Opportunistic Transmission of Control Packets for Faster Formation of 6TiSCH Network

ALAKESH KALITA and MANAS KHATUA, Indian Institute of Technology Guwahati, India

Network bootstrapping is one of the initial tasks executed in any wireless network such as Industrial Internet of Things (IIoT). Fast formation of IIoT network helps in resource conservation and efficient data collection. Our probabilistic analysis reveals that the performance of 6TiSCH based IIoT network formation degrades with time because of the following reasons: (i) IETF 6TiSCH Minimal Configuration (6TiSCH-MC) standard considered that beacon frame has the highest priority over all other control packets, (ii) 6TiSCH-MC provides minimal routing information during network formation, and (iii) sometimes, joined node can not transmit control packets due to high congestion in shared slots. To deal with these problems, this article proposes two schemes—opportunistic priority alternation and rate control (OPR) and opportunistic channel access (OCA). OPR dynamically adjusts the priority of control packets and provides sufficient routing information during network bootstrapping, whereas OCA allows the nodes having urgent packet to transmit it in less time. Along with the theoretical analysis of the proposed schemes, we also provide comparison-based simulation and real testbed experiment results to validate the proposed schemes together. The received results show significant performance improvements in terms of joining time and energy consumption.

CCS Concepts: • **Networks** → **Link-layer protocols**; **Network simulations**; **Network performance analysis**;

Additional Key Words and Phrases: Industrial IoT, 6TiSCH, network formation, opportunistic resource allocation

ACM Reference format:
Alakesh Kalita and Manas Khatua. 2020. Opportunistic Transmission of Control Packets for Faster Formation of 6TiSCH Network. *ACM Trans. Internet Things* 2, 1, Article 5 (December 2020), 29 pages.
https://doi.org/10.1145/3430380

1 INTRODUCTION

6TiSCH is a communication architecture for Industrial Internet of Things (IIoT) [1, 2–4], which provides interoperability between IEEE 802.15.4e Time Slotted Channel Hopping (TSCH) [5–7] MAC layer and IETF's higher protocol layers [8, 9]. Along with interoperability, 6TiSCH also deals with network formation. 6TiSCH Working Group (established by IETF) published 6TiSCH Minimal Configuration (6TiSCH-MC) [10] standard for efficient bootstrapping in 6TiSCH network. 6TiSCH-MC recommends to use only a single shared cell (also called minimal cell) in a slotframe or multi-slotframe for transmitting all control packets in 6TiSCH networks. Again, if we consider secure

Authors' addresses: A. Kalita and M. Khatua, Indian Institute of Technology Guwahati, Computer Science and Engineering Department, Assam, India, 781039; emails: alakesh.kalita1025@gmail.com, manaskhatua@iitg.ac.in.

2577-6207/2020/12-ART5 $15.00
https://doi.org/10.1145/3430380

enrollment of nodes [11, 12], then few other control packets such as join request (JRQ) and join response (JRS) are also transmitted in shared cells.

Due to this limited bandwidth for control packet transmissions, the formation of 6TiSCH network degrades when pledges (new nodes) join with the network. According to 6TiSCH specification, the network advertisement frame, i.e., Enhanced Beacon (EB), has the highest priority among all the control packets. Because of this fixed priority along with scarcity of bandwidth, the other control packets starve to get transmitted by a joined node. Again, the requirement of routing information for a newly joining node is unseen to the already joined nodes. Hence, the joined nodes do not transmit routing information packets immediately when it is required for pledge. And thus, the pledges suffer from long waiting times to get routing information packets even after getting valid EB. Furthermore, uncertainty in accessing the shared cells also contributes to longer network formation time in dense 6TiSCH network. It is observed that none of the existing works address the above-mentioned problems. State-of-the-art approaches (e.g., References [13–17]) focused on either the number of transmissions or scheduling of EB frames during network formation. However, both the EB frame and the routing information packet are necessary during multi-hop 6TiSCH network formation. At first, Vallati et al. [18] considered both control packets in 6TiSCH network formation. Later, Vucinic et al. [19] and Kalita and Khatua [20–22] considered both these control packets along with the control packets for secure enrollment of nodes as per References [11, 12] during 6TiSCH network formation.

To deal with EB's highest priority, incoherency between network information requirement and its sending rate, and uncertainty in accessing shared transmission channel, we propose two schemes. The first one is an intra-node scheme, namely, opportunistic priority alternation and rate control (OPR), which dynamically adjusts the priority of different control packets based on their requirement in the network. It also opportunistically increases the rate of transmission of routing information packet during network formation. The other scheme, namely, opportunistic channel access (OCA), provides a global solution to the nodes having urgent packets to send by allowing them to access shared transmission channel opportunistically. This is basically an inter-node scheme. The proposed schemes together reduce the overall network formation time.

Note that this work is the extended version of our previous work [21]. In the previous work, the demerits of EB's highest priority and the absence of routing information are mentioned. Here, an additional but related problem, i.e., uncertainty of shared channel access during bootstrapping in 6TiSCH network, is coined out. Analytical validation of these problems are also provided. In this work, theoretical analysis of both the solution schemes are provided for validating their effectiveness. Furthermore, along with the simulation experiments, we also perform real testbed experiments at FIT IoT-LAB [23]. The received results show how the proposed schemes outperform the benchmark schemes.

We summarize the major contributions of this article as follows:

- Analytically, we show the demerits of EB's highest priority, absence of routing information, and uncertainty of shared channel access during bootstrapping in 6TiSCH network.
- An OPR scheme is proposed to deal with the demerits of EB's highest priority and absence of routing information.
- An OCA scheme is proposed to transmit urgent packets with minimum delay.
- Theoretical analysis of the proposed schemes are provided using Markov chain model.
- Comparison-based performance evaluation of the proposed solutions are performed using IoT network simulator and FIT IoT-LAB real testbed.

The rest of the article is organized as follows: Section 2 summarizes the existing works related to network formation in 6TiSCH network. In Section 3, we describe the 6TiSCH network formation

procedure along with its shortcomings. In the same section, we also validate the problems using theoretical analysis. Section 4 describes the proposed methodologies along with their theoretical analysis. By presenting the performance evaluation of the proposed schemes and comparing their results with existing benchmark schemes in Section 5 and Section 6, we conclude this article in Section 7.

2 RELATED WORKS

At the very beginning, different works are published considering only EB frame for network formation. When a pledge receives a valid EB, it is considered as a joined node. Therefore, researchers concentrate either on the number of transmitted EB frames [15] or on collision-free EB scheduling [13, 14, 16, 24]. However, this consideration is not applicable in multi-hop networks. Both the EB and routing information carrying control packets are necessary during the formation of multi-hop 6TiSCH network to avoid network inconsistency. In Low power and Lossy Networks (LLNs), usually the Routing over Low Power and Lossy Networks (RPL) [25] is used as network layer routing protocol. RPL constructs a loop-free Destination Oriented Directed Acyclic Graph (DODAG) routing topology. For constructing DODAG, DODAG Information Object (DIO) routing information carrying control packet is used. Only a few works have been published considering both the EB and DIO packets during 6TiSCH network formation [18–20]. Therefore, we divide our literature review into two parts. The works considering only EB frame during network formation are included in the first part; in the second part, we include the works that considered both the EB and DIO packets during 6TiSCH network formation.

2.1 TSCH Network Formation

For faster TSCH network construction, few methods increased the number of transmitted EB frames, whereas, some other works concentrated on the scheduling of EB frames to avoid collision. Guglielmo et al. [14] proposed a simple Random-based Advertisement (RA) algorithm for sending EBs. In RA, already joined nodes send their EBs in random shared slots assigned by their parents. Further, by varying the transmission probability of EB frame depending on various network parameters, EB collision probability is reduced. Vogli et al. [13] proposed two algorithms for faster TSCH network formation—Random Vertical filling (RV) and Random Horizontal filling (RH). In RV, the join registrar/coordinator (JRC) (also know as PAN coordinator) sends EB using channelOffset 0, while the other joined node use any random channelOffsets for the same. Starting timeslot of a multi-slotframe is used to transmit a single EB by the JRC and other joined node. Whereas, in RH, JRC and other joined node send their EBs in channelOffset 0. The JRC and other joined nodes send their EB frames in different slotframes. JRC sends in slotframe 0, whereas other coordinator nodes select any random slotframe. Considering the JRC has enough energy, Vogli et al. [15] published enhanced version of RV and RH, which are Enhanced Coordinated Vertical filling (ECV) and Enhanced Coordinated Horizontal filling (ECH). In both the ECV and ECH approaches, JRC sends EB frame in every slotframe of a multi-slotframe using channelOffset 0, whereas other coordinator nodes follow the same approach as in RV and RH. Guglielmo et al. [16] proposed an EB scheduling, namely, Model-based Beacon Scheduling (MBS) algorithm. By solving an optimization problem, the JRC assigns a unique link to other joined node to transmit their EBs by avoiding collision. Authors also mentioned that equal-spaced shared slots in a multi-slotframe help in faster formation of TSCH network. Instead of using any slot as a shared slot, Khoufi et al. [24] proposed to use fixed advertising slot. To obtain this, they proposed a Deterministic Beacon Advertising (DBA) algorithm for transmitting beacons over all the available frequencies without any collision.

None of the above-mentioned works considered the transmission of routing information during network formation. Furthermore, in some works, newly joined nodes transmit their EB frames using different channelOffsets from their parent nodes at the same time. This can result in de-synchronization between the parent-child nodes.

2.2 6TiSCH Network Formation

There exist only a few works where both the EB frame and DIO packet are considered for network formation. Vallati et al. [18] proved that the 6TiSCH-MC does not provide enough resources to send all the control packets during network bootstrap. Hence, instead of using a single shared cell in a slotframe/multi-slotframe, the authors used more number of shared cells to transmit the control packets in less time. However, the allocation of extra shared cells costs in higher energy consumption and disturb in transmission of general data packets. Further, it does not follow the 6TiSCH-MC standard in terms of resource usages. Again the authors did not consider anything about resource usage by non-deterministic JRQ and JRS frames for secure enrollment of nodes. Vucinic et al. [19] proposed that the beacon transmission probability should be 0.1 irrespective of the number of nodes present in the network for faster network formation. Kalita and Khatua [20, 22] proposed to use dynamic beacons interval instead of using fixed beacon interval. The authors proposed to compute dynamic beacon intervals depending on channel busy ration.

The work in Reference [26] proposed an asynchronous distributed scheduling algorithm where a node autonomously schedules its EB, DIO, and other data packets during network formation. However, they did not evaluate their scheduling algorithm during network formation. The authors only evaluated their proposed method during static network condition. Furthermore, the work used more number of shared cells. Kalita et al. [21] mentioned the problem of EB's highest priority and absence of routing information packet during network bootstrapping. The authors solved these problems within a node. However, after solving these problems, the desired packet needs to be transmitted as early as possible for further reduction of the formation time. This early transmission of the packet is not mentioned in the work. Furthermore, all the above-mentioned works (except Reference [21]) considered EB frame has the highest priority among all the control packets. None of the works considered sufficient transmission of routing information packet during network bootstrapping. Although, few works, such as Reference [27–30], have been published to improve the performance of RPL by tuning various RPL parameters, the works did not consider quick generation of routing information packet when it is actually required. Apart from all these, there is an uncertainty in accessing the shared transmission cells [31, 32] because of the random channel access protocol such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). None of the existing works related to 6TiSCH network formation address the uncertainty in accessing shared transmission cell.

3 PROBLEM STATEMENT AND VALIDATION

At the beginning of this section, we describe the network formation procedure in 6TiSCH network. We observe few shortcomings in the existing methods, which motivate us to propose a better scheme for faster formation of 6TiSCH network. At first, the observations are described, followed by their validations using numerical analysis. Finally, we propose two solution methods to address those problems.

3.1 Network Formation Procedure

The join registrar/coordinator (JRC) or RPL root of 6TiSCH network initiates the network formation process. JRC broadcasts EB frame periodically to advertise a network. An EB contains required information for pledges to join. A pledge turns on its radio and starts listening to a random

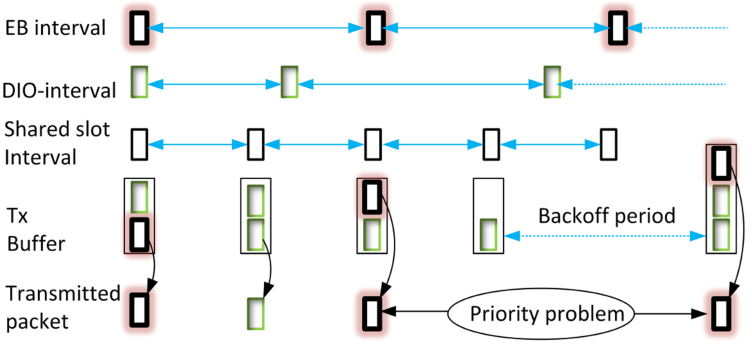


Fig. 1. Starving DIO packet due to EB's highest priority.

channel for EB frame. If it does not get a valid EB within a specified amount of time, it switches to some other channel to listen. This process is repeated until it receives a valid EB frame. Along with EB frame transmission, JRC (or RPL root) also initiates the construction of DODAG topology following the RPL routing protocol by sending DIO packet. Once a pledge receives a valid EB frame from the JRC or any other coordinator, the pledge becomes synchronized with the network. Hence, the pledge gets to know the position of shared cell, i.e., `[slotOffset, channelOffset]` in a slotframe/multi-slotframe. So, the node activates its radio only in shared slot to save energy. To complete the secure joining process, the pledge has to send JRQ frame to its parent node. Once a pledge receives the response of its JRQ frame from its parent, it starts listening for DIO packet. After receiving a valid DIO packet, a pledge is eligible to send its beacon for further network expansion. The importance of DIO packet during network expansion in multi-hop networks is to avoid inconsistency in networks. In this way, all the pledges join the network one by one. The formation of entire network completes once all nodes join the network successfully.

3.2 Starvation Due to EB Priority

Any node having packets to transmit, at first, participates in channel contention to get access in shared cells. The node performs random backoff once it encounters busy channel. Problem occurs when an EB frame is generated during the node's backoff period. Due to the EB's highest priority over all other control packets, that node transmits its newly generated EB frame although there are other control packets waiting in its transmission buffer for a long time. In a dense network, the probability of winning a contention by a node is very less. Hence, the node has higher chance to go to backoff. As a result, other control packets such as DIO, DIS, Keep-Alive, and so on, suffer from starvation because of EB's highest priority. When a pledge waits for a DIO packet to complete its joining process, the pledge needs to wait for longer time duration due to above-mentioned problem. Hence, the total joining time of the pledge increases, which in turn affects the overall network formation time. Figure 1 pictorially describes the packet-level starvation problem happens in intra-network. The figure shows that a DIO packet is replaced by a newly generated EB frame during backoff time of an already joined node.

3.3 Negative Effect of Trickle Algorithm

In 6TiSCH network, the rate of sending DIO packet is decided by the Trickle Algorithm (TA) [33] in network layer. The main use of TA is to save energy consumption by sending minimum routing information packet. Initially, DIO transmission interval starts with a minimum duration of I_{min}. Until the interval reaches its maximum value of I_{max}, the interval duration becomes double at the

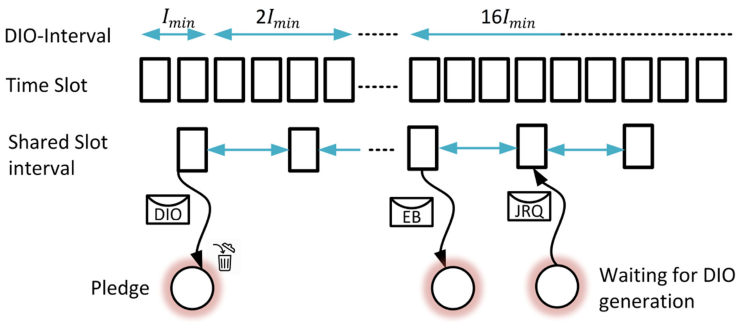


Fig. 2. Negative effect of Trickle Algorithm.

end of each interval. If any inconsistency occurs, such as the parent node changes its rank, then the node resets its Trickle interval to I_{min}. Other than that, the interval is not changed. Hence, the interval of DIO in TA is unaffected by a pledge's JRQ frame. Therefore, in the worst case, a pledge needs to wait for at least maximum Trickle interval time (I_{max}) to get a DIO packet from its parent. Moreover, after generating a DIO packet, a joined node may encounter packet-level starvation as explained in Section 3.2. This scenario leads to higher joining time to a pledge, which is waiting for a DIO packet from its parent node. Figure 2 describes the effect of Trickle Algorithm. Here, a pledge needs to wait for the completion of maximum Trickle interval of its parent to receive a DIO packet.

The works such as References [27–30] have been published to improve the performance of RPL. For this, the authors tried to transmit the DIO packet immediately for faster formation of DODAG. These works mainly tune various parameters such as node-wise varied idle listening time and redundancy constant and perform encapsulation of DIO packet into the EB frame. However, all these works do not consider the quick generation of a DIO packet when it requires during network formation. These works follow the same DIO generation interval set by the TA. Hence, following these approaches, a pledge might need to wait for maximum Trickle interval time (I_{max}) to get a DIO packet. The problem of long waiting for DIO packet during its requirement still persists in these existing works.

3.4 Delayed Channel Access for Control Frames

We observed that the packet-level starvation and negative effect of Trickle Algorithm problem occur inside a node, and thus they can be solved locally within a node. However, the 6TiSCH-MC uses random channel access protocol (i.e., CSMA/CA) to send control packets in shared slots. Although, in the long run or in steady state condition, random channel access provides fairness to all the contending nodes, but it does not give guarantee that a node can access the channel within a minimum time period. Again, with the increasing number of joined nodes, uncertainty in accessing shared channel also increases. In a dense network, a joined node may need to wait for several shared slots to send its control packet, which is already present in its buffer. Furthermore, nodes perform random backoff once they encounter busy channel. When the congestion increases, possibility of getting into the backoff stage also increases. This results in delay in the transmission of control packets, and sometimes, packets are also get discarded after the maximum number of re-tries. Because of this reason, a pledge has to wait for long time to get the required control packets. This results in increasing joining time for pledges, which increases the overall network formation time. None of the existing works considered this problem during network formation. As this problem occurs within a network, therefore, an inter-node solution helps in reducing the overall network formation time of the 6TiSCH network.

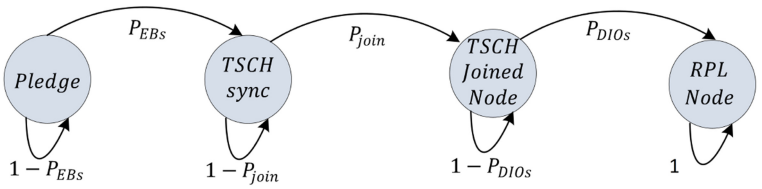


Fig. 3. Markov Chain model of node joining process.

3.5 Objective of This Article

In brief, none of the existing works considered the above-mentioned three problems in the context of 6TiSCH network formation. Therefore, in this article, we try to overcome all these shortcomings with the aim of faster formation of 6TiSCH network. In 6TiSCH network, the EB frame is used to broadcast network existence information. Furthermore, EB frame helps in maintaining synchronization in a network. For these reasons, the highest priority is given to EB frame among all the control packets used in 6TiSCH network. Therefore, it would not be suitable for a 6TiSCH network to consider all types of control packets with equal priority. Again, Trickle Algorithm–based DIO-interval is used to save node energy in RPL-based LLN. Trickle Algorithm suppresses the unnecessary transmission of frequent DIO packet to save energy. Hence, we cannot use fixed rate DIO interval in an LLN because of its energy constraint. Therefore, to deal with the above-mentioned first two problems, we need to dynamically assign priority to control packets based on their network requirement and maintain the rate of sending routing information based on pledge's requirement. Again, to deal with the uncertainty in accessing transmission channel, there should be some mechanism by which the node having important packet(s) in its transmission buffer can access the shared channel before the nodes that do not have important control packets to transmit immediately.

3.6 Analytical Validation of the Problems

For validation of our problem statements, initially, we analyze the 6TiSCH network formation procedure considering the similar resource allocation scheme as described in 6TiSCH-MC following secure enrollment of nodes during network formation. Hence, all bootstrapping traffics such as EB, DIO, JRS, JRQ are considered to be transmitted in shared cell during network formation. We model the node's joining time using Markov Chain as described in Reference [22]. And using the analytical model, we validate our problem statements one by one. Hence, for completeness of the article, that model is described briefly here also. The symbols, used in the analytical model, and their corresponding meaning are tabulated in Table 1.

During network formation, each pledge proceed through four states. Figure 3 shows the different states of the Markov model including an absorbing state. The absorbing state indicates that the pledge has joined with the existing network successfully. That means a pledge has become an RPL joined node and now it can send its own beacon for further expansion of the network. Here, formation of multi-hop 6TiSCH network (as shown in Figure 4) is considered, where all the pledges join the network one by one. The network formation is completed once all the pledges become RPL joined nodes.

In the initial state, a pledge waits for an EB from JRC/RPL root or already joined node. After receiving a valid EB frame with probability P_{EB_S}, a pledge moves to second state. Now, a pledge waits to complete its secure enrollment after synchronizing in the TSCH network. For secure enrollment, a pledge exchanges JRQ and JRS frame with its parent. The probability of finishing the secure enrollment process and reaching the third state by a pledge is P_{join}. In the third state,

Table 1. List of Frequently Used Symbols

Symbol	Meaning
N_c	Number of channels used
L	Slotframe length
I_{eb}	Beacon generation interval
P_{eb}	Transmission probability of a EB frame in a shared cell
P_{EB_S}	Successful transmission probability of a EB frame in a shared cell
P_{dio}	Transmission probability of a DIO packet in a shared cell
P_{DIO_S}	Successful transmission probability of a DIO packet in a shared cell
P_{jrq}	Transmission probability of a JRQ frame in a shared cell
P_{JRQ_S}	Successful transmission probability of a JRQ frame in a shared cell
P_{jrs}	Transmission probability of a JRS frame in a shared cell
P_{JRS_S}	Successful transmission probability of a JRS frame in a shared cell
P_{loss}	Packet loss probability
ASF	Average number of slotframes to join a network
M	Number of neighbor nodes of a pledge
n	Number of neighbor joined nodes of a pledge
AJT	Average joining time of a node
PJT	Average joining time of parent node
P_r	Trickle algorithm reset probability
N_D	Total number of Trickle algorithm states
P_{ebN}	Transmission probability of a EB frame in a shared cell using proposed scheme OPR
P_{dioN}	Transmission probability of a DIO packet in a shared cell using proposed scheme OPR

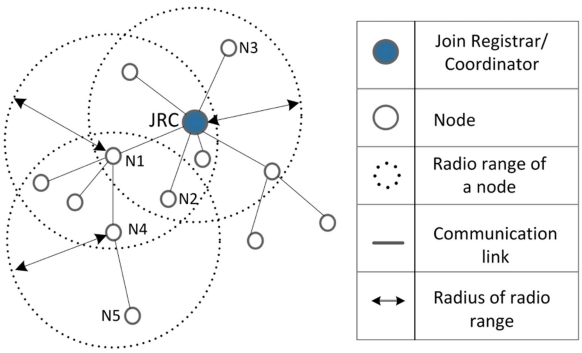


Fig. 4. Topology considered for analysis.

a pledge is waiting for routing protocol information to complete its joining process. And after receiving a valid RPL DIO packet, the node moves to its final absorbing state. The absorbing state denotes the successful joining of a pledge into an existing 6TiSCH network. Again, if we consider that the probability of receiving a DIO packet successfully is P_{DIO_S}, then the transition probability matrix of the Markov model can be written as follows:

$$M = \begin{bmatrix} 1 - P_{EB_S} & P_{EB_S} & 0 & 0 \\ 0 & 1 - P_{join} & P_{join} & 0 \\ 0 & 0 & 1 - P_{DIO_S} & P_{DIO_S} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Using the Markov model, the average number of slotframes (ASF) requires to reach the final absorbing state can be computed as follows:

$$ASF = \frac{1}{P_{EB_S}} + \frac{1}{P_{join}} + \frac{1}{P_{DIO_S}} = \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRS_S}} + \frac{1}{P_{JRQ_S}}\right) + \frac{1}{P_{DIO_S}}. \quad (1)$$

Now, to get the actual value of ASF, we need to calculate the values of P_{EB_S}, P_{DIO_S}, P_{JRS_S}, and P_{JRQ_S}. Let us consider a multi-hop network model as shown in Figure 4, where each node has different number of neighbors. Let P_{eb}, P_{dio}, P_{jrq}, P_{jrs} be the transmission probabilities of EB, DIO, JRQ, and JRS frames/packets in a slotframe, respectively. P_{loss} is the packet loss probability and N_c is the total number of channels used in the network. The probability of receiving an EB frame successfully in a shared cell by a pledge is computed as,

$$P_{EB_S} = \frac{\sum_{n}^{M-n} nP_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n)}{N_c}, \quad (2)$$

where n is the number of joined neighbor nodes, and M is the total neighbor nodes of a pledge. The value of n and M can be different for every node present in a network. The equation follows that the $(M-n)$ pledges join a network one by one, and the $P(N=n)$ denotes that at an instant, the probability of total neighbor joined node is n. The value of $P(N=n)$ is calculated considering uniform probability distribution, which equals to $\frac{1}{M}$. It is because the final joining time of a pledge is calculated considering all its the neighbor nodes join the network one by one and the occurrence of each joined node has a constant and equal probability. Again, a joined node can send its EB frame only when the remaining $(n-1)$ joined nodes do not send any control packet and the neighbor pledges also do not send any JRQ frames. The probability of these conditions equals to $(P_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1})$ and $((1-P_{jrq})^{M-n})$, respectively. The $(1-P_{loss})$ denotes that the transmitted frame is not lost in the channel. Finally, a pledge is not synchronized with its coordinator's channel-hopping sequence at the beginning. Therefore, it searches in all the available N_c number of channels, which reduces the EB success probability by N_c times. Further, the EB frames can be transmitted by any of the neighbors n RPL joined nodes, which result in multiplying the computed probability by n. Note that a pledge can send JRQ frames only after it receives a valid EB frame, and an RPL joined node sends JRS frame only after it receives a valid JRQ frame. So, $P_{jrq} = P_{EB_S}$ and $P_{jrs} = P_{JRQ_S}$.

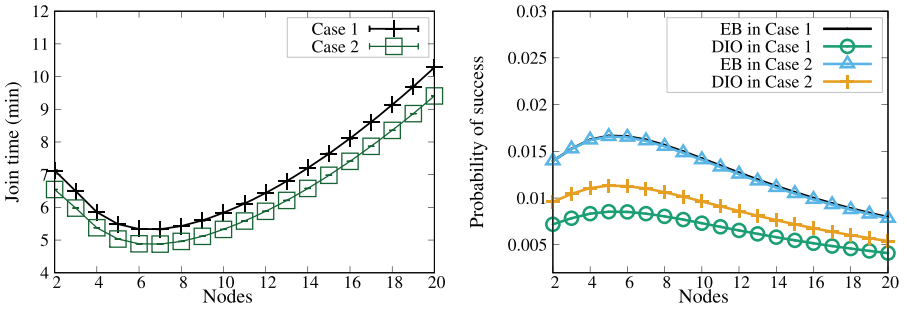
Considering that EB frame has higher priority over DIO packet, the probability of sending a DIO packet successfully in a shared cell is computed as,

$$P_{DIO_S} = \sum_{n}^{M-n} nP_{dio}(1-P_{dio})^{n-1}(1-P_{eb})^{n-1}(1-P_{jrs})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n). \quad (3)$$

Note that, in the above equation, N_c is not used, because a node knows its coordinator's channel-hopping sequence after getting an EB frame. Likewise, the probability of sending a JRQ frame and receiving a JRS frame successfully in a shared cell is also computed as,

$$P_{JRQ_S} = \sum_{n}^{M-n} (M-n)P_{jrq}(1-P_{jrq})^{M-n-1}(1-P_{eb})^{n}(1-P_{dio})^{n}(1-P_{jrs})^{n}(1-P_{loss})P(N=n), \quad (4)$$

$$P_{JRS_S} = \sum_{n}^{M-n} nP_{jrs}(1-P_{jrs})^{n-1}(1-P_{eb})^{n}(1-P_{dio})^{n}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n). \quad (5)$$



(a) Node join time (b) Probability of success

Fig. 5. Comparison on node joining time and the success probability of different control packets. (*Case 1*: EB has highest probability, *Case 2*: EB has equal probability with others)

Now, to compute Equations (2), (3), (4), and (5), we should know the transmission probabilities of EB and DIO in a shared cell. The probability P_{eb} can be calculated as,

$$P_{eb} = \frac{L}{I_{eb}}, \tag{6}$$

where L is the slotframe length and I_{eb} is the beacon interval. Note that the value of I_{eb} is always greater than L. Otherwise, EB frame will always replace the other control packets in the transmission buffer because of its highest priority [18]. We compute the probability P_{dio} following the procedure described in Reference [18] as follows:

$$P_{dio} = (1 - P_{eb}) \frac{2^{N_D}(1-P_r)^{N_D} min(\frac{L}{2^{N_D} I_{min}}, 1) + \sum_{i=0}^{N_D-1} P_r 2^i (1-P_r)^i min(\frac{L}{2^i I_{min}}, 1)}{P_r + 2^{N_D}(1-P_r)^{N_D} + \sum_{j=1}^{N_D-1} P_r 2^j (1-P_r)^j}, \tag{7}$$

where I_{min} is the initial DIO-interval, P_r is the Trickle Algorithm reset probability, and N_D represents the total number of Trickle algorithm states. Finally, we compute the average joining time (AJT) of a pledge as follows:

$$AJT = PJT + ASF \times L, \tag{8}$$

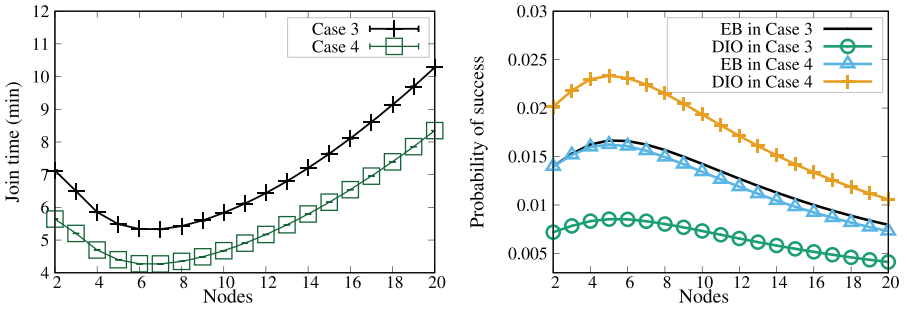
where PJT is the average joining time of parent node of a pledge in multi-hop network. For example, in Figure 4, joining time of node $N4$ is the summation of its own joining time and its parent's joining time (i.e., joining time of node $N1$ or $N2$ whichever $N4$ selects as its parent).

Likewise, joining time of each pledge in a network can be calculated, which depends on its parent's joining time and number of its neighbors. The formation time of an entire network is the joining time of the node added last in the network.

Now, to validate packet-level starvation, we compare the results of network formation method considering EB frame has the highest priority in one case (say, Case 1), and in the other case (say, Case 2), we consider all the control packets have equal probabilities. The probability of sending a DIO packet does not depend on the existence of an EB frame in node buffer for Case 2. So, the probability of sending a DIO packet in a shared cell in Case 2 can be derived from Equation (7) as follows:

$$P_{dio} = \frac{2^{N_D}(1-P_r)^{N_D} min(\frac{L}{2^{N_D} I_{min}}, 1) + \sum_{i=0}^{N_D-1} P_r 2^i (1-P_r)^i min(\frac{L}{2^i I_{min}}, 1)}{P_r + 2^{N_D}(1-P_r)^{N_D} + \sum_{j=1}^{N_D-1} P_r 2^j (1-P_r)^j}. \tag{9}$$

Now, let us consider a network with the following given values: $N_C = 16$, $P_{loss} = 0.2$, $I_{eb} = 4 * L$, $T_i = 10\ ms$, $I_{min} = 8\ ms$, $L = 101\ timeslots$, $N_D = 16$, and $P_r = 0.2$. We plot a graph in Figure 5



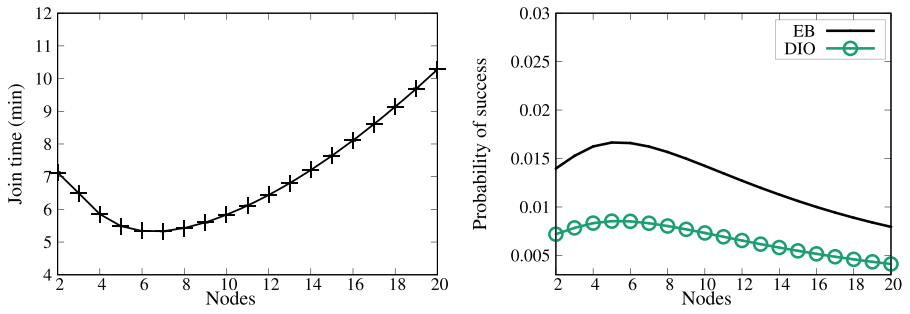
(a) Node join time (b) Probability of success

Fig. 6. Comparison on node joining time and the success probabilities of different control packets. (*Case 3*: Trickle Algorithm based DIO-interval, *Case 4*: fixed DIO-interval)

considering both Case 1 and Case 2. The graph shows that the overall joining time of the network is less in Case 2. The reason is shown in Figure 5(b). It shows how the probability of successfully transmitting a DIO packet in a shared cell increases when EB has equal priority with other control packets. The increase in success probability of DIO packet decreases the average network joining time of the pledges in the network. However, considering EB frame has the highest priority (Case 1), less successful transmission probability of DIO packet can be seen as EB frame suppresses the transmission of DIO packet because of its highest priority. This results in increasing joining time of the nodes.

To validate the problem exists in Trickle Algorithm (TA), we use fixed DIO-interval, i.e., $P_{dio} = .03$ instead of using TA based interval and plot the results in Figure 6 along with TA-based DIO-interval. In the figure results using TA-based DIO-interval and fixed DIO-interval are shown as Case 3 and Case 4, respectively. The figure shows that there is significant improvement in node joining time when we do not allow TA to update the DIO packet transmission interval. Figure 6(b) shows that the probability of successfully transmitted DIO control packet in a shared cell can be improved without using TA. TA decreases the number of transmitted DIO packets in a stable network by increasing DIO generation interval to save energy. However, the decreasing number of transmitted DIO packets has negative effect on the reception probability of DIO packets in a shared cell. Hence, pledges have to wait for longer time for DIO packets, which in turn increases the overall network joining time. Figure 6 shows how TA impacts on the network joining time (Case 3). Note that, here, a stable network means there are no changes in the network topology at the current state.

To validate the third problem statement, we plot the node joining time with respect to increasing number of joined nodes using the Equation (1) in Figure 7. This figure shows the impact of increasing uncertainty of shared channel access due to increasing number of nodes on network joining time considering TA-based DIO packet generation. Initially, when the number of nodes is small (e.g., 2–5 nodes), the pledges need to wait for more amount time because of infrequent transmission of control packets in the network. Therefore, addition of few more nodes (e.g., 6–8 nodes) helps the pledges to join the network in less time because of more number of transmission of control packets by more nodes without congesting the shared channel. However, further increment of nodes (e.g., 9–20 nodes) increases the congestion in the shared channel, which in turn increases the uncertainty in accessing the shared channel. This results in reduction in channel access probability, which increases the joining time of the pledges, as pledges need to wait for more amount of time for control packets. Further, Figure 7 shows that the successful transmission probability for both the EB frame and DIO packet in a slotframe decreases with the increasing number of nodes.



(a) Join time comparison (b) Probability of success

Fig. 7. Impact of number of joined nodes on average network formation time and success probability of control packets.

4 PROPOSED SOLUTION

We propose two schemes to solve all the above three problems. The first one, opportunistic priority alternation and rate control (OPR) scheme, solves the problem of packet-level starvation and negative effect of Trickle Algorithm. The second one is used to provide channel access to the nodes having urgent packets by designing opportunistic channel access (OCA) scheme. Both the schemes are described below.

4.1 Opportunistic Priority Alternation and Rate Control Scheme

The OPR scheme changes the priorities of control packets depending upon network requirements and also opportunistically increases the rate of transmission of DIO packets once it is required. The main purpose of EB frame is to broadcast network availability information and maintain global synchronization within a network. Therefore, in stable networks, we always consider EB must have the highest priority. However, in few cases (e.g., when joined node receives JRQ frame), transmission of DIO packet is more critical than EB frame. In such cases, the proposed scheme gives the highest priority to DIO packet for reducing overall network formation time. Again, the TA based DIO-interval affects the performance of a node in terms of energy consumption. So, we modified the existing TA with little modification such that it resets its interval during network inconsistency as well as when a joined node receives JRQ frame from a pledge. Algorithm 1 describes the OPR scheme.

4.2 Opportunistic Channel Access Scheme

The OCA scheme provides a solution to any node having an urgent packet in its transmission buffer. The DIO packet generated by a joined node after receiving a JRQ frame or DIS packet from its child node is considered as an urgent packet. It is because the child node requires DIO packet immediately to complete its joining process in less time. And the node that contains such an urgent (DIO) packet is considered as an urgent node. As all the nodes present in a network try to access the shared channel at a time, there is a requirement of inter-node solution for opportunistic channel access in association with random channel access. A node does not know whether the neighbor nodes have urgent packets to send or not. Therefore, we need to modify the default channel access mechanism (i.e., CSMA/CA) for providing opportunistic access to nodes. In this modified scheme, we propose that the nodes, which have an urgent packet to transmit, use fixed contention window (CW) even if they sense busy channel during CCA. In other words, the CW size does not increase after an unsuccessful attempt in case of urgent packet transmission. However, the normal nodes,

ALGORITHM 1: Opportunistic Priority Alternation and Rate Control Scheme (OPR)

```
1: if the current Link_type is Shared then
2:     if got the channel access to transmit then                    ▷ Tx mode
3:         Transmit highest priority control packet (EB/DIO) from the buffer
4:         Set highest priority to EB (default value)
5:         Set flagOCA to FALSE                                      ▷ for Algorithm 2
6:     end if
7:     if received a packet then                                     ▷ Rx mode
8:         if received packet is JRQ frame then
9:             Send Join Response (JRS) frame
10:            Set highest priority to DIO
11:            if no DIO packet is available in transmission buffer then
12:                Reset the Trickle interval for generation of DIO
13:                Set flagOCA to TRUE                               ▷ for Algorithm 2
14:            end if
15:        end if
16:        if received packet is DODAG Information Solicitation (DIS) request for DIO then
17:            Set highest priority to DIO
18:            Set flagOCA to TRUE                                   ▷ for Algorithm 2
19:        end if
20:    end if
21: end if
```

which do not have any urgent packet to transmit, use variable CW when they encounter busy channel. In other words, normal nodes calculate their CW size using the default backoff algorithm. Algorithm 2 describes the procedure of opportunistic channel access.

4.3 Theoretical Analysis of Proposed Schemes

In this section, we perform theoretical analysis of the proposed schemes. As OPR solves the intra-node problem and OCA solves the inter-node problem within a network, separate theoretical modelings of both the schemes are performed and shown in the sections below.

4.3.1 Opportunistic Priority Alternation and Rate Control. OPR assigns higher priority to a DIO packet than an EB frame when there is a need for DIO in the network. However, in a stable network, EB frame always has the highest priority over all other control packets. Therefore, the probability of sending an EB frame in a shared cell by our proposed OPR scheme (i.e., P_{ebN}) also depends on the requirement and availability of DIO packet in the network. Hence, P_{ebN} can be calculated as follows:

$$P_{ebN} = (1 - P_{JRQ_S})P_{eb} + P_{JRQ_S}(1 - P_{dioG})P_{eb}, \tag{10}$$

where the term $(1 - P_{JRQ_s})P_{eb}$ denotes that no JRQ frame is received, which indicates no urgent requirement of DIO packet. So, P_{ebN} depends on the probability of generation of EB frame in a slotframe, i.e., P_{eb}. Furthermore, $P_{JRQ_s}(1 - P_{dioG})P_{eb}$ denotes that JRQ frame is received (i.e., urgent requirement of DIO packet) but no DIO packet is available in node's buffer. Here, P_{dioG} denotes the probability of generating a DIO packet in a slotframe and it is calculated using the Equation 15. Hence, EB sending probability depends on the generation of an EB frame in a slotframe. Note that the value of P_{ebN} always will be in between $[0, 1]$. This is because the values of P_{ebN} depend on the values of P_{JRQ_S}, P_{dioG}, and P_{eb}. The values of these variables always remain in between $[0, 1]$, as these are the sending probabilities of different control packets in a shared cell.

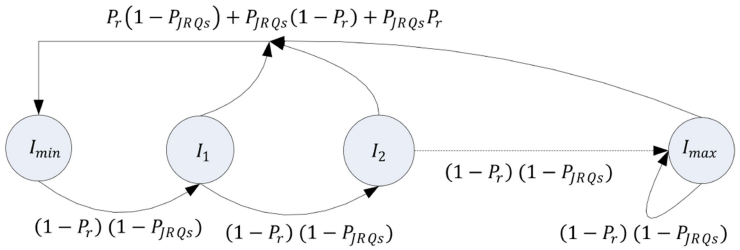


Fig. 8. State transition diagram of DIO generation interval in modified Trickle Algorithm.

ALGORITHM 2: Opportunistic Channel Access Scheme (OCA)

Initialization: NB (number of backoff stages) =0; BE (backoff exponent) =*macMinBE*; ω (random backoff counter) =0
if the current $Link_{type}$ is *Shared* AND ω is not 0 **then**
    ω is decremented by 1
else if the current $Link_{type}$ is *Shared* AND ω *is* 0 **then**
    Node perform Clear Channel Assesment (CCA)
    if channel is free **then**
        Transmit the frame;
        BE=*macMinBE*;
        NB=0;
        Exit
    else if (NB+1) greater than *macMaxFrameRetries* **then**
        Drop the frame and Exit
    else if flagOCA is FALSE **then** ▷ i.e., packet is not urgent and flagOCA is obtained from Algorithm 1
        BE=min(BE+1, *macMaxBE*)
    end if
    Increment of NB by unity;
    Select a random backoff counter $\omega \in [0, 2^{BE}\text{-}1]$
end if
Goto step 2

The generation of DIO packet is controlled by our modified Trickle Algorithm. Hence, the new probability of sending a DIO packet in a shared cell (i.e., P_{dioN}) depends on the reception of JRQ frame from a pledge and the network status. We follow the similar Semi-Markov Process (SMP) approach mentioned in Reference [18] to calculate the value of P_{dioN}. The main difference between the approach in Reference [18] and our approach is the consideration of the non-deterministic JRQ frame along with the network status. The modified Trickle Algorithm of a joined node resets its current interval to minimum value I_{min} during network inconsistency as well as upon receiving a JRQ frame. Figure 8 shows the states of Markov process based on modified TA.

In the figure, the probability P_r denotes the inconsistency in networks. P_{JRQ_S} denotes the probability of successfully receiving a JRQ frame in a slotframe. Therefore, when neither any network inconsistency nor any JRQ frame is received, Trickle state moves from its I_i state to I_{i+1} state. When Trickle state reaches its final state I_{max}, it remains in the same state until there is an event to reset it. In brief, the modified TA resets its DIO-interval to minimum in three conditions:

- When there is inconsistency in the network but does not receive any JRQ frame, i.e., $P_r(1 - P_{JRQ_S})$;

- When there is no inconsistency in the network but receives JRQ frame, i.e., $(1 - P_r)P_{JRQ_S}$;
- When there is inconsistency in the network and also receives JRQ frame, i.e., $P_r P_{JRQ_S}$.

Therefore, the TA resetting probability can be written as: $P_r(1 - P_{JRQ_S}) + (1 - P_r)P_{JRQ_S} + P_r P_{JRQ_S} = P_r + P_{JRQ_S} - P_r P_{JRQ_S}$.

Considering all these conditions, the probability matrix of the Markov process can be written as follows:

$$M = \begin{bmatrix} x & y & 0 & \cdots & 0 \\ x & 0 & y & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x & 0 & 0 & \cdots & y \end{bmatrix},$$

where $x = P_r + P_{JRQ_S} - P_r P_{JRQ_S}$ and $y = (1 - P_r)(1 - P_{JRQ_S})$

Now, having the sojourn time as T_i in state i, the stationary distribution of the state in the embedded Markov chain can be computed as follows:

$$\pi_i = \frac{\mu_i T_i}{\sum_{j=0}^{N} \mu_j T_j},$$

where $\mu = P\mu$. Now, applying this in our Markov process, we get,

$$\pi_0 = \frac{x}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j}, \tag{11}$$

$$\pi_i = \frac{x 2^i y^i}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j}, i \in [1, N_D - 1], \tag{12}$$

$$\pi_{N_D} = \frac{2^{N_D} y^{N_D}}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j}. \tag{13}$$

Again, the average probability of generating a DIO packet in a state i can be calculated as follows:

$$P(P_{dioG} \mid I_i) = \begin{cases} 1 & \text{if L} \geq I_i, \\ \frac{L}{I_i} & \text{otherwise,} \end{cases} \tag{14}$$

where I_i is the duration of state i. Hence, the average probability of generating a DIO packet in a slotframe, i.e., P_{dioG}, can be calculated as follows:

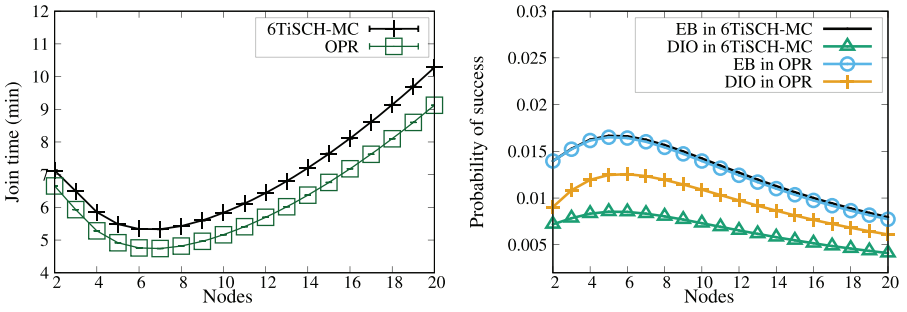
$$P_{dioG} = \sum_{i=0}^{N_D} \pi_i P(P_{dioG} \mid I_i) = \frac{2^{N_D} y^{N_D} min(\frac{L}{2^{N_D} I_{min}}, 1) + \sum_{i=0}^{N_D - 1} x 2^i y^i min(\frac{L}{2^i I_{min}}, 1)}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j}. \tag{15}$$

Hence, probability of sending DIO packet in a slotframe (i.e., P_{dioN}) can be computed as follows:

$$P_{dioN} = (1 - P_{JRQ_S}) \left(1 - \frac{L}{I_{eb}}\right) P_{dioG} + P_{JRQ_S} P_{dioG}. \tag{16}$$

Here, the term $(1 - P_{JRQ_S})(1 - \frac{L}{I_{eb}})P_{dioG}$ denotes that a joined node does not receive JRQ frame and there is no EB frame available in its buffer, and hence, node can send DIO packet. $P_{JRQ_S} P_{dioG}$ denotes that node send its available DIO packet when it receives JRQ frame without conditioning availability of EB frame.

Now, by using the value of P_{ebN} and P_{dioN} in Equations (1–8), we can calculate the average joining time (AJT) of pledge. However, we notice that there exists a circular dependency among the Equations (8), (10), and (16). Therefore, an iterative approach is used to compute the steady state AJT of a pledge. At the beginning, only EB and DIO packets are transmitted, therefore, the initial values of P_{EB_s} and P_{JRQ_s} are calculated using the Equations (6) and (7) and taking the initial



(a) Node join time (b) Probability of success

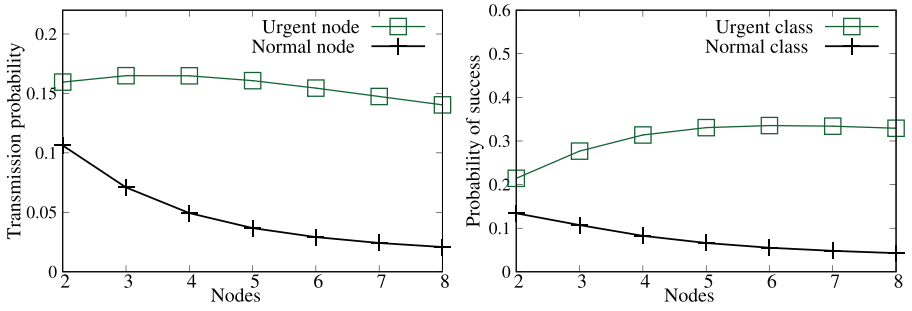
Fig. 9. Comparison on node joining time and the success probabilities of different control packet.

values of P_{jrq} and P_{jrs} as 0 to break the circular dependency. Now, assigning $P_{jrq} = P_{EB_S}$ and $P_{jrs} = P_{JRQ_S}$ and using Equations (4), (6), and (7), we calculate new value of P_{JRQ_S}. Using this calculated P_{JRQ_S} value, we calculate the transmission probability of EB frame P_{ebN} and DIO packet P_{dioN} in a shared cell based on our proposed method. In the next step, we calculate the AJT by using the new values of P_{ebN} and P_{dioN} in Equation (8). Simultaneously, we also obtain the value of P_{JRQ_S} using Equation (4). Using the new value of P_{JRQ_S}, again, we calculate new values of P_{ebN} and P_{dioN}. Again, this newly calculated P_{ebN} and P_{dioN} are used to calculate the AJT and P_{JRQ_S}. The processes are repeated until a steady average joining time is obtained. The final obtained results are plotted in Figure 9. Note that, in this calculation process, the values of other network parameters, such as I_{eb}, P_r, and so on, are taken same as mentioned in Section 3.6. Figure 9(a) shows the improvement in average joining time of a pledge of the proposed scheme over 6TiSCH-MC. And Figure 9(b) shows the reason for this improvement. In the proposed scheme, the negative effect of EB's highest priority and TA are eliminated, and thus, there are improvements in success probabilities of different control packets. As a result of this, average joining time of a network is reduced.

4.3.2 Opportunistic Channel Access. OCA provides higher chance to access the shared transmission channel to the nodes who have urgent packets to send by allowing them to use fixed CW even if they sense busy channel during CCA. The size of the fixed CW is smaller than the CW of other normal node who have experienced busy channel. For analyzing OCA scheme, we consider two classes of nodes. One is urgent priority class, which includes the nodes who have urgent packet to send. The other priority class contains the normal nodes that do not have urgent packet to send. Xiao et al. [34] and Khatua et al. [35] analyzed the random access MAC protocol with different priority classes of nodes. According to Reference [34], the transmission probability τ_i of a node of priority class i in a generic slot can be computed as follows:

$$\tau_i = \frac{1 - {P_i}^{m+1}}{1 - P_i} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_i} \sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}\right] {P_i}^j}, \tag{17}$$

where m is the maximum retries, P_i denotes the probability of a node from priority class i senses busy channel, $W_{i,j}$ denotes the CW size of a node of priority class i in its jth backoff stage. From Equation (17), we can derive the transmission probabilities τ_u and τ_n of an urgent node and a



(a) Transmission probability (b) Probability of success

Fig. 10. (a) Transmission probabilities of an urgent node and a normal node and (b) successful transmission probabilities of whole urgent node class and normal node class.

normal node, respectively, as follows:

$$\tau_u = \frac{1 - {P_u}^{m+1}}{1 - P_u} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_u} \sum_{k=1}^{W_{u,j}-1} \frac{W_{u,j}-k}{W_{u,j}}\right] {P_u}^j}, \tag{18}$$

$$\tau_n = \frac{1 - {P_n}^{m+1}}{1 - P_n} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_n} \sum_{k=1}^{W_{n,j}-1} \frac{W_{n,j}-k}{W_{n,j}}\right] {P_n}^j}. \tag{19}$$

Note that in Equation (18) the value of $W_{u,j}$ remains same as we use fixed CW, whereas, in Equation (19), $W_{n,j}$ varies as $W_{n,j+1} = \alpha W_{n,j}$, where α is a CW incremental factor.

The value of P_u and P_n, i.e., collision probabilities of an urgent node and a normal node, can be computed as follows:

$$P_u = 1 - (1 - \tau_u)^{N_u - 1}(1 - \tau_n)^{N_n}, \tag{20}$$

$$P_n = 1 - (1 - \tau_n)^{N_n - 1}(1 - \tau_u)^{N_u}, \tag{21}$$

where N_u and N_n are the number of urgent nodes and normal nodes, respectively.

Again, Equations (18), (19), (20), and (21) have circular dependency among themselves. Therefore, once again an iterative approach is used to solve this dependency, as mentioned in the previous section. For this purpose, the initial values of τ_u and τ_n are assumed randomly as 0.5. The other initialization parameters $W_{i,0} = 4$, $m = 5$ are taken same for both the classes of nodes. For normal node, we take $\alpha = 2$. In the first step, we calculate the P_u and P_n using τ_u and τ_n. These calculated P_u and P_n values are used to calculate new values of τ_u and τ_n. Again, these new values of τ_u and τ_n are used to further calculate the new values of P_u and P_n. We continue this process until we reach to a steady state, i.e., there is no significant change in the values of τ_u and τ_n.

Let P^u_{succ} and P^n_{succ} denote the probabilities that a successful transmission occurs in a slot from the class of urgent node and normal node, respectively. The value of P^u_{succ} and P^n_{succ} can be computed as:

$$P^u_{succ} = N_u \tau_u (1 - \tau_u)^{N_u - 1}(1 - \tau_n)^{N_n}, \tag{22}$$

$$P^n_{succ} = N_n \tau_n (1 - \tau_n)^{N_n - 1}(1 - \tau_u)^{N_u}. \tag{23}$$

The ratio $\frac{P^u_{succ}}{P^n_{succ}}$ denotes the opportunistic channel access by the class of urgent nodes over normal nodes. The propositions 4.1 and 4.2 prove that $\frac{P^u_{succ}}{P^n_{succ}} > 1$. This, in turn, confirms that the proposed scheme provides opportunistic channel access to urgent packet containing nodes.

Figure 10(a) shows that an urgent node obtains higher transmission probability because of non-increasing small CW. And Figure 10(b) shows the opportunistic successful transmission probabilities of the urgent node class against the whole normal node class.

Proposition 4.1. After first backoff stage, OCA scheme provides $\tau_u > \tau_n$ *always.*

Proof. From Equation (17), by expanding the term $\sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}$, we get,

$$\begin{aligned} &= \frac{W_{i,j}-1}{W_{i,j}} + \frac{W_{i,j}-2}{W_{i,j}} + \cdots\cdots + \frac{W_{i,j}-(W_{i,j}-1)}{W_{i,j}} \\ &= \frac{W_{i,j}-1}{2}. \end{aligned} \tag{24}$$

Hence, the value of τ is reciprocal to CW size. At the first stage, the values of $W_{u,1}$ and $W_{n,1}$ are same in Equations (18) and (19). Hence, we get same values for τ_u and τ_n, which, in turn give same values for P_u and P_n. But in the later backoff stages, say, stage 2, $i = 2$, $W_{n,2} > W_{u,2}$, $\tau_u = \tau_n$ and $P_u = P_n$ and all these values result in $\tau_u > \tau_n$ and $P_u < P_n$. This is because the term $\sum_{j=0}^{m}[1 + \frac{1}{1-P_i}\sum_{k=1}^{W_{i,j}-1}\frac{W_{i,j}-k}{W_{i,j}}]{P_i}^j$ in Equation (17) increases with larger values of CW. And the larger value of the term $\sum_{j=0}^{m}[1 + \frac{1}{1-P_i}\sum_{k=1}^{W_{i,j}-1}\frac{W_{i,j}-k}{W_{i,j}}]{P_i}^j$ reduces the value of τ_i and increases P_i. In the later backoff stages also $\tau_u > \tau_n$ for $W_{n,i} > W_{u,i}$. □

Proposition 4.2. After first backoff stage, OCA scheme provides $P^u_{succ} > P^n_{succ}$ *always.*

Proof. If we calculate the ration $\frac{P^u_{succ}}{P^n_{succ}}$, we get:

$$\begin{aligned} \frac{P^u_{succ}}{P^n_{succ}} &= \frac{N_u\tau_u(1-\tau_u)^{N_u-1}(1-\tau_n)^{N_n}}{N_n\tau_n(1-\tau_n)^{N_n-1}(1-\tau_u)^{N_u}} \\ &= \frac{N_u\tau_u(1-\tau_n)}{N_n\tau_n(1-\tau_u)}. \end{aligned} \tag{25}$$

We already show that if $W_{n,i} > W_{u,i}$, then it results $\tau_u > \tau_n$. From the Equation (25), it is clear that $\frac{P^u_{succ}}{P^n_{succ}} > 1$ when $\tau_u > \tau_n$. So, $P^u_{succ} > P^n_{succ}$. □

5 SIMULATION EXPERIMENTS

5.1 Simulation Setup

The performance of the proposed schemes is investigated by using the Cooja simulator on Contiki-4.5 operating system [36]. The used parameters and their corresponding values are mentioned in Table 2. In our evaluation, the nodes are deployed in a fixed square size grid area where join registrar/coordinator (JRC), i.e., RPL root, is always placed at the top left corner of the grid. Increment of the number of nodes in a fixed area results in increasing node density. As a result, congestion in the network also changes with varying number of nodes. The JRC initializes network formation by broadcasting network advertisement information carrying EB frame and routing information carrying DIO packet. The realistic Multipath Ray-tracer Medium (MRM) channel model is used in simulations. This model provides various propagation effects such as multi-path, refraction, and diffraction in the physical channel. Along with the EB frame and various RPL control packets, few other control packets, such as JRQ, JRS, and keep-alive, are also considered during our evaluation.

5.2 Benchmark Schemes and Performance Metric

We compared our proposed scheme, i.e., opportunistic transmission of control packets (OTCP) with the existing benchmark schemes 6TiSCH-MC [10] and BS [19]. Note that the OTCP is nothing but

Table 2. Simulation Parameters

Parameter	Value
Operating System	Contiki-4.5
Mote type	Cooja Mote
Network size	16, 25, 36, 49 nodes
Timeslot length (L)	10 ms
Slotframe size (SF)	33, 101 timeslots
Number of channels	16
Propagation model	MRM
EB interval	4 sec
RPL version	RPL Lite
RPL DIO-interval	Trickle interval
DIO interval (min-max)	$2^{10} - 2^{18}$, $2^{12} - 2^{20}$, $2^{13} - 2^{21}$, $2^{14} - 2^{22}$ ms
RPL Objective Function	MRHOF - ETX
RPL Redundancy constant	10
Keep-alive interval	30 sec
Simulation duration	2 hours

the execution of both the OPR and OCA schemes together. We also show how OPR and OCA schemes work individually and compare them with the benchmark schemes.

Mainly, three performance metrics, such as TSCH joining time, 6TiSCH joining time, and maximum energy consumption, are considered in evaluation. The TSCH joining time denotes the maximum time required to receive at least one EB frame (i.e., TSCH synchronization time) by a pledge among all the pledges in the network. In other words, maximum time required by the last pledge to reach to TSCH sync state of the Markov chain, as shown in Figure 3. The 6TiSCH joining time denotes the maximum time required to receive at least one EB frame followed by exchanging of JRQ and JRS frames and receiving a valid DIO packet by a pledge among all the pledges present in the network. Similar with TSCH joining time, the 6TiSCH joining time is the maximum time required by the last pledge to reach to RPL node state of the Markov chain, as shown in Figure 3. The maximum energy consumption by the last pledge admitted in the network is shown using the energy consumption metric. The procedure of energy consumption computation is described below. Basically the energy consumption by a node is the summation of energy consumption during communication (E_{Comm}) and energy consumption during CPU active state (E_{CPU}).

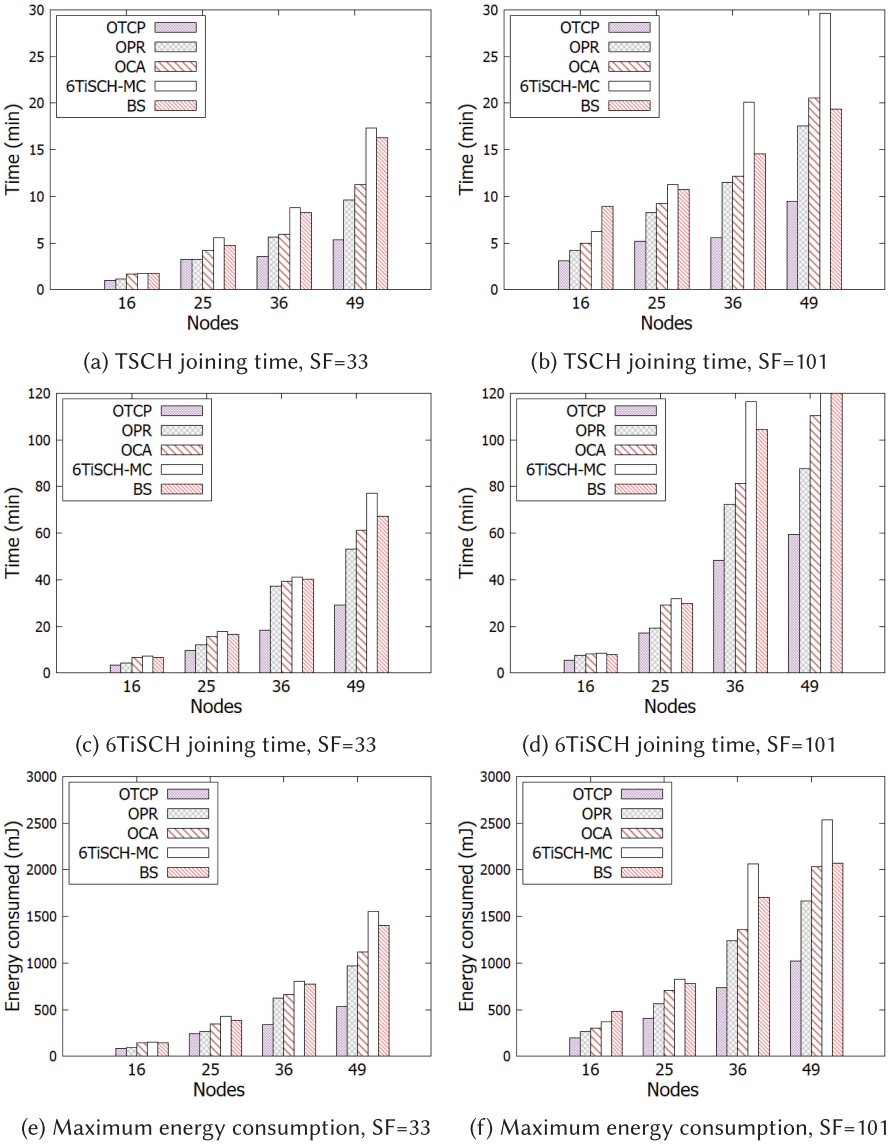
Energy consumption of a node in communication (E_{Comm}) computed as:

$$E_{Comm} = \frac{Tx \times 18.8\ mA\ +\ Rx \times 17.4\ mA}{(RTIMER\_SECOND)} \times 3Volts, \tag{26}$$

where Tx and Rx are the total number of ticks the radio has been in transmit and receive mode, respectively, during network formation. These values are obtained by the available Energest module in Contiki-4.5. The $RTIMER\_SECOND$ denotes the number of ticks per second and its value is 32,768. Here, we consider CC2420 wireless chip, which provides IEEE 802.15.4 connectivity for estimating energy consumption. CC2420 operates in low voltage of 2.1–3.6 V and it consumes 18.8 mA, 17.4 mA, and 1.8 mA during transmission (Tx), reception (Rx), and CPU active state, respectively. Similarly, energy consumption of a node during CPU active period (E_{CPU}) is computed as,

$$E_{CPU} = \frac{T_{CPU} \times 1.8\ mA}{(RTIMER\_SECOND)} \times 3Volts, \tag{27}$$

where T_{CPU} is the number of ticks the CPU has been in active mode during network formation.



(a) TSCH joining time, SF=33 (b) TSCH joining time, SF=101

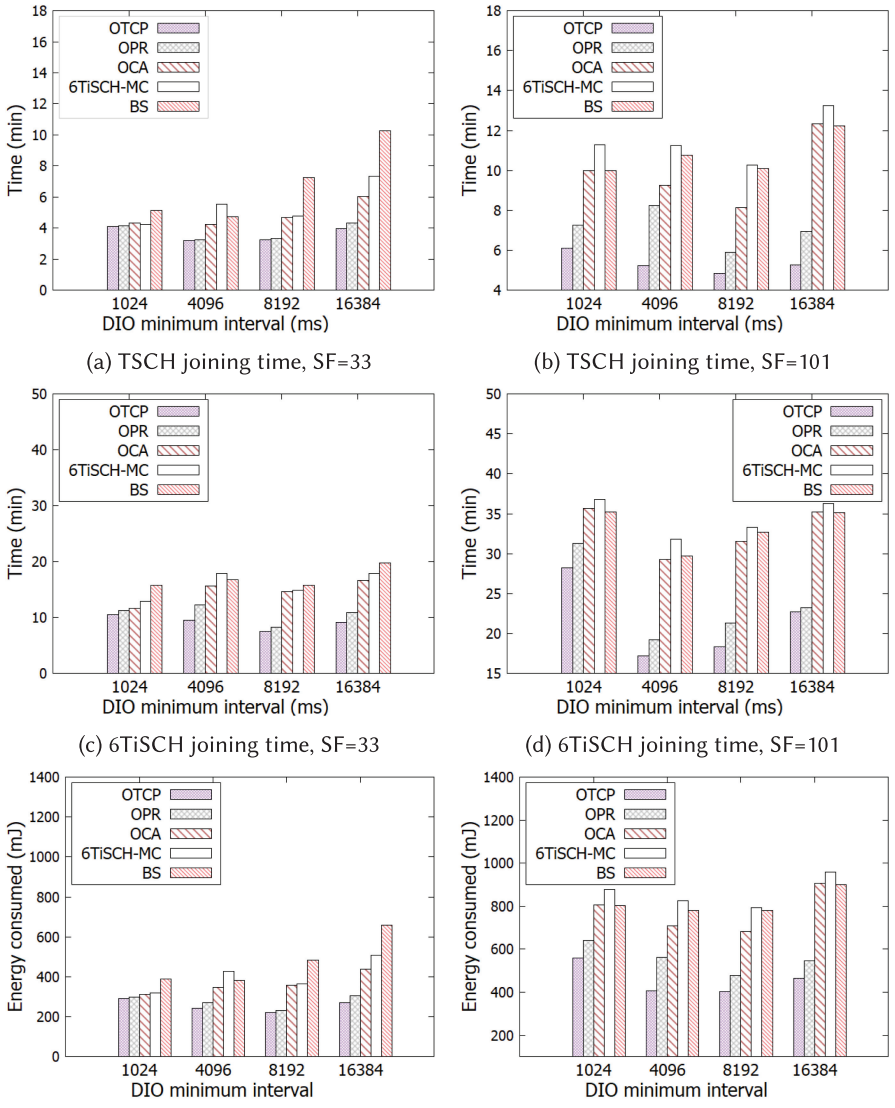
(c) 6TiSCH joining time, SF=33 (d) 6TiSCH joining time, SF=101

(e) Maximum energy consumption, SF=33 (f) Maximum energy consumption, SF=101

Fig. 11. Simulation results using different size of slotframe lengths (SF) and grid topologies (number of nodes).

5.3 Results and Discussion

Simulation results are shown in Figures 11 and 12. Figure 11 shows the results using different network sizes, and Figure 12 shows the results using different sets of DIO intervals. The evaluation is done using two different slotframe lengths (SF), such as 33 and 101 timeslots. The frequency of the occurrence of shared cells using $SF = 33$ is more than $SF = 101$ within a fixed time interval. Hence, nodes can transmit their control packets more frequently using $SF = 33$. Figures 11(a) and 11(b) show the TSCH joining time, and Figures 11(c) and 11(d) show the 6TiSCH joining time, respectively, in varied size of grid topologies. From the result, we can see that the OTCP



(a) TSCH joining time, SF=33 (b) TSCH joining time, SF=101

(c) 6TiSCH joining time, SF=33 (d) 6TiSCH joining time, SF=101

(e) Maximum energy consumption, SF=33 (f) Maximum energy consumption, SF=101

Fig. 12. Simulation results using different sets of DIO intervals and *slotframe* lengths in a (5×5) grid topology.

outperforms all the benchmark schemes. Furthermore, OPR and OCA also individually show better results than the 6TiSCH-MC and BS. This is because of priority alternation of control packets and providing sufficient routing information packet along with opportunistic channel access to the nodes that have generated DIO packet after receiving JRQ frame or DIS request. The proposed schemes allow the pledges to join the network in less time and transmit their control packets as soon as possible. This, in turn, helps the multi-hop distant pledges to get EB frame in less time. Again, using $SF = 33$, nodes perform better than using $SF = 101$. It is because of the frequent occurrence of shared cells using $SF = 33$. The joined nodes can transmit more control

packets in less time, and hence, it improves the performance of the pledges in terms of joining time. Therefore, all the benchmark schemes and proposed schemes show similar performance when the number of nodes is less (e.g., 16 and 25 nodes) using $SF = 33$. However, when the number of nodes increases, congestion in the shared cell is also increased, which in turn results in longer joining time of the pledges. Significant effect of congestion can be seen when more number of nodes present in the network (e.g., 36 and 49 nodes) and slotframe length is 101 timeslots. It is because of the severe congestion in less occurring shared cells. All the nodes try to send their control packets in less number of available shared cells, which reduces the probability of getting required control packets by the pledges. Hence, longer joining time can be observed due to high congestion in the shared cell. Note that the chances of occurring all the three problems, mentioned in Section 3, increase with the increasing congestion in the shared cell.

Similarly, the proposed scheme OTCP performs better than the benchmark schemes in terms of energy consumption also. It is because of the less amount of channel scanning time as the pledges join TSCH network more quickly. The pledges need to keep their radio active before TSCH synchronization (during this period radio duty cycle of a pledge is 100%) to receive a valid EB frame, which in turn consumes more energy. Using the proposed scheme, the multi-hop nodes receive their EB frame in less time, which saves their energy. Figures 11(e) and 11(f) show comparison of the maximum energy consumption by a node using slotframe lengths $SF = 33$ and $SF = 101$. The energy consumption by the nodes is almost same in all the schemes when less number of nodes use more shared cells (in less slotframe length) to transmit their control packets. It is because of the less congestion in the frequently occurring shared cell. The pledges spend less amount of time to join both the TSCH and 6TiSCH networks. More energy consumption can be observed when there is severe congestion in the shared cells, i.e., when the network size is large, and slotframe length is more. OTCP shows better performance than the benchmark schemes, because it allows the nodes to transmit their urgent packet even when the congestion in the network increases. Furthermore, OTCP provides sufficient routing information packets in a network during its formation along with assigning priority to the control packets based on their requirement. OTCP does not always allow the joined nodes to consider EB frame as the highest priority packet. Hence, in the proposed scheme, OTCP performs significantly better than the benchmark schemes in terms of joining time and energy consumption even in a severely congested network.

To provide sufficient routing information, the proposed scheme changes the behavior of the Trickle Algorithm during network formation. Therefore, we further evaluate the performance of the proposed schemes using different sets of minimum and maximum DIO generation intervals. The results are shown in Figure 12 using a 5×5 grid topology and different slotframe lengths such as $SF = 33$ and $SF = 101$ timeslots. The Figures 12(a) and 12(b) show the TSCH joining time and Figures 12(c) and 12(d) show the 6TiSCH joining time, respectively, using different sets of DIO intervals. In the graphs, it can be seen that the network formation time is high for very low as well as very high value of DIO interval. The frequent transmission of DIO packet congests the shared cells, which in turn results in longer joining time. And again, due to very less frequent transmission of DIO packet, the pledges need to wait for longer amount of time to receive a valid DIO packet, which again results in higher network formation time. In case of BS scheme, due to less frequent transmission of both the EB frame and DIO packet, both the TSCH and 6TiSCH joining times increase even using $SF = 33$. It is because the pledges need to wait for more amount of time to get both the control packets. However, the same BS scheme performs better than 6TiSCH-MC when $SF = 101$ is used. It is due to more congestion in the shared cells in 6TiSCH-MC. All the joined nodes try to transmit their control packets in less number of shared cells. Whereas, in BS scheme, joined nodes transmit less number of EB frames than that in 6TiSCH-MC, and thus, less congestion in the shared cells. It can also be seen that the OPR performs almost similar to OTCP in

Table 3. Results of Number of DIO Suppression and Trickle Reset Count within 60 *minutes* of Simulation Using 5×5 Grid Topology and *slotframe* length=101

Simulation Time	Scheme	I_{min} = 1,024ms			I_{min} = 4,096ms		
		k=1	k=3	k=10	k=1	k=3	k=10
First 30 min	6TiSCH-MC	<95, 916>	<14, 544>	<0, 1018>	<107, 870>	<14, 949>	<0, 907>
	BS	<67, 759>	<9, 413>	<0, 873>	<93, 771>	<8, 713>	<0, 857>
	OTCP (this work)	<45, 488>	<7, 280>	<0, 325>	<82, 549>	<6, 515>	<0, 325>
Next 30 min	6TiSCH-MC	<84, 707>	<14, 649>	<0, 760>	<63, 855>	<21, 773>	<0, 661>
	BS	<79, 471>	<11, 612>	<0, 756>	<58, 798>	<12, 551>	<0, 573>
	OTCP (this work)	<56, 277>	<9, 514>	<0, 378>	<29, 464>	<6, 352>	<0, 378>

The results are presented as follows: <number of DIO suppression, Trickle reset count>.

terms of 6TiSCH joining time and energy consumption, as their behaviors are almost same except the opportunistic channel access in OTCP. Whereas OCA performs similar to BS and 6TiSCH-MC in terms of 6TiSCH joining time and energy consumption when $SF = 33$ is used. It is because the OCA does not change the behavior of the TA. Therefore, the number of routing packets generated in the network is similar to BS and 6TiSCH-MC. The longer joining time of the nodes also affects in their energy consumption. Figures 12(e) and 12(f) show the maximum energy consumption of a node during network formation using different DIO intervals. Due to higher joining time, nodes consume more energy when the value of minimum DIO interval equals $1,024$ ms and $16,384$ ms. Here also, our proposed scheme OTCP outperforms all the existing benchmark protocols, as the pledges do not keep the radios active for more amount of time in this scheme.

It is worth mentioning that the proposed scheme transmits sufficient routing information packet by opportunistically altering the priority of control packets EB and DIO, resetting the Trickle interval and providing channel access by adjusting backoff exponent. However, the TA has a DIO suppression mechanism that suppresses the transmission of generated DIO when sufficient number of consistent DIOs are already transmitted by neighbors in the network. For this suppression mechanism, a redundancy constant k is used as a threshold on the number of overheard DIOs. If a joined node receives more than or equal number of k consistent DIOs from its neighbours in a Trickle interval, then the node suppresses its own DIO to save energy. It is pertinent to mention that the proposed scheme has added one more reason for Trickle reset. There are many other reasons for the same, such as multicast DIS request, DODAG reset time reached, reception of inconsistent DIO, changed in DODAG version, and so on, as mentioned in the RPL [25]. The number of DIO transmission increases if a node performs Trickle reset frequently. The authors in Reference [27] have already shown that the increasing number of DIO transmission leads to DIO suppression of neighbours, which in turn forces the neighbors to remain silent for longer amount of time resulting in poor performance. However, the resetting of Trickle interval is not a continuous event in the proposed scheme, as it only occurs during network formation under the given condition as mentioned in Algorithm 1. Combining with the other modifications as mentioned in Algorithms 1 and 2, the proposed scheme OTCP actually suppresses less number of DIO transmission. Table 3 shows a comparison on the number of DIO suppression and the number of Trickle reset observed in a network of 5×5 grid topology nodes. Simulation is performed for 60 min. It can be seen from Figure 12(d) that the 6TiSCH network is formed completely within 30 min and 20 min for I_{min} = 1,024 ms and 4,096 ms, respectively, for the above-mentioned simulation setup. Beyond this time period, the network performs regular operation such as data transmission. Therefore, we have tabulated the DIO suppression results for the initial 30 min and next 30 min separately in the table.

The results are shown by varying DIO minimum interval (I_{min}) and redundancy constant (k), as they also play role in DIO suppression [27]. The results show that the proposed scheme suppresses less number of DIOs as well as performs less number of Trickle reset compared to that in the benchmark schemes. Please note that if the benchmark schemes are used, the already joined nodes reset the Trickle interval more frequently due to more number of multicast DIS requests sent by other nodes. One of the reasons for sending multicast DIS request by a node is that when the node does not hear anything from its parent for a long period of time. It is already mentioned that, in the benchmark schemes, nodes need to wait for longer amount of time on average to get DIOs from their parents, so the number of multicast DIS transmission is more, which increases the Trickle reset count as well. Thus, it results in higher value of suppression count as shown in the table. However, in the proposed scheme, Trickle reset count is less due to less number of multicast DIS request. This is because of the opportunistic transmission of DIOs by which the waiting time to get a DIO packet for the nodes actually reduces. Furthermore, the number of DIO transmission is also less, as the number of Trickle reset count is less. It further results in less congestion in shared cell and also saves energy by transmitting less number of DIOs.

Finally, quick reception of DIO packet helps the pledges in finding best route or new route immediately. This, in turn, reduces the possibility of network inconsistency. Hence, the proposed scheme helps in best route selection or multiple route selection during (or after) network formation by allowing opportunistic transmission of DIO packets. Additionally, the proposed schemes help the multi-hop distance pledges to join the network in less time by quickly providing sufficient routing information to the nodes.

6 TESTBED EXPERIMENTS

6.1 Testbed Experimental Setup

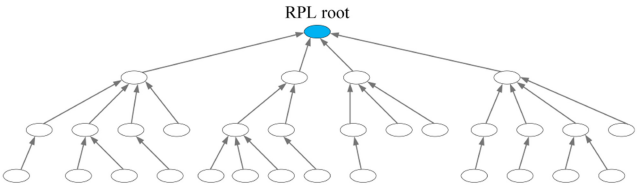
The implementation of OTCP scheme in Contiki-4.5 has been flashed in a real testbed at FIT IoT-LAB [23]. We use 31 IoT-Lab M3 nodes using two different topologies deployed in Grenoble locations for testing our proposed scheme in the real testbed. The used M3 node is an STM32 (ARM Cortex M3) micro-controller-based node that supports FreeRTOS, Contiki, and RIOT operating systems. The used topologies are shown in Figure 13. The nodes are distributed very densely in topology 1, whereas in topology 2 nodes are distributed sparsely. Again, the maximum hop length is more in topology 2 than the topology 1. The used parameters during the testbed experiments are mentioned in Table 4.

6.2 Benchmark Schemes and Performance Metric

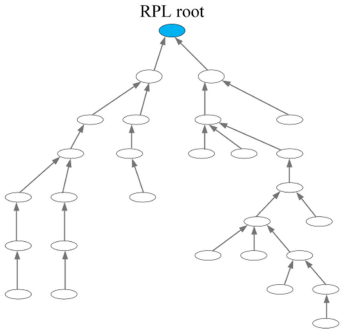
In the testbed experiments, same benchmark schemes and performance metrics are used, as mentioned in Section 5.2, to evaluate the proposed scheme. Unlike the plots of simulation results in which total joining time of all the pledges are plotted, the experimental results are plotted in different ways. In this section, how many pledges have become the TSCH and 6TiSCH joined nodes within a fixed time interval such as 0–3, 6–9, 12–15 minutes are shown in the plots. Note that a pledge is called a TSCH joined node when it receives its first valid EB frame. However, a TSCH joined node becomes a 6TiSCH joined node when it receives a valid DIO packet after its secure enrollment in the network. Similarly, the average energy consumption of the nodes are also shown in such intervals.

6.3 Testbed Results and Discussion

The received results are plotted in Figure 14 using 95% confidence interval. Figures 14(a) and 14(c) show the TSCH joining time and Figures 14(b) and 14(d) show the 6TiSCH joining time for both the



(a) Topology 1



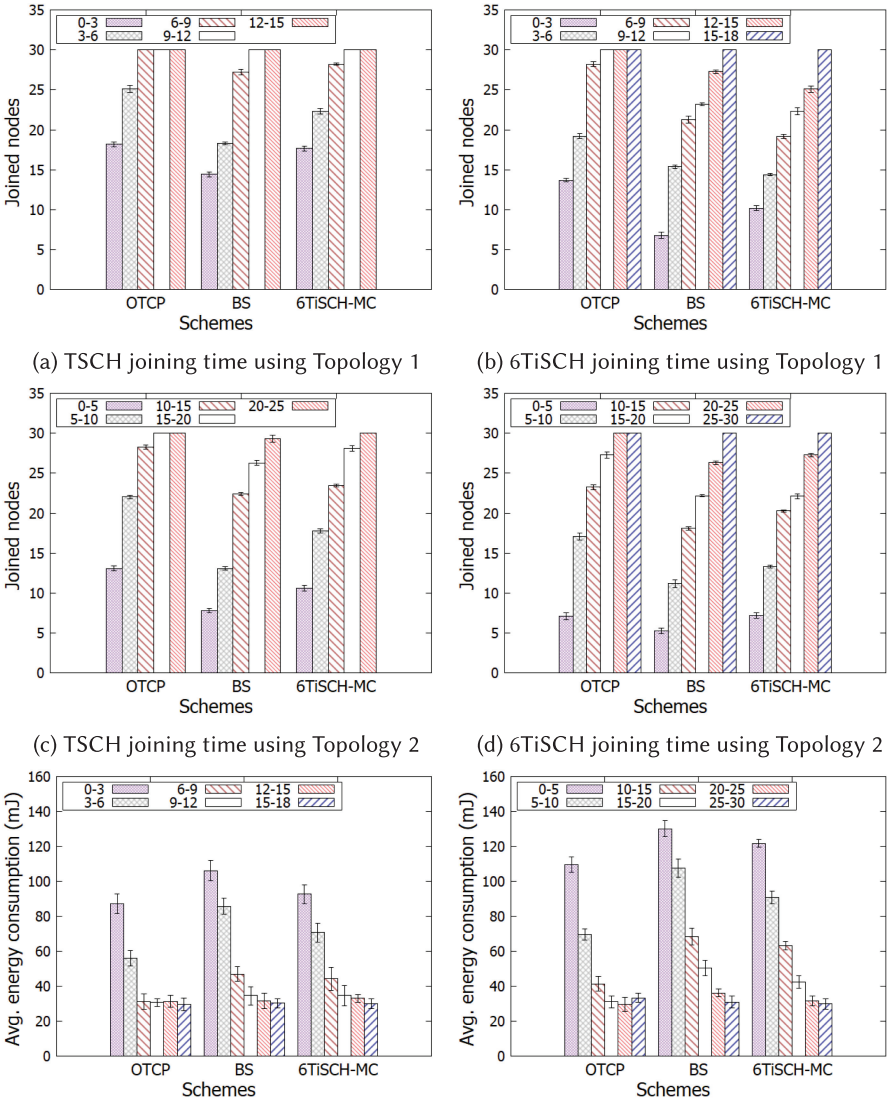
(b) Topology 2

Fig. 13. Two different topologies considered for testbed experiments.

Table 4. Expermental Parameters

Parameter	Value
Operating System	Contiki-4.5
Testbed	Grenoble, FIT IoT-LAB
Node type	IoT-Lab M3
Network size	31
Number of channels	16
Timeslot length (L)	10 ms
Slotframe size	101
EB interval	4 sec
RPL version	RPL Lite
DIO interval (min-max)	$2^{12} - 2^{20}$ ms
Keep-alive interval	30 sec
RPL DIO-interval	Trickle interval
RPL Redundancy constant	10
RPL Objective Function	MRHOF - ETX

topologies. The received results show that the proposed scheme OTCP outperforms the benchmark schemes in terms of joining time and energy consumption. The reason is due to providing sufficient routing information by alternating the priority of different control packets and resetting the TA during the requirements of routing information packet. Furthermore, the proposed scheme also allows the urgent packet containing nodes to access the transmission channel opportunistically, which in turn reduces the channel access delay during network formation. The received results also signify that when the hop distance increases from the RPL root node, the joining times are also increased. This is because a node needs to join the network completely, i.e., a node should get both



(a) TSCH joining time using Topology 1 (b) 6TiSCH joining time using Topology 1

(c) TSCH joining time using Topology 2 (d) 6TiSCH joining time using Topology 2

(e) Avg. energy consumption for Topology 1 (f) Avg. energy consumption for Topology 2

Fig. 14. Testbed results on different node joining time and average energy consumption using *slotframe* length SF=101 timeslots, DIO minimum interval=4,096 ms. Each bar in Figures 14(a), 14(b), 14(c), 14(d) shows the number of joined nodes and in Figures 14(e), 14(f) shows the average energy consumption under various time intervals (e.g., 0–3, 6–9, 12–15). The measuring units of time intervals are minutes.

EB frame and DIO packet, before sending its control packets. So, the nodes located in more depth need to wait for more amount of time than the nodes that are in less depth from the RPL root node to get a valid EB frame and DIO packet. Hence, both the TSCH and 6TiSCH joining time of the pledges are more in topology 2 in all the (existing and proposed) schemes, as it has nodes with more hop distance from the RPL root than the topology 1. Furthermore, the 6TiSCH-MC outperforms BS scheme in topology 2, as the nodes are distributed sparsely in topology 2. Hence, nodes need to wait for more amount of time to receive their EB frames, because less number of EB frames are

transmitted by the less numbers of neighbors. It results in less EB reception probability, which in turn increases the overall formation time of the network. As longer joining time contributes to more energy consumption, hence, our proposed scheme outperforms the benchmark scheme in terms of energy consumption, as shown in Figures 14(e) and 14(f).

From the received results of testbed experiments using two different routing topologies, it can be concluded that the proposed scheme performs better than benchmark schemes in terms of joining time and energy consumption.

7 CONCLUSION

In this article, we propose opportunistic transmission of control packets (OTCP), which is the combination of opportunistic priority alternation and rate control (OPR) scheme and opportunistic channel access (OCA) scheme for faster association of nodes in 6TiSCH networks. In 6TiSCH network, by default the EB frame is given the highest priority over other control packets. Further, the rate of routing information packet transmission by joined nodes is independent of pledge's joining process. Therefore, at the beginning, analytically, we show that the performance of 6TiSCH network formation degrades because of the EB's highest priority and the un-coordination between the rate of sending routing information packet and its requirement by pledges. To overcome these two problems, we propose the scheme OPR. In OPR, a joined node gives the highest priority to DIO packet over EB frame whenever it is required. OPR also allows a joined node to reset its DIO transmission interval in Trickle Algorithm when it receives a JRQ frame from a pledge to transmit a DIO packet immediately. Receiving a DIO packet in less time also helps the pledges to choose their routing parents efficiently, which further reduces inconsistency in a network. Apart from these two problems, this article deals with the channel congestion problem of CSMA/CA. Analytically, we show that an increasing number of joined nodes delay the network formation process because of increasing congestion in shared slots. The joined nodes or pledges may not be able to send their urgent control packet within a minimum time when it is required, which, in turn, increases network formation time. Therefore, we propose an OCA scheme for such urgent packet containing nodes to access the shared slot immediately. Theoretical results are provided for the validation of the proposed schemes. Again, simulation and real testbed (FIT IoT-LAB) results of the proposed schemes are compared with the benchmark schemes 6TiSCH-MC and BS. The obtained results show significant improvement with respect to network joining time of 6TiSCH networks.

REFERENCES

[1] Diego Dujovne, Thomas Watteyne, Xavier Vilajosana, and Pascal Thubert. 2014. 6TiSCH: Deterministic IP-enabled industrial internet (of things). *IEEE Commun. Mag.* 52, 12 (Dec. 2014), 36–41.

[2] Thomas Watteyne, Pere Tuset-Peiro, Xavier Vilajosana, Sofie Pollin, and Bhaskar Krishnamachari. 2017. Teaching communication technologies and standards for the industrial IoT? Use 6TiSCH! *IEEE Commun. Mag.* 55, 5 (May 2017), 132–137.

[3] Thomas Watteyne, Vlado Handziski, Xavier Vilajosana, Simon Duquennoy, Oliver Hahm, Emmanuel Baccelli, and Adam Wolisz. 2016. Industrial wireless IP-based cyber-physical systems. *Proce. IEEE* 104, 5 (May 2016), 1025–1038.

[4] Thomas Watteyne, Ankur Mehta, and Kris Pister. 2009. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. 116–123.

[5] IEEE Standards Association. 2012. IEEE standard for local and metropolitan area networks-Part 15.4: Low-rate wireless personal area networks (LR-WPANs) Amendment 1: MAC sublayer. IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011). IEEE Standards Association, 1–225.

[6] Atis Elsts, Xenofon Fafoutis, George Oikonomou, Robert Piechocki, and Ian Craddock. 2020. TSCH networks for health IoT: Design, evaluation and trials in the wild. *ACM Trans. Internet Things* 1, 2 (2020).

[7] Nicola Accettura, Elvis Vogli, Maria Rita Palattella, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. 2015. Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation. *IEEE Internet Things J.* 2, 6 (Dec. 2015), 455–470.

[8] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. 2013. Standardized protocol stack for the Internet of (Important) Things. *IEEE Commun. Surv. Tutor.* 15, 3 (Mar. 2013), 1389–1406.

[9] Xavier Vilajosana, Thomas Watteyne, Tengfei Chang, Mališa Vučinić, Simon Duquennoy, and Pascal Thubert. 2020. IETF 6TiSCH: A tutorial. *IEEE Commun. Surv. Tutor.* 22, 1 (2020), 595–615. https://ieeexplore.ieee.org/document/8823863.

[10] Xavier Vilajosana, Kristofer Pister, and Thomas Watteyne. 2017. *Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration.* RFC 8180. IETF. Retrieved from https://tools.ietf.org/html/rfc8180.

[11] Michael Richardson. 2019. *6TiSCH Zero-Touch Secure Join Protocol.* Internet Draft. Retrieved from https://tools.ietf.org/html/draft-ietf-6tisch-dtsecurity-zerotouch-join-04.

[12] Malisa Vucinic, Jonathon Simon, Kris Pister, and Michael Richardson. 2017. *Minimal Security Framework for 6TiSCH.* Internet Draft. Retrieved from https://tools.ietf.org/html/draft-ietf-6tisch-minimal-security-02.

[13] Elvis Vogli, Giuseppe Ribezzo, Luigi Alfredo Grieco, and Gennaro Boggia. 2015. Fast join and synchronization schema in the IEEE 802.15.4e MAC. In *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops.* 85–90.

[14] Domenico De Guglielmo, Alessio Seghetti, Giuseppe Anastasi, and Marco Conti. 2014. A performance analysis of the network formation process in IEEE 802.15.4e TSCH wireless sensor/actuator networks. In *Proceedings of the IEEE Symposium on Computers and Communications.* 1–6.

[15] Elbis Vogli, Giuseppe Ribezzo, Luigi Alfredo Grieco, and Gennaro Boggia. 2018. Fast network joining algorithms in industrial IEEE 802.15.4 deployments. *Ad Hoc Netw.* 69 (2018), 65–75.

[16] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. 2016. A model-based beacon scheduling algorithm for IEEE 802.15.4e TSCH networks. In *Proceedings of the IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks.* 1–9.

[17] Thang Phan Duy, Thanh Dinh, and Younghan Kim. 2016. A rapid joining scheme based on fuzzy logic for highly dynamic IEEE 802.15. 4e time-slotted channel hopping networks. *Int. J. Distrib. Sens. Netw.* 12, 8 (2016).

[18] Carlo Vallati, Simone Brienza, Giuseppe Anastasi, and Sajal K. Das. 2019. Improving network formation in 6TiSCH networks. *IEEE Trans. Mob. Comput.* 18, 1 (Jan. 2019), 98–110.

[19] Malisa Vucinic, Thomas Watteyne, and Xavier Vilajosana. 2017. Broadcasting strategies in 6TiSCH networks. *Internet Technol. Lett.* 1, 1 (2017).

[20] Alakesh Kalita and Manas Khatua. 2019. Faster joining in 6TiSCH network using dynamic beacon interval. In *Proceedings of 11th International Conference on Communication Systems Networks.* 454–457. DOI: 10.1109/COMSNETS.2019.8711460

[21] Alakesh Kalita and Manas Khatua. 2020. Opportunistic priority alternation scheme for faster formation of 6TiSCH network. In *Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN'20).* DOI: http://dx.doi.org/10.1145/3369740.3369779

[22] Alakesh Kalita and Manas Khatua. 2020. Channel condition based dynamic beacon interval for faster formation of 6TiSCH network. *IEEE Trans. Mob. Comput.* (2020). DOI: http://dx.doi.org/10.1109/TMC.2020.2980828

[23] Cedric Adjih, Emmanuel Baccelli, Eric Fleury, Gaetan Harter, Nathalie Mitton, Thomas Noel, Roger Pissard-Gibollet, Frederic Saint-Marcel, Guillaume Schreiner, Julien Vandaele, and Thomas Watteyne. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *Proceedings of the IEEE 2nd World Forum on Internet of Things (WF-IoT'15).* 459–464.

[24] Ines Khoufi, Pascale Minet, and Badr Rmili. 2017. Beacon advertising in an IEEE 802.15.4e TSCH network for space launch vehicles. In *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences.*

[25] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, J. P. Vasseur, and Roger Alexander. 2012. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.* RFC 6550. IETF. Retrieved from https://tools.ietf.org/html/rfc6550.

[26] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. 2015. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems.* 337–350.

[27] Carlo Vallati and Enzo Mingozzi. 2013. Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol. In *Proceedings of the Sustainable Internet and ICT for Sustainability (SustainIT'13).* 1–9.

[28] Malisa Vucinic, Gabriele Romaniello, Laurene Guelorget, Bernard Tourancheau, Franck Rousseau, Olivier Alphand, Andrzej Duda, and Laurent Damon. 2014. Topology construction in RPL networks over beacon-enabled 802.15.4. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'14).* 1–7.

[29] Muneer Bani Yassein, Shadi Aljawarneh, and Esra'a Masa'deh. 2017. A new elastic trickle timer algorithm for Internet of Things. *J. Netw. Comput. Applic.* 89 (2017), 38–47. DOI: http://dx.doi.org/10.1016/j.jnca.2017.01.024

[30] Baraq Ghaleb, Ahmed Y. Al-Dubai, Elias Ekonomou, Imed Romdhani, Youssef Nasser, and Azzedine Boukerche. 2018. A novel adaptive and efficient routing update scheme for low-power lossy networks in IoT. *IEEE Internet Things J.* 5, 6 (2018), 5177–5189.

[31] M. Udin Harun Al Rasyid, Isbat Uzzin Nadhori, Amang Sudarsono, and Ridho Luberski. 2014. Analysis of slotted and unslotted CSMA/CA wireless sensor network for E-healthcare system. In *Proceedings of the International Conference on Computer, Control, Informatics and Its Applications (IC3INA'14)*. 53–57.

[32] Ahmed Naseem Alvi, Safdar Hussain Bouk, Syed Hassan Ahmed, and Muhammad Azfar Yaqub. 2016. Influence of backoff period in slotted CSMA/CA of IEEE 802.15.4. In *Wired/Wireless Internet Communications*. Springer International Publishing, 40–51.

[33] Phil Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and Jeonggil Ko. 2011. *The Trickle Algorithm*. RFC 6206. IETF. Retrieved from https://tools.ietf.org/html/rfc6206.

[34] Yang Xiao.2005. Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs. *IEEE Trans. Wirel. Commun.* 4, 4 (July 2005), 1506–1515.

[35] Manas Khatua and Sudip Misra. 2016. D2D: Delay-aware distributed dynamic adaptation of contention window in wireless networks. *IEEE Trans. Mob. Comput.* 15, 2 (Feb. 2016), 322–335.

[36] Adam Dunkels, B. Gronvall, and Thiemo Voigt. 2004. Contiki—A lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th IEEE International Conference on Local Computer Networks*. 455–462.

Received January 2020; revised August 2020; accepted October 2020