Autonomous Allocation and Scheduling of Minimal Cell in 6TiSCH Network

Alakesh Kalita and Manas Khatua, *Member, IEEE*

***Abstract*—6TiSCH standardization helps the Industrial Internet of Things (IIoT) to achieve reliable and timed data delivery. It also deals with minimal resource allocation during network formation. However, faster network formation using minimum resources remains an active research issue. 6TiSCH minimal configuration standard (6TiSCH-MC) recommends to use only one cell per slotframe known as the minimal cell for all the nodes to transmit their network bootstrapping traffic. It is observed that, including 6TiSCH-MC, the existing schemes did not use all the available cells, and so, all the physical channels at the timeslot where this minimal cell resides, i.e., at timeslot zero. It results in underutilization of channel resources, and thus, higher network formation time. To leverage the available cells at timeslot zero of each slotframe, and thus to improve the network formation performance, an autonomous allocation and scheduling of minimal cell (TACTILE) is proposed. The main challenge is to utilize the available cells at timeslot zero as there is a minimal cell already scheduled for network bootstrapping. To address this issue, TACTILE distributes the location of the minimal cell as per nodes' `EUI64` addresses along the different physical channels followed by scheduling them intelligently to avoid de-synchronization among nodes. Combined with Markov chain-based theoretical analysis, evaluation of TACTILE is done on the FIT IoT-LAB real testbed. The testbed results show that TACTILE can achieve 87% and 42% improvements in terms of joining time and energy consumption, respectively, compared to 6TiSCH-MC.**

***Index Terms*—6TiSCH, autonomous cell allocation, autonomous scheduling, Industrial Internet of Things (IIoT), network formation.**

I. Introduction

THE COMMUNICATION protocol layers in the Industrial Internet of Things (IIoT) provide many services, such as network bootstrapping and interoperability. Faster bootstrapping of IIoT network helps in energy saving and efficient sensory data collection. 6TiSCH is one type of IIoT networks that leverages the traditional IP network and builds upon IEEE 802.15.4 time slotted channel hopping (TSCH) with 6LoWPAN [1]–[5]. Formation of the 6TiSCH network is the initial stage of establishing communication among the nodes. At the beginning of network formation, only the join registrar/coordinator (JRC) transmits network information using an enhanced beacon (EB) frame. A new node, i.e., pledge tries to join the network by randomly scanning the channels for receiving network advertisement control packet EB. Once the pledge receives a valid EB, it transmits the join request (JRQ) frame to the JRC or selected parent, and subsequently, it receives the join response (JRS) frame from the JRC/parent for secure enrollment [6], [7]. After securely joining the network, the pledge waits for the routing layer control packet, which is a destination oriented directed acyclic graph information object (DIO) used by the network layer routing protocol for low power and lossy network (RPL) [8]. Once the pledge successfully receives DIO, it becomes an RPL node, and thus it can transmit its own EB frame, which helps to join any new node further in the network.

Although IEEE 802.15.4e standard defines the mechanism to follow a schedule for both data and control packets, at the same time, it did not define how the schedule is built and maintained. Basically, resource allocation is needed because the applications (e.g., smart home, wearable devices, and industrial automation system) that use 6TiSCH communication do not have predefined or periodic traffic patterns, i.e., they could have nondeterministic traffic pattern or event-based traffic generation. Furthermore, nodes can be added or removed dynamically as per applications requirement [9], [10].

Therefore, 6TiSCH minimal configuration standard (6TiSCH-MC) [11] was published to provide detailed information about minimal resource usage during network bootstrapping. This standard recommends to use only a single shared cell, also known as the minimal cell in the shared slot of a slotframe, for transmitting all control packets, such as EB, JRQ, JRS, and DIO. The minimal cell of a slotframe is shown in Fig. 1. Note that the position of minimal cell for all nodes of a network remains same, i.e., at the location (`slotOffset=0`, `channelOffset=0`). To maintain synchronization, the default allocation policy forces all nodes to use minimal cells at the same location $[0, 0]$ of a slotframe. The remaining cells starting from timeslot one of a slotframe are called managed cells, and they are used for data transmission using a communication schedule assigned by a scheduling function, such as [9] and [10].

A. Motivation

Exiting works [12]–[16] revealed that the performance of 6TiSCH network formation degrades when new nodes join in the network due to the allocation of only one shared cell per slotframe for control packet transmissions. To address this problem, Vallati *et al.* [12] increased the number of

Manuscript received October 1, 2020; revised December 23, 2020; accepted February 17, 2021. Date of publication February 25, 2021; date of current version July 23, 2021. This work was supported by the Science and Engineering Research Board, Government of India. *(Corresponding author: Alakesh Kalita.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India (e-mail: alakesh.kalita1025@gmail.com; manaskhatua@iitg.ac.in).

Digital Object Identifier 10.1109/JIOT.2021.3062115

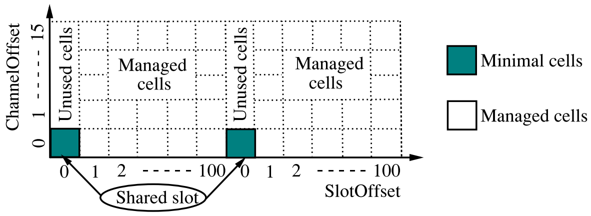


Fig. 1. Position of shared slot and minimal cell in slotframe.

shared cells per slotframe dynamically, i.e., depending on the number of control packets generated in a network. However, the additional shared cells are added in different timeslots of a slotframe, and thus, the default data transmission schedule is affected by these additional shared slots. On the other hand, Vucinic *et al.* [13], and Kalita and Khatua [14]–[16] proposed various other approaches for the faster formation of 6TiSCH network without violating the use of a single shared cell per slotframe as suggested in 6TiSCH-MC. However, these works turn out to be least efficient in terms of node's joining time as the problem of bandwidth scarcity still exists due to the use of a single shared cell per slotframe.

Furthermore, because of the minimal cell's fixed position at [0, 0], only a single channel offset is used corresponding to that shared timeslot. The usage of a single channel offset has several disadvantages while multiple channels are available.

1) The other channel offsets remain unused in `slotOffset` 0 (i.e., shared slot). These channel offsets, and so the corresponding physical channels are not used even for data packet transmission. Fig. 1 shows the unused cells in `slotOffset` 0.
2) All joined nodes participate in contention at the same physical channel for transmitting their control packets. This scenario reduces the probability of getting EB by the pledges even when more joined nodes are available because the pledges might continue scanning in different physical channels. Furthermore, it increases congestion in a single channel.

B. Contribution

To improve the performance of network formation, this work wants to leverage the available channels at timeslot zero of each slotframe up to their full extent. However, the main challenge is to utilize the available cells at timeslot zero as there is a minimal cell at [0, 0] already scheduled for network bootstrapping. To address the above issue, this article proposes autonomous allocation and scheduling of minimal cell (TACTILE). Unlike the previous approaches, the proposed approach modifies the static allocation policy of the minimal cell. Instead of allocating it at a fixed position [0, 0] for all the nodes in the network, this work proposes to autonomously allocate in any cell at timeslot zero based on the given allocation policy. This modification increases the number of shared cells used in a network at `slotOffset` 0. In such allocation, the position of a minimal cell varies from node to node. Therefore, there is a need of synchronization among them. A hierarchical scheduling technique is proposed to maintain synchronization among the nodes. The proposed scheme improves the EB reception probabilities of the pledges. Furthermore, it reduces congestion in a shared cell, and does not hamper data packet transmission as the modified allocations are done in unused cells.

The proposed scheme is implemented on Contiki-NG [17] and evaluated using FIT IoT-LAB [18] a large scale open IoT testbed. Results show that TACTILE performs better than all the benchmark schemes in terms of joining time and energy consumption.

The contributions of this article are summarized as follows.

1) An autonomous minimal cell allocation scheme (ALLOT) is proposed for faster transmission of control packets to all nodes.
2) A hierarchical odd-even minimal cell scheduling scheme (CHOICE) is proposed for maintaining synchronization between a sender and a receiver in a network.
3) Markov chain-based theoretical analysis, and experimental performance evaluation using FIT IoT-LAB real testbed are performed for the proposed scheme TACTILE which is the combination of both ALLOT and CHOICE schemes.

C. Paper Organization

The remainder of this article is organized as follows. Section II summarizes the existing works related to network formation in the 6TiSCH network. Section III briefly describes the proposed methodology. In Section IV, theoretical analysis of the proposed scheme is provided followed by the evaluation on FIT IoT-LAB testbed. Finally, conclusion is drawn in Section V.

II. Related Works

At the beginning, many works considered only TSCH synchronization, and for that, the works concentrated either on the number of transmitted EB frames [19] or collision-free EB scheduling [20]–[23]. However, a node should also get routing information after TSCH synchronization to completely join in a multihop 6TiSCH network and transmit its own beacon to extend the network further. Otherwise, network inconsistency (i.e., inconsistency in DODAG formation) may occur [11] in a multihop network topology. So, the main issues of these works are that the authors did not consider multihop 6TiSCH network formation; rather they considered only the TSCH synchronization. Also, many works did not consider synchronization between sender-receiver pairs which creates unstable networks. Furthermore, the works are not energy efficient because more EB frames are transmitted by using more shared cells. The extra allocation of shared cells further affects the data transmission schedule.

Only a few works have been published considering both the EB and DIO packets during 6TiSCH network formation [12]–[16]. These works are briefly described here. Vallati *et al.* [12] proved that the 6TiSCH-MC does not provide enough resources to transmit all the generated control packets during network bootstrap. Hence, instead of using a single shared cell per slotframe, the authors allocated more number of shared cells

TABLE I
EXISTING WORKS IN 6TISCH NETWORK FORMATION

Scheme	Control packet considered			Multi-hop network formation	Shared cell per node	Sender receiver sync.	Synchronized multiple channel usage at a time
	EB	DIO	JRQ/JRS				
[19]	✓	x	x	x	>1	x	x
[20]	✓	x	x	x	>1	x	x
[21]	✓	x	x	x	>1	x	x
[22]	✓	x	x	x	>1	x	x
[23]	✓	x	x	x	>1	x	x
[12]	✓	✓	x	✓	>1	✓	x
[13]	✓	✓	✓	✓	1	✓	x
[14], [15]	✓	✓	✓	✓	1	✓	x
[16]	✓	✓	✓	✓	1	✓	x
[24]	✓	✓	x	✓	>1	✓	x
TACTILE	✓	✓	✓	✓	1	✓	✓

per slotframe to transmit the generated control packets in less time. Vucinic *et al.* [13] proposed that the beacon transmission probability should be 0.1 irrespective of the number of nodes present in the network. Here, less EB sending probability is used to reduce the congestion in a shared cell. However, this less number of transmitted EB may increase the joining time of the nodes with less joined neighbors. Kalita and Khatua [14], [15] proposed to use dynamic beacon interval instead of fixed beacon interval to reduce congestion in a shared cell. The authors computed beacon intervals of each node depending on shared cell congestion status. However, this approach cannot reduce congestion significantly, and so the formation time. Again the authors showed the importance of DIO packet during network formation in [16], but they did not consider the channel congestion issue in that work. Even though the works [13]–[16] achieved faster formation than 6TiSCH-MC, but failed to achieve same performance as compared to DRA [12] in terms of joining time. This is because DRA allocated more number of shared cells per slotframe. However, the allocation of extra shared cells in [12] has several disadvantages. First, nodes consume more energy as they need to keep their radios active in all the allocated shared cells. Second, it hampers the data transmission schedule as some data transmitted dedicated cells are converted into shared cells. Apart from this, in total ((number of channels used in a network –1) $\times$ number of allocated shared cells) number of cells remain unused in each slotframe. Finally, the allocation is not efficient in terms of channel contention as it is not decided which node is going to transmit in which allocated shared cell. The work in [24] proposed an asynchronous distributed scheduling algorithm where a node autonomously schedules its EB, DIO, and other data packets. However, they did not evaluate their scheduling algorithm during network formation.

It is important to note that all the above mentioned works fail to take advantage of multiple channels at the shared slots, specifically at timeslot zero in each slotframe. This results in high congestion at the shared cells, and thus low EB reception probability of the pledges. This, in turn, increases network joining time and energy consumption of the pledges. It is noteworthy that as these works, including the proposed work, are related to control packet transmission; hence, any data scheduling scheme, such as [9], [10], and [25] should be used along with the proposed scheme for data transmission. The important features of existing works related to 6TiSCH network formation are shown in Table I comparing with the proposed scheme TACTILE.

III. PROPOSED SOLUTION

The works in [12]–[16] mentioned that the number of cells per slotframe allocated by 6TiSCH-MC is not sufficient to transmit all the generated control packets during 6TiSCH network bootstrapping. To validate this claim, we perform simulation experiments using the Cooja simulator on Contiki NG [17]. Two grid topologies having 5×5 and 6×6 nodes are used for the simulation. It also uses slotframe length equals 101 *timeslots* where each timeslot duration is 10 ms. Like 6TiSCH-MC, we consider only one shared cell per slotframe at `slotOffset=0` to simulate first scenario, and for second scenario, we consider two shared cells per slotframe at `slotOffset=0` and `1`. Results show that the pledges join the 6TiSCH networks, i.e., receive EB frame followed by their secure enrollment and DIO packet reception, in less time in the second scenario compared to the first scenario. The joining times are improved by 37% and 23% in 5×5 and 6×6 grid topologies, respectively. Hence, it can be claimed that the formation time of 6TiSCH networks can be improved by increasing the number of shared cells per slotframe. A similar type of improvement is also proved by the scheme DRA [12]. However, such improvement consumes multiple shared slots per slotframe which slots could be used for other communications such as data transfer otherwise.

Based on the above understanding, this work proposes, an autonomous allocation and scheduling of minimal cell (TACTILE) for allocating multiple shared cells by leveraging two attributes: 1) the location of node links in the DODAG topology and 2) the availability of multiple channels at a fixed timeslot. TACTILE contains two subschemes. At first, autonomous minimal cell allocation (ALLOT) subscheme is proposed for distributing the allocation of minimal cell vertically at `slotOffset` 0 based on the nodes' position in the DODAG topology. It autonomously allocates at most three shared cells as minimal cells at `slotOffset` 0 to a joined node for transmitting different types of control packets. As all the shared minimal cells are allocated in the same timeslot, only one cell can be used at a time by the joined node, and thus, it follows the 6TiSCH-MC standard, but, unlike DRA, the number of shared cells per slotframe is increased by allocating the shared cells in one of those unused cells in `slotOffset` 0 intelligently. This modification gives a better performance during network formation by reducing frame collision utilizing multiple channels at a time.

Now the important question is which minimal cell will be used and when by which node? Furthermore, how does the receiver know which channel the transmitter is using currently? In short, the ALLOT scheme creates a synchronization issue. For solving it, this article proposes a CHOICE subscheme. So, when a pledge wants to join the network, it should follow both the subschemes together. How the proposed subschemes are followed together is mentioned in the TACTILE framework. In the below sections,

Algorithm 1 ALLOT Autonomous Minimal Cell Allocation

```
if node is JRC then
    Rx, Tx cell = [0,hash(EUI64(JRC),N_c)]
else if node is TSCH sync but not join DODAG then
    if parent of the node is JRC then
        Rx, Tx cell = [0,hash(EUI64(JRC),N_c)]
    else
        Rx cell=[0,hash(EUI64(parent),N_c)]
        Tx cell=[0,hash(EUI64(grandparent),N_c)]
    end if
else if  node is TSCH sync and join DODAG then
    Tx cell = [0,hash(EUI64(own),N_c)]
else                      ▷ node is not TSCH sync or disconnected
    Scan channel for EB frame
end if
```

both the subschemes and the framework are discussed in detail.

A. Autonomous Minimal Cell Allocation

Algorithm 1 describes the procedure for autonomous minimal cell allocation (ALLOT) by the nodes of a 6TiSCH network. At first, only the JRC is there in the network, and thus, it assigns one minimal cell at `[0,hash(EUI64(JRC),`N_c`)]`. The function `hash(EUI64(XY),`N_c`)` returns a value between $[0, N_c - 1]$ based on the `EUI64` address of the node XY. Here, N_c is the total number of channels available in the network. Detailed procedure of the hash function is described in [26]. If a pledge receives an EB frame from the JRC, then the pledge allocates a minimal cell at `[0,hash(EUI64(JRC),`N_c`)]` for both transmission (Tx) and reception (Rx). Otherwise, it allocates two minimal cells at `[0,hash(EUI64(parent),`N_c`)]` and `[0,hash(EUI64(grandparent),`N_c`)]` considering that it has received the EB frame from another joined node who is selected as its parent node. Here, `EUI64(parent)` and `EUI64(grandparent)` are the `EUI64` addresses of parent and grandparent of the pledge, respectively. The minimal cell at `[0,hash(EUI64(parent),`N_c`)]` is used for listening any control frame from both its parent and child nodes. Whereas, the minimal cell at `[0,hash(EUI64(grandparent),`N_c`)]` is used for transmitting control frames, such as JRQ and DIS to its parent. After this allocation, when a TSCH synchronized node gets a valid DIO packet, i.e., the node joins to DODAG of the 6TiSCH network, then the node allocates one more minimal cell at `[0,hash(EUI64(own),`N_c`)]` to transmit its own control packets. Here, `EUI64(own)` is the `EUI64` address of the node itself. It is noteworthy that all the allocations are done at `slotOffset=0`, and so, in the unused cells.

In this process, a hash collision may occur when the number of joined nodes surplus the number of available channels. The hash collision results in the same minimal cell allocation to the colliding nodes. Note that the congestion in the shared minimal cell occurs only when the hash colliding nodes generate their control packets simultaneously. The channel offsets are calculated based on nodes `EUI64` addresses, making the allocation distinct. Information about parent `EUI64` address is implicitly mentioned in EB frame, whereas, grandparent `EUI64` address can be added in the information elements (IE) of an EB frame. Hence, no other control packet needs to be transmitted to provide information about added minimal cells' location. Thus, ALLOT allocates extra shared cells without affecting the data transmission schedule and increasing the radio duty cycle of the nodes.

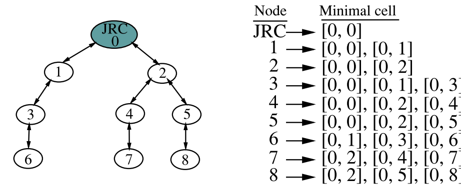


Fig. 2. Autonomous minimal cell allocation by ALLOT.

Algorithm 2 Hierarchical Odd-Even Minimal Cell Scheduling

```
if node is JRC and odd-even selection is not done then
    Randomly decide odd-even shared cell
    for listen/transmit               ▷ Do it once at the beginning
else if node's parent transmits in this shared slot then
    Schedule Rx cell[0,hash(EUI64(parent),N_c)]
else if node has packet to send then                ▷ Transmit
    if packet is broadcast or unicast to child then
        Schedule Tx cell[0,hash(EUI64(own),N_c)]
    else if node's parent is JRC then
        Schedule Tx cell[0,hash(EUI64(JRC),N_c)]
    else
        Schedule Tx cell[0,hash(EUI64
        (grandparent),N_c)]
    end if
else             ▷ Listen as node does not have packet to send
    Schedule Rx cell[0,random no % N_c)]
end if
```

Fig. 2 depicts an example of cell allocations by ALLOT. In the figure, node 6 allocates three shared minimal cells in `slotOffset` 0 which are [0, 1], [0, 3], and [0, 6]. Channel offsets 1, 3, and 6 are calculated using the `EUI64` addresses of the grandparent, parent, and the node itself, receptively. On the other hand, node 1 allocates two shared cells using channel offsets 1 and 0 as its parent and grandparent are the same.

B. Hierarchical Odd-Even Minimal Cell Scheduling

A synchronization problem will occur as the ALLOT scheme allocates minimal cells at different locations for the different nodes. So, when a parent node transmits its control packet, its immediate child node may be busy transmitting its own control packet using a different channel offset. In this case, control packet reception might be missed by the child node. This event can lead to inconsistency in a network. Therefore, proper scheduling of the allocated cells is necessary for maintaining synchronization among the pairs of parent-child nodes in a network. Algorithm 2 describes the procedure of the proposed scheduling scheme.

The absolute slot number (ASN), which denotes the total number of timeslots that has elapsed since the network started, is used in the proposed scheduling scheme. Note that if the ASN value of the minimal cell in ith slotframe is odd, then that in $(i + 1)$th slotframe is even, and *vice* *versa* when a slotframe length is an odd number. At the beginning, the JRC randomly decides whether it is going to transmit its control packets in the minimal cell associated with either odd or even ASN value. The other nodes decide their Rx and Tx cells based on their parent nodes' Tx and Rx strategies. A joined node listens from both its parent and child nodes during its parent node's transmission slot, i.e., at `[0,hash(EUI64(parent),`N_c`)]`. On the other hand, a joined node transmits its own control packet or unicast response/request to its child node during its parent node's listening slot, i.e., at `[0,hash(EUI64(own),`N_c`)]`. At the same listening slot of its parent, unicast request/response to its parent is also transmitted but using the shared cell at `[0,hash(EUI64(JRC),`N_c`)]` or `[0,hash(EUI64(grandParent),`N_c`)]` according to node's position in the DODAG. A node uses grandparent's `EUI64` address to transmit control packets to its parent because at that time the parent node listens using the grandparent's (i.e., parent of the parent) `EUI64` address. Note that when a node does not have a grandparent, its parent node is treated as its grandparent node for allocating and scheduling Tx and Rx minimal cells. In this way, hierarchical scheduling is performed to transmit control packets in a network from top to bottom of a DODAG to maintain synchronization.

This scheduling technique reduces congestion in a network as a node allows to transmit only when its parent is listening and vice versa. Referring to Fig. 2, node 6 transmits its request control packets (such as JRQ and DIS) to its parent node 3 using the `EUI64` address of node 1. This is because, at that time, node 3 listens for control packets using its parent `EUI64` address. Furthermore, when node 6 and node 1 transmits at the even ASN timeslot, node 3 used to listen in that timeslot. Hence, there is no scheduling conflict between a pair of parent-child nodes.

C. TACTILE Framework

It is already mentioned that both the ALLOT and CHOICE schemes are running together in TACTILE. Fig. 3 shows a complete framework of a new node's joining process for the construction of the 6TiSCH network. A node goes through four states: 1) pledge; 2) TSCH synced node; 3) TSCH joined node; and 4) 6TiSCH joined node/RPL node for joining to a 6TiSCH network. Initially, a pledge waits for a valid EB on a randomly selected channel. Once it receives an EB, the node moves to TSCH synced state. At this state, the node is TSCH synchronized but not yet completed the secured enrollment steps by exchanging JRQ and JRS frames. Once it is done, the node moves to TSCH joined node state. At this state, the node waits for RPL routing information required for multihop network formation. Once the node gets the RPL routing information through the DIO control packet, it moves to the last state, which is the RPL node or 6TiSCH joined node. At this state, the joining procedure to a 6TiSCH network is completed for that node. During this bootstrapping steps, a node allocates minimal cells for transmitting or listening control packets. The framework in Fig. 3 shows when the node will allocate minimal cell using Algorithm 1 and when the cell will be scheduled using Algorithm 2. Note that if a node gets disconnected from the network at any point of time, the node needs to rejoin the network as a fresh pledge node. After rejoining, the previous allocation and scheduling status get changed based on the newly selected parent. Note that steps 1 and 2 in both the algorithms are not mentioned in the framework, as they are associated with the JRC but the framework is for a new node.

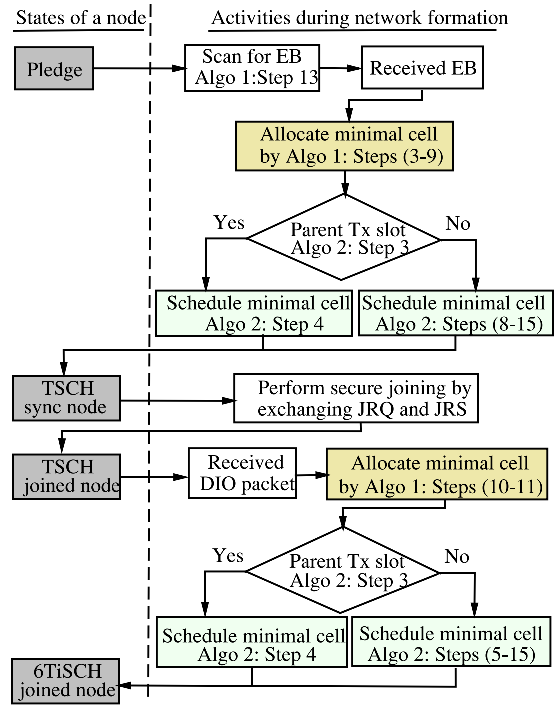


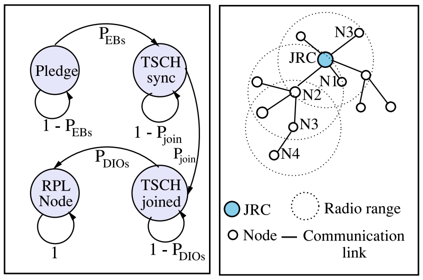
Fig. 3. TACTILE framework describing various states and activities of a pledge during its joining process.

TABLE II
LIST OF FREQUENTLY USED SYMBOLS

Symbol	Meaning
N_c	Total number of channels
L	Slotframe length
I_{eb}	Beacon generation interval
P_{eb}	EB transmission probability in a shared cell
P_{EB_S}	Successful EB transmission probability in a shared cell
P_{dio}	DIO transmission probability in a shared cell
P_{DIO_S}	Successful DIO transmission probability in a shared cell
P_{jrq}	JRQ transmission probability in a shared cell
P_{JRQ_S}	Successful JRQ transmission probability in a shared cell
P_{jrs}	JRS transmission probability in a shared cell
P_{JRS_S}	Successful JRS transmission probability in a shared cell
P_{loss}	Packet loss probability
N_D	Maximum number of trickle states

IV. Performance Analysis

In this section, the TACTILE scheme is evaluated using Markov Chain-based theoretical model followed by real testbed experiments on FIT IoT-LAB testbed. Table II summarizes few symbols and their corresponding meaning frequently used in this section.



(a) (b)

Fig. 4. (a) Markov chain model of node joining process. (b) Multihop network topology considered for the analysis.

A. Theoretical Modeling

In the framework of TACTILE, it is already mentioned that a pledge goes through four states for joining to a 6TiSCH network. As the probability of each set of activities, as mentioned in the framework, depends only on the state attained by the previous set of activities, we can model the behavior of a pledge using the Markov Chain. Those four states of a pledge would be the states of the Markov Chain model, as shown in Fig. 4(a). The average time spent in each distinct state is critical to estimate the pledge's average joining time (AJT). So, the Markov Chain model-based analysis is used to estimate the AJT of a pledge following the computation methodology as described in [15]. This is nothing but the average time required to reach to the final absorbing state of the Markov Chain starting from the initial state. The initial state shows that a pledge waits for an EB to know about the basic parameters of the network. Once the pledge receives a valid EB frame, it moves to the second state. Let us assume P_{EB_S} is the probability of successfully receiving EB frame at first state. Similarly, a pledge moves to the third state after finishing its secure enrollment process. Let us assume P_{join} is the probability of completing the secured enrollment process. In the third state, a joined node waits for routing protocol information, and once it receives at least one fresh/recent DIO packet with probability P_{DIO_S}, it moves to the final absorbing state. In this state, a node successfully joins a 6TiSCH network and becomes an RPL node or 6TiSCH joined node. So, the transition probability matrix of the Markov Chain model can be written as follows:

$$S = \begin{bmatrix} 1-P_{\mathrm{EB}_S} & P_{\mathrm{EB}_S} & 0 & 0 \\ 0 & 1-P_{\mathrm{join}} & P_{\mathrm{join}} & 0 \\ 0 & 0 & 1-P_{\mathrm{DIO}_S} & P_{\mathrm{DIO}_S} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The average number of slotframes (ASF) required to reach the absorbing state can be computed as follows:

$$\mathrm{ASF} = \frac{1}{P_{\mathrm{EB}_S}} + \left(\frac{1}{P_{\mathrm{JRQ}_S}} + \frac{1}{P_{\mathrm{JRS}_S}}\right) + \frac{1}{P_{\mathrm{DIO}_S}}. \quad (1)$$

Note that a TSCH joined node transmits the JRQ frame to its selected parent in the secured joining process, followed by receiving the JRS frame from the parent. Therefore, the probability P_{join} is written in terms of P_{JRQ_S} and P_{JRS_S}. Now, the values of P_{EB_S}, P_{DIO_S}, P_{JRS_S}, and P_{JRQ_S} are required to compute the value of ASF. For this, a multihop network model is considered as shown in Fig. 4(b), in which each node has different number of neighbors. According to the TACTILE, every node sends its control packet using its own Tx cell `[0,hash(EUI64(own),`N_c`)]`. Furthermore, if a node sends its control packet in an even shared slot, it listens in an odd shared slot, and *vice versa*. Let us assume a node has n joined neighbor nodes and M total neighbor nodes in its surrounding. In the best case, there is at least one joined neighbor, and all the remaining $(M-1)$ neighbors are a pledge. In this case, only one node transmits its control packet, so no collision occurs. All the n joined nodes transmit their control packets at the same shared slot in the worst case. Hence, there is no transmission by these n joined nodes in the immediate next shared slot. So, $(n/2)$ nodes transmit their control packets in each shared slot on average. Following this, the probability of receiving an EB frame successfully by a pledge in a shared minimal cell is computed as follows:

$$P_{\mathrm{EB}_S} = \sum_{n}^{M-n} \left(\frac{\frac{n}{2}}{N_c}\right) \frac{n}{2} P_{\mathrm{eb}} (1-P_{\mathrm{eb}})^{\frac{n-2}{2N_c}} (1-P_{\mathrm{dio}})^{\frac{n-2}{2N_c}} \times \left(1-P_{jrs}\right)^{\frac{n-2}{2N_c}} \left(1-P_{jrq}\right)^{\frac{M-n}{N_c}} (1-P_{\mathrm{loss}}) P(N=n) \quad (2)$$

where, the value of n and M can be different for every node of a network. Here, $(M-n)$ pledges join the network one by one, and the $P(N=n)$ denotes the probability of being total n joined neighbor node at any instant of time which equals to $(1/M)$ considering uniform probability distribution. When an RPL node sends its EB frame, on an average the remaining $([n/2]-1)$ RPL nodes should not send their control packets (EB/DIO/JRS). The probability of this condition equals to $(P_{\mathrm{eb}}(1-P_{\mathrm{eb}})^{(n-2/2N_c)}(1-P_{\mathrm{dio}})^{(n-2/2N_c)}(1-P_{jrs})^{([n-2]/2N_c)})$ in (2). Additionally, the neighbor new nodes also should not send their JRQs, which is represented by the probability $((1-P_{jrq})^{[M-n/N_c]})$. The term $[n-2/2N_c]$ denotes that, in TACTILE, collision occurs in worst case only when $(n/2)-1 > N_c$ in a single hop scenario. Otherwise, all the joined nodes send their control packets using different channel offset. $(1-P_{\mathrm{loss}})$ is the probability that a transmitted frame should not be lost in the channel. Finally, successful transmission probability of EB is reduced by $[n/2/N_c]$ as a pledge is not synchronized initially. Furthermore, the EB frame can be transmitted by any of the $(n/2)$ neighbor joined nodes which result in multiplying the computed probability by $(n/2)$.

Similarly, the probability of sending a DIO packet successfully in a minimal cell is computed as

$$P_{\mathrm{DIO}_S} = \sum_{n}^{M-n} \frac{n}{2} P_{\mathrm{dio}} (1-P_{\mathrm{dio}})^{\frac{n-2}{2N_c}} (1-P_{\mathrm{eb}})^{\frac{n-2}{2N_c}} \times \left(1-P_{jrs}\right)^{\frac{n-2}{2N_c}} \left(1-P_{jrq}\right)^{\frac{M-n}{N_c}} (1-P_{\mathrm{loss}}) P(N=n). \quad (3)$$

Note that in the above equations, $((n/2)/N_c)$ is not used because a node knows its coordinator's channel hopping sequence after getting an EB frame successfully. Likewise, the probability of sending a JRQ frame and receive a JRS frame

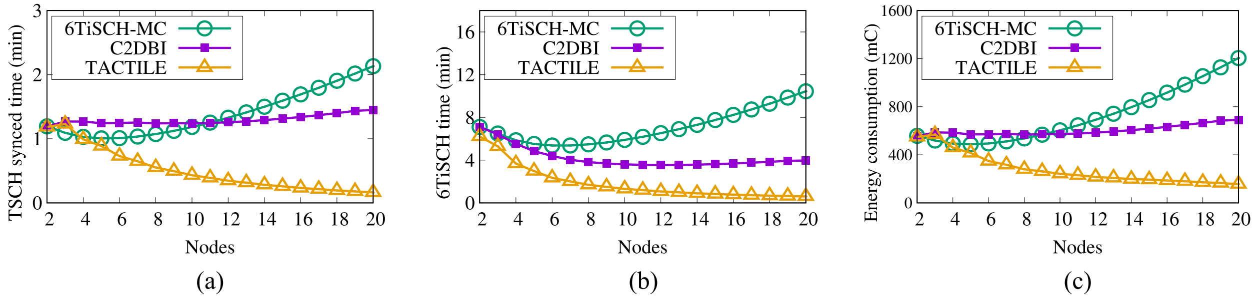


Fig. 5. Analytical results on (a) TSCH synchronization (b) 6TiSCH formation time and (c) energy consumption by a node.

successfully in a minimal cell is also computed as

$$P_{\mathrm{JRQ}_S} = \sum_{n}^{M-n} (M-n)(1-P_{\mathrm{eb}})^{\frac{\frac{n}{2}}{N_c}}(1-P_{\mathrm{dio}})^{\frac{\frac{n}{2}}{N_c}} P_{jrq} \times \left(1-P_{jrq}\right)^{\frac{M-n+1}{N_c}}\left(1-P_{jrs}\right)^{\frac{n}{2N_c}}(1-P_{\mathrm{loss}})P(N=n) \tag{4}$$

$$P_{\mathrm{JRS}_S} = \sum_{n}^{M-n} \frac{n}{2} P_{jrs}\left(1-P_{jrs}\right)^{\frac{n-2}{2N_c}}(1-P_{\mathrm{eb}})^{\frac{n-2}{2N_c}}(1-P_{\mathrm{dio}})^{\frac{n-2}{N_c}} \times \left(1-P_{jrq}\right)^{\frac{M-n+1}{N_c}}(1-P_{\mathrm{loss}})P(N=n). \tag{5}$$

Now, to compute the above mentioned probabilities, we should know the transmission probabilities of EB and DIO in a minimal cell which are P_{eb} and P_{dio}, respectively. The probability P_{eb} can be calculated as

$$P_{\mathrm{eb}} = L/I_{\mathrm{eb}} \tag{6}$$

where, L is the slotframe length and I_{eb} is the beacon interval. We compute the probability P_{dio} following the procedure described in [12] as follows:

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

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, the AJT of a new node is calculated as follows:

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

where PJT is the AJT of the parent node of a new node in a multihop network. The formation time of an entire network is the joining time of the node added last in the network.

B. Performance Metric and System Setup

Three performance metrics are mainly considered in evaluation: 1) TSCH synchronization time; 2) 6TiSCH joining time; and 3) energy consumption. Here, the TSCH synchronization time is the time required by a pledge to receive the first EB frame successfully as shown in Fig. 4(a). On the other hand, 6TiSCH joining time denotes the time required by the pledge to reach the final absorbing state of the Markov Chain model in Fig. 4(a). In theoretical evaluation, the energy consumption is calculated for GINA motes using the similar approach described in [15]. Here, the average energy consumption of a node considers both the energy consumptions when the node was pledge and after the pledge becomes a 6TiSCH joined node. As the pledges join in the network one after another, the energy consumption of a 6TiSCH joined node until full network formation is considered in the calculation. However, in testbed experiments, a node's energy consumption is calculated from its average radio duty cycle.

We consider a random network topology with the values of few parameters as follows for theoretical evaluation of the proposed scheme: $N_C = 16$, $I_{\mathrm{eb}} = 4$ s, $P_{\mathrm{loss}} = 0.2$, $I_{\min} = 8$ ms, $L = 101$ *timeslots*, $T_i = 10$ ms, $N_D = 16$, and $P_r = 0.2$. For evaluating C2DBI, we consider minimum beacon interval $I_{\mathrm{eb}}^{\min} = 4 \times L$, and maximum beacon interval $I_{\mathrm{eb}}^{\max} = 10 \times L$.

To perform the testbed experiments, at the beginning, the proposed TACTILE scheme is implemented in Contiki-NG. As the grandparent `EUI64` address is also used in the scheme, therefore, a node include its parent `EUI64` address in the IE of its EB frame. So, the child nodes can use that `EUI64` address as their grandparent `EUI64` address. Subsequently, the implementation of TACTILE has been flashed in a real testbed FIT IoT-LAB [18] for performance evaluation. The fixed beacon interval I_{eb} is set to 4 s for 6TiSCH-MC, DRA, and TACTILE, whereas, for C2DBI, $I_{\mathrm{eb}}^{\min}$ is set to 4 s and $I_{\mathrm{eb}}^{\max}$ is set to 12 s. The channel busy ratio (CBR) is calculated in the interval of 8 s. Note that in the testbed experiments, the TACTILE is also compared with one more benchmark scheme called DRA along with the 6TiSCH-MC and C2DBI. Here, a maximum 8 *shared slots* are allocated to DRA in a slotframe. In the testbed, the IoT-Lab M3 nodes are deployed using a (2×12) linear topology and a (5×5) grid topology in Grenoble and Lille locations, respectively. Other configuration settings used in the real testbed experiments are tabulated in Table III.

C. Theoretical Results

Fig. 5 pictorially depicts the theoretical results along with a comparative study with two benchmark schemes 6TiSCH-MC [11] and C2DBI [15]. Note that the theoretical analysis of 6TiSCH-MC and C2DBI are given in [15]. In this article, we have used the same analysis to draw the plots. Fig. 5(a) and (b) show the TSCH synchronization time and 6TiSCH joining time, respectively.

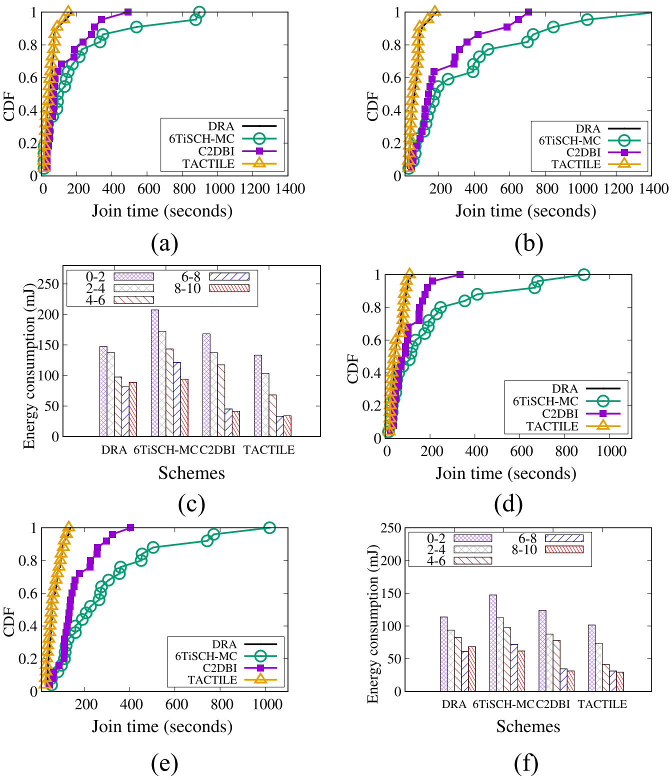


Fig. 6. Testbed results on TSCH and 6TiSCH formation time and the average energy consumption corresponding to various experimental time intervals (e.g., 0–2 min, 2–4 min, etc.,) under the (2×12) and (5×5) network topologies. (a) TSCH formation in 2×12. (b) 6TiSCH formation in 2×12. (c) Energy consumption in 2×12. (d) TSCH formation in 5×5. (e) 6TiSCH formation in 5×5. (f) Energy consumption in 5×5.

TABLE III
EXPERIMENTAL SETTINGS

Parameter	Value
Operating System	Contiki NG
Testbed	FIT IoT-LAB, Grenoble, Lille
Mote	IoT-LAB M3
Number of channels	16
Timeslot length	10 ms
Slotframe length	101 timeslots
RPL version	RPL Lite
RPL DIO interval	Trickle Algo.
Experiment duration	60 minutes

Results show that the pledges' joining time increases with the increasing number of joined nodes using the benchmark schemes, i.e., 6TiSCH-MC and C2DBI. It is because both the schemes use only a single physical channel in the shared slots to transmit control packets even though IEEE 802.14.4e allows to use multiple physical channels at a time. So, the congestion in the single transmission channel increases with the increasing number of nodes. This behavior is common in random channel access protocols such as CSMA/CA-based wireless communication. When the number of nodes increases, congestion in the transmission channel also increases. In 6TiSCH communication, this behavior results in delayed packet transmission due to congestion in shared cells. This delayed transmission of control packets increases the formation time of the 6TiSCH network. Although the congestion in a shared cell is reduced a bit in C2DBI by increasing the beacon interval, that much of reduction is not sufficient compared to that in TACTILE.

On the other hand, TACTILE reduces the congestion using the distribution of minimal cell position of different nodes along the channel offset axis per slotframe. Therefore, instead of using a single channel for all nodes, TACTILE uses all the available physical channels in the shared timeslots when the number of nodes and number of hops in the network topology is more. Hence, even when the number of nodes increases, congestion in a particular transmission channel does not increase abruptly. So, less delay in transmission of control packets, unlike the 6TiSCH-MC and C2DBI. This helps in improving the formation time. Furthermore, as the pledges scan random channels for EB, in TACTILE, the transmission of more EBs by multiple joined nodes simultaneously using different physical channels increases the EB reception probability of the pledges.

Fig. 5(c) shows the energy consumption of a node during network formation. It is observed that when the number of nodes increases, the congestion in the minimal cell also increases. Because of this increasing congestion, pledges need to keep their radios active for more amount of time to get an EB frame. In other words, nodes are having 100% radio duty cycle for a longer duration, and it causes a quick energy drain. Hence, 6TiSCH-MC shows worst result than C2DBI and TACTILE. On the other hand, leveraging all the physical channels for transmitting EB frames by different nodes in different levels of the topology, TACTILE reduces the TSCH synchronization time of the pledges, which in turn reduces their energy consumption.

D. Testbed Results

The received results are plotted in Fig. 6 using the cumulative distribution function (CDF) of TSCH and 6TiSCH formation time. Fig. 6(a) and (d) show the time required by a pledge to receive its first EB frame corresponding to two different topologies. On the other hand, Fig. 6(b) and (e) shows the time required by a pledge to receives its first DIO packet after the completion of its TSCH joining process. Received results show that the TACTILE performs better than 6TiSCH-MC and C2DBI, whereas it shows almost similar results with DRA in terms of both TSCH and 6TiSCH joining times. The reason of comparative performance by the DRA is the usage of more number of shared slots in a slotframe. So, DRA and TACTILE mitigate congestion for control packet transmissions in a slotframe.

Fig. 6(c) and (f) show the average energy consumption of a pledge during the initial 10 min of network formation. In the initial 10 min most of the nodes are synchronized with TSCH, and so, energy consumption of the network almost remains constant after this period. In DRA, all the nodes need to keep their radios active in all the allocated shared slots as the node does not know in which shared slot its expected control packet will be transmitted. This increases the radio duty cycle of the nodes, which in turn consumes more energy consumption. On the other hand, in TACTILE, a receiver knows the exact location of a sender's transmission cell. Hence, the nodes save their energy by not keeping their radios active in more than one cell in a slotframe. It can be seen that DRA consumes more energy, even more than C2DBI.

In brief, the obtained results show that the proposed TACTILE scheme achieve 87%, 67%, and 3% improvements in terms of 6TiSCH joining time, and 42%, 23%, and 29% improvements in terms of average energy consumption over 6TiSCH-MC, C2DBI, and DRA, respectively, in 5×5 grid topology. Similarly, results show significant improvements in joining time and energy consumption in 2×12 grid topology.

V. Conclusion

It is observed that the existing schemes do not use all the available physical channels at timeslot zero of each slotframe for the transmission of bootstrapping control packets. It results in lesser EB reception probability and higher shared channel congestion for the pledges. This, in turn decreases the performance of the network formation process with respect to joining time and energy consumption. To deal with the underutilization of channels, more number of shared cells are allocated by repositioning the minimal cell followed by a scheduling algorithm in the proposed scheme TACTILE. Following the 6TiSCH-MC standard, minimal cell allocation is done at timeslot zero of each slotframe, but the channel offset of the minimal cell is distributed among all the available channels. At most three minimal cells are allocated to each node and the position of the cells are different in a slotframe. So, proper scheduling is necessary to synchronize the transmission and reception among each pair of nodes. To achieve this, a scheduling scheme is also proposed in TACTILE. Along with the Markov Chain-based theoretical analysis, real testbed experiments are also performed to evaluate the proposed scheme and compare it with the existing benchmark schemes. The received results show that TACTILE can achieve 87% and 42% improvements in terms of joining time and average energy consumption, respectively, compared to 6TiSCH-MC.

References

[1] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: Deterministic IP-enabled Industrial Internet (of Things)," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[2] T. Watteyne, P. Tuset-Peiro, X. Vilajosana, S. Pollin, and B. Krishnamachari, "Teaching communication technologies and standards for the Industrial IoT? Use 6TiSCH!" *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 132–137, May 2017.

[3] T. Watteyn *et al.*, "Industrial wireless IP-based cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1025–1038, May 2016.

[4] X. Vilajosana, T. Watteyne, T. Chang, M. Vucinic, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 595–615, 1st Quart., 2020.

[5] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), Apr. 2016.

[6] M. Richardson, "6TiSCH zero-touch secure join protocol," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-6tisch-dtsecurity-zerotouch-join-04 [work-in-progress], Jul. 2019.

[7] M. Vucinic, J. Simon, K. Pister, and M. Richardson, "Constrained join protocol (CoJP) for 6TiSCH," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-6tisch-minimal-security-15 [work-in-progress], Dec. 2019.

[8] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 6550, Mar. 2012.

[9] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 455–470, Dec. 2015.

[10] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH operation sublayer (6top) protocol (6P)," Internet Eng. Task Force, Fremont, CA, USA, RFC 8480, Nov. 2018.

[11] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) configuration," Internet Eng. Task Force, Fremont, CA, USA, RFC 8180, May 2017.

[12] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving network formation in 6TiSCH networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 98–110, Jan. 2019.

[13] M. Vucinic, T. Watteyne, and X. Vilajosana, "Broadcasting strategies in 6TiSCH networks," *Internet Technol. Lett.*, vol. 1, no. 1, p. e15, 2018. [Online]. Available: https://doi.org/10.1002/itl2.15

[14] A. Kalita and M. Khatua, "Faster joining in 6TiSCH network using dynamic beacon interval," in *Proc. 11th Int. Conf. Commun. Syst. Netw.*, Jan. 2019, pp. 454–457.

[15] A. Kalita and M. Khatua, "Channel condition based dynamic beacon interval for faster formation of 6TiSCH network," *IEEE Trans. Mobile Comput.*, early access, Mar. 16, 2020. [Online]. Available: https://doi.org/10.1109/TMC.2020.2980828

[16] A. Kalita and M. Khatua, "Opportunistic priority alternation scheme for faster formation of 6TiSCH network," in *Proc. 21st Int. Conf. Distrib. Comput. Netw*, 2020, pp. 1–5.

[17] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, 2004, pp. 455–462.

[18] C. Adjih *et al.*, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *Proc. IEEE 2nd World Forum Internet Things*, Dec. 2015, pp. 459–464.

[19] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia "Fast network joining algorithms in industrial IEEE 802.15.4 deployments," *Ad Hoc Netw.*, vol. 69, pp. 65–75, Feb. 2018.

[20] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast join and synchronization scheme in the IEEE 802.15.4e MAC," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops*, Mar. 2015, pp. 85–90.

[21] 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 *Proc. IEEE Symp. Comput. Commun.*, Jun. 2014, pp. 1–6.

[22] D. Dec Gugleilmo, S. Brienza, and G. Anastas, "A model-based beacon scheduling algorithm for IEEE 802.15.4e TSCH networks," in *Proc. IEEE 17th Int. Symp. World Wireless Mobile Multimedia Netw.*, Jun. 2016, pp. 1–9.

[23] I. Khoufi, P. Minet, and B. Rmili, "Beacon advertising in an IEEE 802.15.4e TSCH network for space launch vehicles," in *Proc. 7th Eur. Conf. Aeronaut. and Aerosp. Sci.*, Jul. 2017, pp. 1–15.

[24] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. 13th ACM Conf. Embedded Netw. Sens. Syst.*, 2015, pp. 337–350.

[25] S. Ali, A. Ferdowsi, W. Saad, N. Rajatheva, and J. Haapola, "Sleeping multi-armed bandit learning for fast uplink grant allocation in machine type communications," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5072–5086, Aug. 2020.

[26] M. V. Ramakrishna and J. Zobel, "Performance in practice of string hashing functions," in *Proc. 21st Int. Conf. Distrib. Comput. Netw*, 1997, pp. 215–223.

Alakesh Kalita received the B.Tech. degree from Assam Don Bosco University, Guwahati, India, in 2012, and the M.Tech degree from Assam University, Silchar, India, in 2016.

He has been a Doctoral Researcher with the Department of CSE, Indian Institute of Technology Guwahati, Guwahati, since January 2018. His research interests include Internet of Things, wireless sensor network, and cloud computing.

Manas Khatua (Member, IEEE) received the Ph.D. degree from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2015.

He was an Assistant Professor with IIT Jodhpur, Jheepasani, India, from 2016 to 2018, and was a Postdoctoral Research Fellow with SUTD, Singapore, from 2015 to 2016. He has been an Assistant Professor with the Department of CSE, Indian Institute of Technology Guwahati, Guwahati, India, since 2018. He was associated with Tata Consultancy Services, Kolkata, India for two and half years, and worked as a Lecturer with BUIE, Bankura, India, for more than two years. His research interests include performance evaluation of communication protocols, Internet of Things, wireless LANs, sensor networks, and network security.

Dr. Khatua is a member of ACM.