the channel offset and timeslot to dynamically change because the receiving and transmitting nodes are certain. However, in the case of EB scheduling, it is impossible to know the existence of a node that wants to participate in the network. That is, the sender must transmit the EB without knowing the existence of the sender. So, it is almost impossible to dynamically distribute the scheduling of EBs. Therefore, the proposed scheduling method allocates one Timeslot for each node by using a RANK value and a random

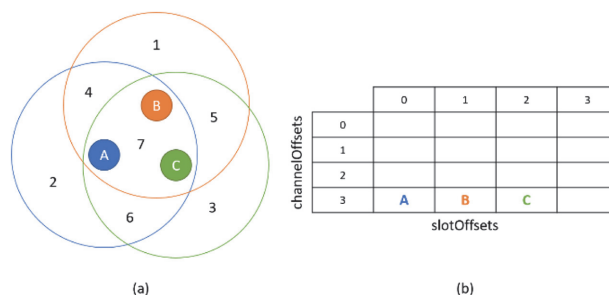


Fig. 4. An example of increasing the number of EB broadcast frequencies according to node density. (a) Three nodes participating in the network with overlapping communication ranges. (b) The timeslots assigned by nodes A, B, and C based on.

value. This mechanism is able to prevent network congestion due to EB collision because it is assigned a timeslot different from neighboring nodes in the network. In addition, since the channel movement is performed in accordance with the increase of the ASN, a node joining the network can join the network at a maximum number of 16 or smaller, regardless of which channel the node is listening to. It is possible to calculate the maximum network join time differently from other randomly based dissertation techniques.

Another advantage of this mechanism is that as the number of nodes capable of communicating in the network range increases, the network join time of the node that wants new joining of that part decreases. Fig. 4(a) shows three nodes participating in the network with overlapping communication ranges. Fig. 4(b) shows the timeslots assigned by nodes A, B, and C based on RANK. Since each node broadcasts EB at different timeslots, there is no possibility of collision between EBs. In Fig. 4(a), the broadcasted EB of node A is broadcast within the range of 4, 2, 6, and 7, respectively. In the same way, the node B emits EBs in zones 1, 4, 5, and 7, and the node C broadcasts EBs in zones 5, 3, and 6. Here, zone 2, zone 3, and zone 1 are ranges in which EBs of each node are broadcast one by one. Therefore, the maximum network participation time in this area is the time to join after traveling through all 16 channels. Next, for zone 4, zone 6, and zone 5, the EBs of the two nodes overlap. Therefore, a total of 2 EBs are transmitted per channel, and the maximum network join time is less than the maximum network lead time mentioned above. Finally, in the case of zone 7, all the EBs of nodes A, B, and C are broadcasted. Therefore, the maximum network time is further shortened.

B. Rank-based Timeslot Duplicate Prevention Mechanism

This section describes how each node selects its own timeslot using the RANK used in the RPL protocol. In order to recycle the limited slot frame, we first discuss the

strategy of partitioning. This method uses the N_{depth_total} value indicating how many nodes on the RPL can communicate with based on the root node. Eq. (2) is to find the number of timeslots per section. Eq. (3) represents an expression for obtaining the starting point of a section by using the number of timeslots per section obtained by Eq. (2). This starting point serves as the basis for the section for timeslot recycling.

$$N_{section_timeslot} = \frac{N_{slotframe}}{N_{depth_total}} \quad (2)$$

$$N_{section_start_timeslot}(i) = \frac{N_{section_number}(i)}{N_{section_timeslot}} \quad (3)$$

The node joining the network uses N_{depth_total} and $N_{slotframe}$ contained in EB to obtain $N_{section_timeslot}$. Based on this, we select timeslot by itself using Algorithm 1.

Algorithm 1. Timeslot selection algorithm

```

1: receive R(P) and  $N_{depth\_total}$  from
   preferring parent table
2:  $R(N) = R(P) + INCREASE$ 
3:  $OFFSET = R(N) \% N_{section\_timeslot}$ 
4:  $Timeslot = K[N_{depth}] + OFFSET$ 

```

N_{depth} indicates whether the current node is connected via a certain node from the root node. This value is incremented by 1 based on the parent node. The RANK value of the node is obtained as a result of the Object Function (OF) calculation. OFFSET specifies the position in the section by dividing the R(N) value obtained in the second line by $N_{section_timeslot}$. Next, select the section on line 4. If you put N_{depth} in the K function, it prints the value of the start timeslot that matches the section. Add it to OFFSET to determine the final Timeslot because the section varies depending on the N_{depth} of the node, the section can be recycled every N_{depth} . This is useful when applied to large networks.

C. Timeslot Duplication Prevention Mechanism using RPL's DAO and DAO ACK

The previously proposed RANK-based time slot selection technique is a mechanism in which the node itself determines its own time slot using the parent's RANK value. However, the preceding mechanism is problematic in two cases. First, the number of nodes to be allocated is larger than the size of the timeslot per section. The second problem arises when the values of OFFSET are the same and are assigned to the same timeslot, and the timeslot selected by the node itself is duplicated. These two problems occur when nodes join the network frequently. Therefore, this section discusses

the mechanisms that complement these two problems.

The RANK-based timeslot mechanism discussed above uses the RPL technology to form a network. We transmit the timeslot allocation information of the nodes in the network to the Root node using the DAO message, which is the control message used for network formation.

Algorithm 2 shows the node-pointed Timeslot duplication prevention mechanism. After setting the timeslot using the method shown in the previous section, send timeslot and $M_{section}(N_{section} = N_{depth} \% N_{depth_total})$ to DAO message to complete the network configuration. Then, the DAO ACK is received by the root node. If it is the same as the previously selected timeslot, EB is broadcast at that location. If it is different, EB is broadcast in the changed timeslot. Line 1 sets timeslot with the RANK-based timeslot selection algorithm discussed earlier. Lines 2 and 3 contain the timeslot and the depth number to which it belongs in the DAO, sent to the root node, and waiting for the ACK of DAO message. Line 4–5 receives the ACK and changes to the new timeslot sent by the root node if the timeslot does not overlap with the one calculated by itself. If the duplicated timeslot comes, it confirms the timeslot that was set first and broadcast EB.

Algorithm 2. Timeslot duplicate correction algorithm (Node perspective)

```

1:  $Timeslot = K[N_{depth}] + OFFSET$ 
2: make the DAO message including
   Timeslot and  $N_{section}$ 
3: wait the DAO Ack
4: receive the DAO Ack
5: change Timeslot or determine
   Timeslot

```

Algorithm 3. Timeslot duplicate correction algorithm (Root perspective)

```

1: receive the DAO message
2: check TimeslotList[Timeslot]
   and section
3: if (Timeslot is used)
4:   check  $N_{depth}$ 
5:   if ( $depth_{before} == depth_{now}$ )
6:     include < new timeslot >
7:   end if
8:   include < default message >
9: else
10:  include < default message >
11: end if
12: send the DAO Ack

```

Algorithm 3 shows the Timeslot redundancy correction

algorithm from the root node viewpoint. Root node receives DAO and sends DAO ACK.

The root node receives a DAO message from each node in the network. And the root node checks the depth of the node assigned to the timeslot from the existing network information. If there is no node assigned to the transmitted timeslot, the value of the timeslot is sent as it is. If there is an assigned node, selects an empty timeslot and sends it to the message. The first line receives a DAO message from the node. Line 2 is the process of verifying that there is a duplicate timeslot node in the network using the timeslot and section information in the DAO. Lines 3–8 indicate the process of determining if a timeslot is used, if it is used. Line 5 compares the depth of the node using the existing timeslot using timeslot to the depth of the node that is currently using it. If it is the same depth, another timeslot is assigned. If it is not the same depth, it is regarded as timeslot reuse and put the default message in the DAO as shown on line 8. Line 10 indicates that an existing message is sent in a DAO when timeslot is not used. Finally, the DAO ACK is sent to the node that sent the DAO. Here, the method for determining the new timeslot in line 6 can be made through various strategies.

Fig. 5 shows the flow diagram of the previous two algorithms: the node view for network participation and the root node view. In Fig. 5(a), the flow diagram starts when the node belonging to the network receives the broadcast EB. We compute timeslot with the algorithm described above, create a DAO message, and wait for ACK after sending it to the root node. If the timeslot in the ACK from the root node matches the timeslot calculated by the ACK, it starts EB broadcasting as it is, changes its timeslot, and starts broadcasting. In Fig. 5(b), it is always ready to receive the DAO. When a DAO is received, it compares the specified timeslot of the entire network node of the root with the timeslot received from

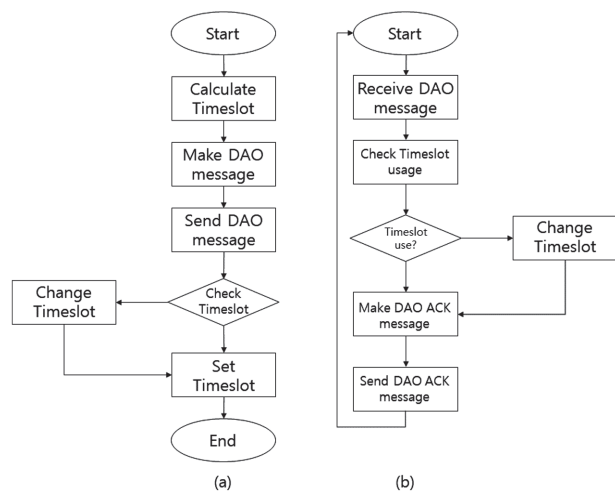


Fig. 5. Timeslot duplicate correction algorithm flow diagram: (a) node and (b) root.

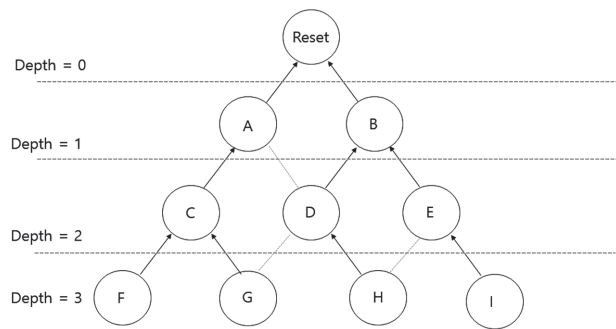


Fig. 6. RANK-based Timeslot selection mechanism example network diagram.

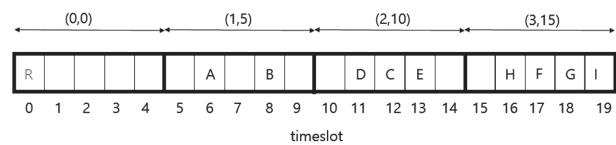


Fig. 7. RANK-based Timeslot selection mechanism timeslot example.

any node. If you do not use any node on the network, it will send the value back to ACK and, if you are using it, send the specified timeslot information to ACK using a different strategy.

Figs. 6 and 7 show examples of RANK-based time slot selection mechanism and redundancy prevention mechanism. The depth is divided into four depths, and the beginning of each depth consists of 0, 5, 10, and 15 values. The length of the slot frame is 20. For the first depth, the timeslot starts at 5. Nodes A and B receive the EB message from the root and join the network. One higher depth is determined than the upper depth value. Using the formula, as discussed earlier, we calculated the timeslot. The calculation result is added to 5, the starting point of section 1, and its own timeslot is confirmed. The determined value is stored in the DAO message, sent to the root, and broadcasted at that location after the timeslot confirm operation is completed. Next, since C, D, and E belong to depth 2, the timeslot starts at 10 and is calculated based on the parent node's RANK value. The following sections apply in the same way. When a new node of F, G, H, or I is entered, the section changes back to zero. In this case, you can use channel 0, which is used by root. That is, a node having a sufficient distance and outside the communication, distance can reuse the time slot.

IV. EXPERIMENTS

All experiments conducted in this study were prepared on an OpenWSN [4] stack, and all the experiments were implemented using OpenMote [3].

Fig. 8 shows the experimental network configuration.

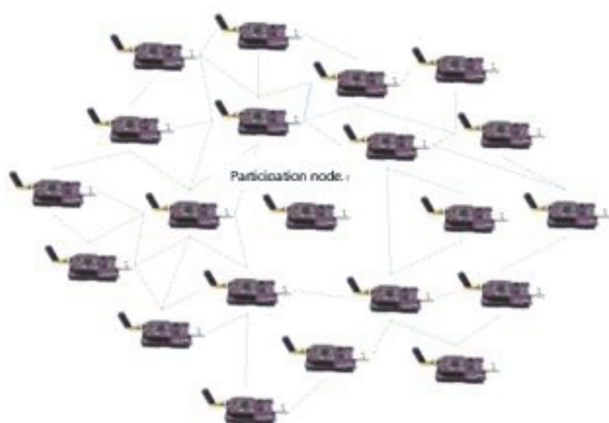


Fig. 8. Experimental network configuration.

A total of 30 OpenMote [3] experiments were conducted. All 30 nodes are within communication range and maintain the DODAG according to the RPL protocol. The participating node selects one of the 16 channels for network entry at the network center and awaits an EB reception. The timeslot length is 15 ms, and the slot frame consists of 101 timeslots. A multi-slot frame consists of 10 slot frames. The network participation time is measured by calculating the minimum network joining time after the power of the node for network participation is turned on, and the ASN value is converted into seconds. The average network join time represents an average of 100 values, and the maximum network join time represents the maximum value among the 100 times measurements.

A. Comparison of Average Network Join Time

Fig. 9 shows the average network joining time of RV, which is a kind of random-based beacon scheduling. The vertical axis represents the average network engagement

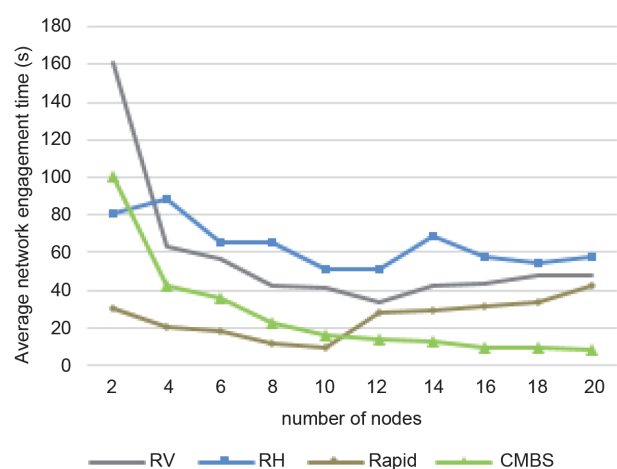


Fig. 9. Average network joining time.

time, and the horizontal axis represents the nodes participating in the network. As the number of nodes participating in the network ranges from 1 to 13, as the number of nodes increases, the number of EBs broadcast on the channel increases, so the average network engagement time decreases naturally. However, when the number of network participating nodes is more than 14, the engagement time of the network is gradually increasing. This is because the nodes participating in the network randomly select channels to broadcast EBs. When the number of nodes increases, the number of conflicts between EBs increases because the number of nodes selected for the overlapping channels increases.

Fig. 9 shows the average network joining time of our proposed collision-minimizing beacon scheduling (CMBS). The results are shown in Fig. 9. The average network joining time decreases steadily as the number of nodes increases. We applied the strategy of broadcasting the EB by separating timeslot. As a result, there is almost no collision between EBs, and the number of EB transmissions per slot frame increases proportionally to the number of nodes in the network. The average network participating time of the nodes that want to join the network has steadily decreased. Especially, from the point where the number of nodes where channel saturation occurs is more than 17, it is different from Fig. 9. We can expect this trend to continue until the slot frame is saturated. In this case, if we use the section reuse scheme through slot frame partitioning, we can use the collision minimization beacon scheduling method regardless of the slot frame length.

Also, Fig. 9 shows a graph of the average network joining time of the conventional random-based beacon scheduling scheme and the scheduling scheme presented in this paper. In the case of RV, the average network engagement time is steadily decreased to 12 nodes, and it decreases to 34 seconds maximum. However, the network engagement time tends to increase slightly from 12 or more. The reason for this is that EBs are randomly selected from among 16 channels, and EBs collide. In the case of RH, the engagement time was continuously decreased to 12, and the time was reduced to 50 seconds. The reason why RH takes network engagement time longer than RV is that RH has more than 10 randomly selectable slots, which means more conflicts and a longer average engagement time. In the case of the Rapid technique, the average network participation time is the shortest in 2–10. This is because, between 1 and 2 EBs, three EBs are broadcast between 5 and 3 and 6, and one EB is broadcast continuously. Therefore, it can be said that the engagement time is faster because one node emits several EBs. However, since the number of EBs broadcasting one EB per node is decreasing, the number of network engagement time is gradually increasing due to collision between EBs and RVs after the 12th. In the case of CMBS proposed in this paper, we can see that the

average network participation time decreases continuously as the number of nodes increases. Experimental results show that the number of nodes is more than 25–27 seconds faster than the average network participation time of other techniques. The gap becomes more different as the number of nodes increases. This difference occurs because the nodes participating in the network broadcast the EB by dividing the timeslot, so there is no conflict with EB. It can be expected that the scheduling technique presented in the paper shows a better effect in a factory environment where electronic devices and equipment are concentrated.

B. Comparison of Maximum Network Participation Time

Fig. 10 is a graph showing the result of measuring the maximum value of the CMBS network participation time that we proposed. Since each timeslot is assigned to a node, the maximum network participation time decreases as the number of nodes in the network increases. The special point is that when the number of channel nodes is 16 or more, the value is not changed anymore but is changed between 30 and 50 seconds. This is because the value no longer shrinks due to the saturation of one section. Fig. 10 compares the maximum value of the network participation time between the conventional random channel selection-based technique and the proposed paper technique. Conventional random channel selection-based techniques change the maximum value without special tendency because the channel is selected at random. This can serve as a variable in terms of network design. Since it is possible to measure the joining time only through experiments, it is necessary to obtain the value through multiple measurements each time. However, because of the random selection of the nodes belonging to the network instantaneously, we cannot

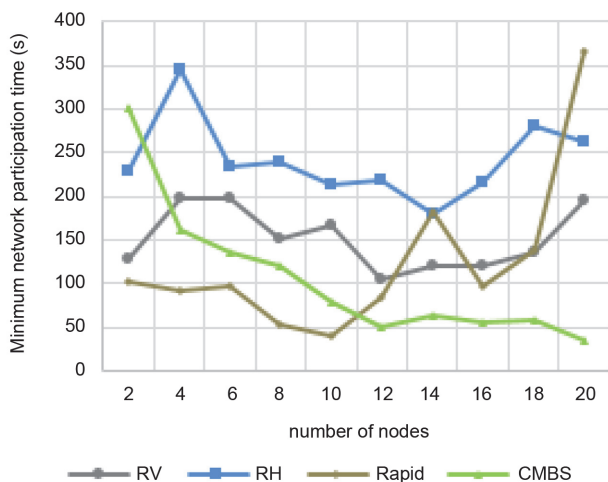


Fig. 10. Comparison of network joining time.

trust the value through the measurement. However, in the case of the CMBS proposed by us, the maximum network participation time tends to decrease when the number of nodes increases and the average joining join time between 30 and 50 seconds can be predicted. This can contribute significantly to the stability of the network design.

C. Comparison of the Proportion of EB Slots in the Total Number of Timeslots

Some of the timeslots that make up the slot frame are used to send EBs, and the rest are used to exchange data. As the number of slots used to transmit EBs in the entire timeslot is smaller, the number of slots to which data can be transmitted is increased, and the data rate of the entire network is increased. In this experiment, Rapid [8], a comparison algorithm, compares our algorithm with the technique of changing the number of EBs according to neighbor nodes. The experiment was divided into five nodes and 20 nodes.

First, the experiment results of the EB slot utilization when the neighbor nodes are five are shown in Fig. 11. The horizontal axis of the graph represents the number of channel offsets, and the vertical axis represents the EB timeslot utilization rate. This shows the ratio of the EB slot to the total number of slot frames. In the case of Rapid, it always uses five ADV slots to transmit EB, with channel offset \times 5 timeslot slot, always keeping a 4.95% share. Since RV uses one ADV slot as in the standard, it is used for EB by the number of 1 timeslot \times channel offset regardless of the number of channel offset. Therefore, EB Timeslot share of 0.99% is always maintained. However, in the case of CMBS, EB broadcasts in the same slot as the number of nodes participating in the network. Therefore, if the channel offset increases and the number of available timeslots increases, the share of the EB slot decreases. This shows that the EB slot usage rate is lower

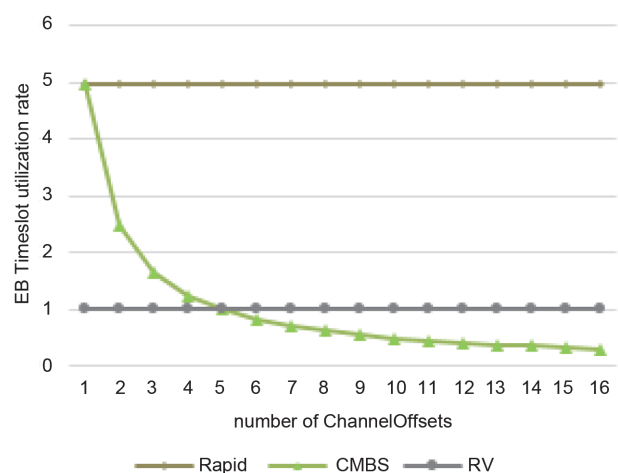


Fig. 11. EB slot utilization rate with five network nodes.

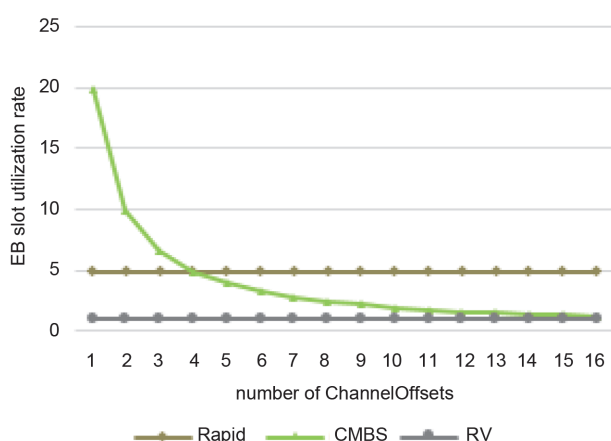


Fig. 12. EB slot utilization rate with 20 network nodes.

than Rapid in all cases except for one channel offset, and EB slot usage rate is lower than RV for channel offset of 6 or more.

Fig. 12 shows the usage rate of 20 EB slots of the entire network node. RV slot utilization and Rapid slot utilization are still maintained at 0.99% and 4.95%, as in the previous experiment. However, in the case of CMBS, the slot is allocated according to the number of nodes, so it is 20% for one channel offset, and lower than that for Rapid in the case of 4. In the case of 16, it shows the same EB slot usage rate as RV. If you use less channel offset, there is not enough slot for data exchange in CMBS. This is because one EB slot is assigned to each node. However, since the number of timeslots increases with the use of channel offset more, the utilization rate of the EB slot of CMBS becomes lower as the number of channel offset increases. In the case of the strategies of the previous paper, since the EB slot is fixed to the advance slot in advance, an unnecessary EB slot is generated. This may waste a slot in a large-scale sensor network in which data exchange is frequent, so data exchange may not be performed properly.

In this experiment, we can see the network in which CMBS is used efficiently. It is optimized in a large-scale sensor network environment where more nodes than 16 channels which are standard channels of TSCH, exist in the network.

One extra time slot is $1/101 \times 100\% = 0.99\%$ of the total slot frame, which is less than 1%. Considering that the length of the timeslot is 15 ms, it can be inferred that the processing speed decreases by $15 \text{ ms} \times 0.01 = 0.15 \text{ ms} = 0.00015 \text{ s}$, but this can be considered insignificant.

V. CONCLUSION

In this paper, we propose a beacon scheduling scheme that minimizes collision in TSCH-based dense IoT environment. The IEEE 802.15.4e TSCH standard does

not specify how to form a network. To solve this problem, other papers randomly selected a channel to reduce network participation time. However, because these methods are based on randomness, the expected convergence time cannot be predicted, and network congestion occurs due to collision between EBs. To solve this problem, we proposed a method to allocate Timeslot using the RPL value of RPL, which is commonly used as a routing protocol in TSCH. To demonstrate the performance of this technique, a comparison experiment with existing papers was conducted using OpenMote with the OpenWSN stack installed. In this paper, we prove that the CMBS scheme proposed in this paper reduces network join time and maximum participation time in the dense network environment. In this paper, the number of nodes is limited to 30, and the process is in progress. In the future, we will use simulation to demonstrate that the proposed method is effective in nodes with more than 50 large-scale network environments. In addition, since EB is scheduled to be extended to the entire slot frame, a method of linking with scheduling for actual data exchange will also be added.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Grand Information Technology Research Center support program (No. IITP-2020-2016-0-00318) supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP)

REFERENCES

1. *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-rate wireless personal area networks (LR-WPANs)*, IEEE 802.15.4-2011, 2011.
2. T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, “RPL: IPv6 routing protocol for low-power and lossy networks,” Internet Engineering Task Force, Fremont, CA, *RFC 6550*, 2012.
3. OpenMote [Online]. Available: <http://openmote.com>.
4. Berkeley’s OpenWSN [Online]. Available: <https://openwsn.atlassian.net/wiki/spaces/OW/overview>.
5. D. De Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, “A performance analysis of the network formation process in IEEE 802.15.4e TSCH wireless sensor/actuator networks,” in *Proceedings of 2014 IEEE Symposium on Computers and Communications (ISCC)*, Funchal, Portugal, 2014, pp. 1-6.
6. E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, “Fast join and synchronization schema in the IEEE 802.15.4e MAC,” in *Proceedings of 2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, New Orleans, LA, 2015, pp. 85-90.

7. C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving network formation in 6TiSCH networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 98-110, 2019.
8. T. P. Duy, T. Dinh, and Y. Kim, "A rapid joining scheme based on fuzzy logic for highly dynamic IEEE 802.15.4e time-slotted channel hopping networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, article no.1550147716659424, 2016.
9. A. Karalis, "ATP: a fast joining technique for IEEE802.15.4-TSCH networks," in *Proceedings of 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Chania, Greece, 2018, pp. 588-599.



Jaeyoung Kim

Jaeyoung Kim received the B.E. and M.E. degrees in computer engineering from Pusan National University, Busan, Korea, in 2016 and 2018, respectively. He is currently a researcher with the KEPCO, Naju, Korea. His research interests include active industrial wireless sensor network, IoT, and future Internet.



Sanghwa Chung <https://orcid.org/0000-0003-1329-1188>

Sanghwa Chung received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1985, the M.S. degree in computer engineering from Iowa State University, Ames, IA, in 1988, and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, in 1993. He was an assistant professor with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, from 1993 to 1994. He is currently a professor with the Department of Computer Engineering, Pusan National University, Busan, Korea. He is currently the head of Dong-Nam Grand ICT R&D Center in Busan, Korea. His research interests are in the areas of industrial wireless sensor networks, IoT, Future Internet, and SDN.



Hyungteak Shin

Hyungteak Shin received the B.S. degree in Chemistry from Pusan National University, Busan, Korea, in 2015. He joined the College of Computer Science Engineering, Pusan National University in March 2019. He is currently studying for a master's degree. His research interests include active industrial wireless sensor network, IoT, and future Internet.