# Channel Condition Based Dynamic Beacon Interval for Faster Formation of 6TiSCH Network

Alakesh Kalita and Manas Khatua, *Member, IEEE*

**Abstract**—Industrial applications of Internet of Things (IoT) demand high reliability, deterministic latency, and high scalability with energy efficiency to the communication and networking protocols. 6TiSCH is a time slotted channel hopping (TSCH) medium access control (MAC) protocol running under the IPv6 enabled higher layer protocols for industrial IoT (IIoT). In this paper, we theoretically analyze the network formation protocol in 6TiSCH network. Analysis reveals that the performance of the 6TiSCH network degrades when a pledge (new node) joins as it increases channel congestion by allowing to transmit beacon message. On the other hand, beacon transmission is essential to expand or reorganize the present network topology. To overcome this performance tradeoff, a channel condition based dynamic beacon interval (C2DBI) scheme is proposed in which beacon transmission interval varies with channel congestion status during network formation. Channel congestion status is estimated by each joined node in distributed manner, and subsequently changes its beacon generation interval to best fit with present condition. Finally the performance of C2DBI is compared with the minimal configuration standard and few other benchmark protocols. Analytical, simulation and real testbed results show that the proposed scheme outperforms the state of the art protocols in terms of joining time and energy consumption during network formation.

**Index Terms**—Industrial IoT, 6TiSCH, network formation, channel estimation

## 1 Introduction

The Industrial Internet of Things (IIoT) aims to provide better efficiency, higher scalability, time and cost savings for different industrial applications [1], [2]. It tries to reshape the current industrial sector by placing resource constrained sensors and actuators in the physical environment for real-time process monitoring and controlling operations. Efficiently collected and processed sensory data help in finding counterproductive output or predictive maintenance of an industry. In short, IIoT brings people, smart machines, reliable communications, and advanced analytics all together for monitoring, collecting, exchanging, and analyzing information. The IIoT demands that wireless communication protocols should provide high reliability, bounded latency, energy efficiency, and scalability. However, the most widely used protocols (e.g., IEEE 802.15.4 [3]) for resource constrained wireless networks miserably fail to provide such urgent requirements in IIoT. Therefore, IEEE 802.15.4e [4] has been proposed in which five new Medium Access Control (MAC) protocols (called MAC behavior modes) are mentioned to support different application wise requirements of IIoT. One of the modes is Time Slotted Channel Hopping (TSCH). TSCH uses time division multiple access with channel hopping and allows several parallel communications at a time. It can achieve low power consumption, high reliability, and data delivery in bounded latency.

• *The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam 781039, India. E-mail: alakesh.kalita1025@gmail.com, manaskhatua@iitg.ac.in.*

*Manuscript received 17 July 2019; revised 4 Mar. 2020; accepted 5 Mar. 2020. Date of publication 16 Mar. 2020; date of current version 3 June 2021. (Corresponding author: Alakesh Kalita.)*
*Digital Object Identifier no. 10.1109/TMC.2020.2980828*

To connect the resource constrained TSCH enable devices, used in industry, into Internet, *Internet Engineering Task Force (IETF)* created a 6TiSCH Working Group (WG). The main aim of this group is to establish interoperability between TSCH and IPv6 [5]. It provides an open communication stack to connect industrial multihop and lossy 802.15.4e TSCH networks to existing IPv6 networks. TSCH is limited in establishing global synchronization among the participated constrained devices. The 6TiSCH layer is necessary to fill the gap between the IETF low-power IPv6 communication stack (such as 6LoWPAN, RPL, CoAP) [6] and TSCH. Apart from the interoperability, the most important task is network bootstrapping by which pledges (new nodes) join into existing networks. During network bootstrapping, several requirements are essential to consider, such as less joining time, low energy consumption, and less resource allocation. For this purpose, 6TiSCH-WG defined a new bootstrapping protocol called 6TiSCH Minimal Configuration (6TiSCH-MC) [7]. This standard defines the minimum allocation of resources for control packets. According to the standard, only one shared slot in a slotframe (or multi-slotframe) is used for the communication of all types of control packets generated in a network. In 6TiSCH-MC, the allocation of resources is static; i.e., it does not vary with network condition. All the nodes (i.e., already joined nodes and pledges) present in a network should transmit/exchange their bootstrapping traffic and other control packets in the shared slots only. For transmitting/exchanging information in a shared slot, a node needs to contend for the channel using the random channel access protocol such as carrier sense multiple access with collision avoidance (CSMA/CA). During network formation, several control packets such as Enhanced Beacon (EB) frames are sent by already joined nodes, and then DIO packets [8] are sent by the same nodes to

construct Destination Oriented Directed Acyclic Graph (DODAG) topology following the routing protocol-Routing over Low Power and Lossy Networks (RPL) [8]. Further, for secure enrollment of nodes, a pledge and its parent (joined) node exchange join request (JRQ) and join response (JRS) frames [9] [10]. Exchanging of these frames to ensure secure joining are also done in shared slot. Therefore, the rate of transmitting these frames/packets in limited number of shared slots plays a significant role in network formation process of a 6TiSCH based IIoT network.

Several researchers (e.g., [11], [12], [13], [14], [15]) considered that only EB frame is transmitted during network formation process. The key focus of those works was to transmit as many EB as possible for faster formation of single hop TSCH network. This intension results in allocation of more bandwidth (i.e., shared slots). In a recent work, Vallati *et al.* [16] have proposed a multihop 6TiSCH network formation scheme considering both the EB frame and DIO control packets during network formation. They proposed a dynamic resource allocation scheme to provide enough resources (i.e., shared slots) to the nodes for transmitting their control packets. Earlier, Vucinic *et al.* [17] considered EB, DIO, and other control packets for secure joining in their network formation analysis. From the results of the analysis, they suggested that the beacon transmission probability should be 0.1 irrespective of the number of nodes present in the network for faster network formation. However, it is observed that allocation of several shared slots costs in higher energy consumption. We also notice that the performance of a 6TiSCH network formation degrades when a pledge joins in the network.

In this paper, the main target is to achieve an energy efficient scheme for faster association of pledges into the existing 6TiSCH network. At the outset, a Markov chain based analytical model is designed for estimating node joining time and energy consumption in a 6TiSCH network which follows a fixed beacon interval by default. Analytical results show that, due to the use of fixed beacon intervals, performance of network formation protocol degrades in terms of joining time and energy consumption. Subsequently, a *channel condition based dynamic beacon interval (C2DBI)* scheme is proposed for energy efficient faster joining in 6TiSCH network. In the proposed scheme, beacon generation interval of a joined node is varied based on the channel congestion status around the node in a network. A theoretical analysis is done for the proposed scheme to show the advantages of dynamic beacon interval over fixed beacon interval. Finally, the proposed method is validated by both the simulation on Cooja network simulator and the real testbed experiment on FIT IoT-LAB [18]. Note that this work is the extended version of our previous work [19]. In this version, an analytical model is added for computing node joining time and energy consumption of all nodes (i.e., pledge and joined) present in a multihop 6TiSCH network. It is also considered that the nodes have different number of neighbor nodes, i.e., all nodes can not hear from each other. More numerical results on node joining time and energy consumption are included to show the effect of fixed beacon interval. In our previous work, we noticed that the beacon interval of a joined node is less affected by the minimum value of network congestion around the node. Therefore, in this work, we update the procedure for calculating the beacon interval of a joined node. Furthermore, we include the theoretical analysis of the proposed scheme and compare the various results obtained from it with the fixed interval based 6TiSCH-MC scheme. More simulation results are added on node joining time using different slotframe lengths such as 33, 67, 101 *timeslots* and using realistic Multipath Ray-tracer Medium (MRM) channel model compared to our previous work to show the insight of dynamic beacon interval on node joining time and energy consumption. Apart from simulations, comparison based real testbed experiment results are also added for validating the proposed approach.

We summarize the contributions of this paper as follows,

- A Markov chain model is provided for analyzing the node joining process in a multihop 6TiSCH network.
- Analytical results show the demerits of fixed beacon interval scheme during network formation with respect to node joining time and energy consumption.
- A *channel condition based dynamic beacon interval (C2DBI)* scheme is proposed for efficient joining of nodes in 6TiSCH network based on channel congestion status.
- Theoretical analysis of the proposed scheme is provided to compute node joining time and energy consumption.
- Comparison based simulations on Cooja simulator and real testbed experiments on FIT IoT-LAB testbed are conducted to validate the proposed scheme.

The rest of the paper is organized as follows. Section 2 describes the background of 6TiSCH network and its formation process. Section 3 summarizes the existing works related to network formation in 6TiSCH network. In Section 4, an analytical model for network formation is provided to show the demerits of fixed beacon interval. Section 5 presents our proposed channel condition based dynamic beacon interval scheme. Sections 6 and 7 present the performance evaluation of the proposed scheme using simulation and real testbed, respectively. Finally, this paper is concluded in Section 8.

## 2 Network Formation in 6TiSCH

IETF 6TiSCH is an open communication stack used to connect low power 802.15.4e TSCH enable devices into IPv6 networks. It is a low power wireless technology used in IIoT to provide seamless internet connection among the tiny low power devices. TSCH is limited to global synchronization and does not deal with the scheduling of communication slots. Therefore, 6TiSCH provides scheduling of both the dedicated and shared slots among the nodes, adapting a sublayer called 6top [20]. 6TiSCH also deals with resource allocation during network bootstrap. It defines how a pledge securely joins into an existing 6TiSCH network. For basic interoperability and bootstrapping, 6TiSCH defines Minimal Configuration (6TiSCH-MC) protocol. The 6TiSCH-MC deals with resource allocation during network bootstrap. 6TiSCH-MC suggests to have only a single shared slot in each slotframe or multiple-slotframe. Again, for secure enrollment of nodes, JRQ and JRS frames are exchanged between a pledge and its parent node in shared slot. Therefore, including all the bootstrapping traffics, other control packets are also transmitted in shared slot.

At first, a 6TiSCH network formation is initiated by a Personal Area Network coordinator (PC). The PC periodically broadcasts enhanced beacon (EB) frames for advertising the

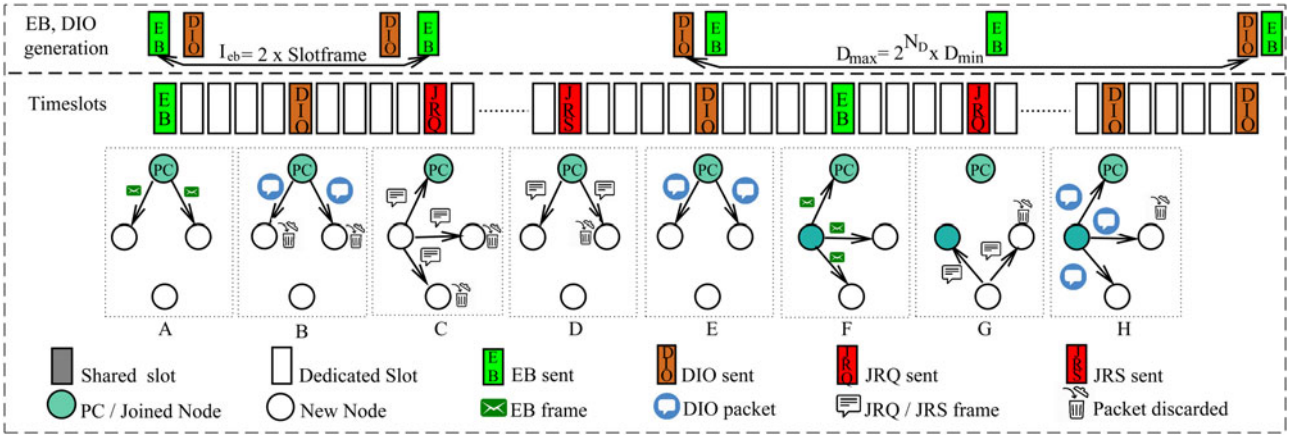



Fig. 1. 6TiSCH network formation.

availability of a network. Each beacon contains all the necessary information required for a pledge to join into the network. A pledge passively scans all the available channels (at most 16 channels according to 6TiSCH-MC) to receive an EB. This scanning is required as a pledge does not know in which channel transmission of EB will occur. Therefore, a pledge starts scanning a random channel and changes into another channel after a specified interval if it does not receive any valid EB in the current channel. This scanning process is repeated over time until a pledge gets a valid EB from the PC or an already joined node, i.e., parent. Once a pledge receives a valid EB, it gets synchronized into the network. That means it gets all the information such as timeslot duration, channel hopping sequence, slotframe/multi-slotframe size, etc., that are currently being used for communication in the network. To save its energy, a pledge sets its wake up schedule to get/send other information rather than remaining active for the entire slotframe duration once it gets a valid EB frame. After getting synchronized with the network, a pledge sends join request (JRQ) frame to its parent in shared slot for secure enrollment with the network. As a response, the parent node sends join response (JRS) frame. Once the pledge gets a valid JRS frame, it is considered as a securely enrolled node. In this stage, although a pledge is securely admitted in the network but not allowed to send its own beacon for further expansion of the network. It is allowed to do so only after getting at least one routing information packet i.e., DODAG Information Object (DIO) packet from its currently assigned parent node. A DIO packet contains all the necessary routing configuration parameters required for a node to reach the RPL root (i.e., PC). The rates of sending EB frames and DIO packets are independent of each other as both of them are handled by different layers in the protocol stack. Generally, the EB generation rate depends on the network administrator/joined node, and the DIO generation rate is governed by the trickle algorithm [21] in the network layer. The trickle algorithm generates a DIO packet within a DIO interval. The DIO generation interval varies with network condition and time. Initially, the interval starts with a minimum duration $D_{min}$. In a stable network, this interval doubles at the end of each interval until it reaches the maximum value $D_{max}$. For any kind of inconsistency in the network, it resets to minimum value $D_{min}$. This way every pledge becomes a joined node, and then, is eligible to send its own beacon. Network formation ends when all the pledges join the network.

An example of network formation process of a small network is presented in Fig. 1. In state A, pledges get EB frame from the PC. From the received EB frame, pledges adapt the underlying TSCH network information. In state B, pledges discard the received DIO packet as secure enrollments of the pledges are not done yet. In the next shared slots, both the pledges continuously try to send their JRQ frames to the PC. In state C, one of the pledges sends its JRQ frame, and in a later shared slot, it receives a valid JRS frame from its parent. Thus, the pledge successfully joins the secure TSCH network and waits for a DIO packet to complete its network admission process. In state E, the pledge receives a fresh valid routing information packet (i.e., RPL DIO packet) from the RPL root (i.e., PC). Now, the pledge becomes a joined node and eligible to send its own beacon by encapsulating various information in its beacon. In the next state F, the newly joined node sends its own beacon frames for further expansion of the network. In state G, one more pledge sends its JRQ frame to the newly joined node and receives JRS frame. This pledge will become a joined node once it receives a valid routing information packet from its parent node which is shown in state H. In this way, every pledge in a network joins into the network one by one and completes the network formation process.

## 3 Literature Review

At the very beginning, researchers considered only EB frame for single hop TSCH network construction. A node is said to be a joined node if it successfully receives an EB frame from an already joined node. The researcher did not consider the selection of a routing path to reach the RPL root for the completion of joining process. Recently, many works started considering multihop network formation where both the EB and DIO control packets are used for network formation. DIO control packet is used for the construction of the path to reach the RPL root in multihop fashion. Therefore, depending on the previous work of different authors, we divide our literature review into two parts. In the first part, only the works that considered EB frame for network formation are included. In the second part, the works considering both the EB and DIO packets for network formation are included.

### 3.1 TSCH Network Formation

In TSCH network formation, different schemes were proposed for transmitting the EB frames efficiently to form the network rapidly. Guglielmo *et al.* [12] analyzes the TSCH network formation using a simple Random-based Advertisement (RA) algorithm for sending EBs. In RA, joined nodes send their beacons in the random advertisement slots assigned by their parents. Here, the possibility of EB collision is reduced by varying transmission probabilities of EB frames. Vogli *et al.* [11] proposed two algorithms for TSCH synchronization – Random Vertical filling (RV) and Random Horizontal filling (RH). In RV, PC is allowed to send EB using channel offset 0, while the other joined nodes use any random channel offsets. All the PC and other joined nodes send their EB at the first timeslot of the first slotframe of a multi-slotframe. Each PC/joined node sends a single EB in a multi-slotframe. Whereas in RH, including the PC, all the joined nodes send their EBs using channel offset 0. The PC is allowed to send EB at the first time slot of the first slotframe, whereas the other joined nodes randomly send at the first timeslot of any slotframe in a multi-slotframe. Again, Vogli *et al.* [13] proposed the enhanced version of RV and RH, namely Enhanced Coordinated Vertical filling (ECV) and Enhanced Coordinated Horizontal filling (ECH) algorithms, respectively, for faster TSCH synchronization. Assuming that PC has enough energy resources, they proposed to increase the number of beacons send in ECV and ECH. PC sends EB frame in every slotframe of a multi-slotframe using channel offset 0 in both the approaches. The other joined nodes send beacon either by selecting available slot offset or channel offset. Thus, by sending more beacons, they have tried to reduce the node synchronization time. Guglielmo *et al.* [14] mentioned a Model-based Beacon Scheduling (MBS) algorithm. Here, PC assigns links to all joined nodes to transmit their EBs by solving an optimization problem. They also mentioned that equal spaced shared slots in a multi-slotframe helps in faster network formation. Khoufi *et al.* [22] mentioned that beacon interval period and slotframe size have effects in the selection of a physical channel for transmitting beacons. They proposed to use fixed advertising slot instead of selecting any slot as an advertising slot. To obtain this, they proposed a Deterministic Beacon Advertising (DBA) algorithm for transmitting beacons over all the available frequencies without any collision.

### 3.2 6TiSCH Network Formation

Only a few works have been done for network formation in 6TiSCH network. Vallati *et al.* [16] mentioned that a pledge should receive both the EB frame and DIO packet in order to completely join into a 6TiSCH network. They also analytically proved that the 6TiSCH-MC does not provide enough resources to send control packets generated during network bootstrap. To overcome the problem of unavailability of sufficient resources (i.e., shared slots), the authors allocate more number of shared slots to transmit more control packets in less time. The allocation of extra shared slots costs in higher energy consumption and also hampers in data packet transmission. Further, it does not follow the 6TiSCH-MC standard in terms of the use of number of shared slots in each slotframe/multi-slotframe. Further, the authors did not consider resource usage by the nondeterministic join traffic for secure enrollment. Vucinic *et al.* [17] considered the join traffic for secure enrollment along with the EB and DIO packets in 6TiSCH network formation. Consequently, they proposed that the beacon transmission probability should be 0.1 irrespective of the number of nodes present in the network for faster network formation.

TABLE 1
Existing Works in 6TiSCH Network Formation

| Procedure | Control packets considered | | | Shared slots | DCC ◦ | DBI* |
|---|---|---|---|---|---|---|
| | EB | DIO | Join | | | |
| RA [12] | ✓ | x | x | > 1 | ✓ | x |
| RV & RH [11] | ✓ | x | x | > 1 | x | x |
| MBS [14] | ✓ | x | x | > 1 | x | x |
| DBA [22] | ✓ | x | x | > 1 | x | x |
| ECV, ECH [13] | ✓ | x | x | > 1 | x | x |
| Orchestra [23] | ✓ | ✓ | x | > 1 | x | x |
| 6TiSCH-MC [7] | ✓ | ✓ | ✓ | 1 | x | x |
| DRA [16] | ✓ | ✓ | x | > 1 | ✓ | x |
| BS [17] | ✓ | ✓ | ✓ | 1 | x | x |
| This paper | ✓ | ✓ | ✓ | 1 | ✓ | ✓ |

◦ *DCC: Dynamic congestion control.*
**DBI: Dynamic Beacon Interval.*

Apart from this TSCH synchronization and 6TiSCH network formation, the authors of [23] 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 consider the network formation task. Further, the authors evaluated their proposed method in a static network.

It is observed that the performance of a 6TiSCH-MC based network degrades when the number of joined nodes increases. It is because of the static allocation of resources by 6TiSCH-MC. Again, increasing amount of resource allocation for bootstrapping traffic or control packet transmission hampers the flows of sensory data packets, and also consumes more resources. Therefore, in this paper, we propose a dynamic beacon interval allocation scheme in which the beacon interval varies with network congestion in the network. In Table 1, we have tabulated the important features of existing works related to 6TiSCH network formation and compared them with the proposed scheme C2DBI.

## 4 Network Formation Analysis

In this section, 6TiSCH network formation procedure is analyzed considering the similar resource allocation scheme as mentioned in 6TiSCH-MC. In this analysis, secure enrollment of nodes during network formation is also considered. Hence, transmissions of all bootstrapping control traffic such as EB, DIO, JRS, and JRQ happen in shared slots during network formation. We model the node's joining time and energy consumption during network formation. Finally, it is shown that the existing 6TiSCH-MC method for 6TiSCH network formation contributes to higher join time and energy consumption during network formation. At the outset, the symbols in the analytical model, and their corresponding meaning are tabulated in Table 2.

### 4.1 Analytical Model for Joining Time

We model the behavior of a pledge during network formation using Markov chain. During network formation, each node passes through four states. Fig. 2 shows the different

TABLE 2
List of Frequently Used Symbols

| Symbol | Meaning |
|---|---|
| $N_c$ | Number of available channels |
| $L$ | Slotframe length |
| $I_{eb}$ | Beacon generation interval |
| $I_{eb}^{min}$ | Minimum beacon interval |
| $I_{eb}^{max}$ | Maximum beacon interval |
| $P_{eb}$ | Transmission probability of EB in shared slot |
| $P_{dio}$ | Transmission probability of DIO in shared slot |
| $P_{jrq}$ | Transmission probability of JRQ in shared slot |
| $P_{jrs}$ | Transmission probability of JRS in shared slot |
| $P_{loss}$ | Packet loss probability |
| $P_{EB_S}$ | Successful transmission probability of EB |
| $P_{DIO_S}$ | Successful transmission probability of DIO |
| $P_{JRQ_S}$ | Successful transmission probability of JRQ |
| $P_{JRS_S}$ | Successful transmission probability of JRS |
| $P_{jns}$ | Probability of transmission in shared slot by a joined node |
| $P_{nns}$ | Probability of transmission in shared slot by a new node |
| $P_{ts}$ | Probability of transmission in shared slot |
| $P_{empty}$ | Probability of no transmission in shared slot |
| $W$ | Time interval for CBR calculation |
| $T_i$ | Timeslot duration of $i$th slot |
| $CBR$ | Channel busy ratio |
| $E_{RxData}$ | Energy require to receive a data packet |
| $E_{TxData}$ | Energy require to transmit a data packet |

states of the Markov model. The absorbing state at the end indicates that a pledge has joined with the existing network successfully. That means a pledge has become a joined node, and now, it can send its own beacon for further expansion of the network.

At the initial state, a pledge waits for an EB from the PC or any already joined node. Once a pledge receives a valid EB frame with probability $P_{EB_S}$, it moves to second state. The second state represents that a pledge gets synchronized with the TSCH network and waiting to complete its secure enrollment to the network. For secure enrollment, a pledge sends JRQ frame and waits for the corresponding JRS frame. After finishing the secure enrollment process with probability $P_{join}$, a pledge moves to the third state. In the third state, a pledge waits for routing protocol information. Once it receives at least one fresh/recent DIO packet from its preferred parent, the pledge moves to its final absorbing state. In the absorbing state, a pledge successfully joins the TSCH 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,

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

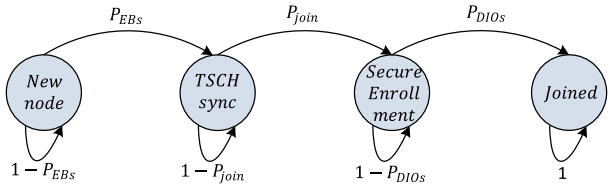



Fig. 2. Markov Chain model of node joining process.

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 multihop network model as shown in Fig. 3, where each node has a different number of neighbors. Let $P_{eb}$, $P_{dio}$, $P_{jrq}$, $P_{jrs}$ are the transmission probabilities of EB, DIO, JRQ and JRS frame/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 by a pledge in a shared slot 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 neighbors of a pledge. The values of $n$ and $M$ can be different for every node present in a network. The above equation follows that the $(M-n)$ pledges join a network one by one, and the $P(N=n)$ denotes the probability that, at an instant, total number of joined neighbors is $n$. Again, when a joined node sends its own EB frame, the remaining $(n-1)$ joined nodes should not send their EB frames, DIO packets, and JRS frames. The probability of this condition equals to $(P_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1})$. Additionally, the neighbors but pledges also should not send any JRQ frames, which is defined by the probability equals to $((1-P_{jrq})^{M-n})$. Further, the transmitted frame should not be lost in the channel, which is computed by probability $(1-P_{loss})$. 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 $n$ joined neighbor nodes, which result in multiplying the computed probability by $n$. Note that a pledge can send JRQ frame only after receiving a valid EB frame, and a joined node sends JRS frame only after receiving 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 slot is computed as,

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

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 probabilities of sending a JRQ frame and receiving a JRS frame successfully in a shared slot are also computed as,

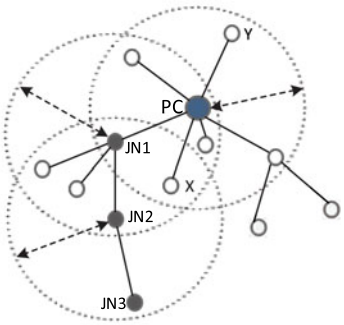



Fig. 3. An example of multihop network topology used in this work.

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

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

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

$$P_{eb} = \frac{L}{I_{eb}}, \quad (6)$$

where, $L$ is the slotframe length and $I_{eb}$ is the beacon interval. We compute the probability $P_{dio}$ following the procedure described in [16] as follows,

$$P_{dio} = (1-P_{eb})\frac{\begin{array}{c}2^{N_D}(1-P_r)^{N_D}min(\frac{L}{2^{N_D}D_{min}},1)\\ +\sum_{i=0}^{N_D-1}P_r2^i(1-P_r)^i min(\frac{L}{2^iD_{min}},1)\end{array}}{\begin{array}{c}P_r+2^{N_D}(1-P_r)^{N_D}\\ +\sum_{j=1}^{N_D-1}P_r2^j(1-P_r)^j\end{array}}, \quad (7)$$

where, $D_{min}$ 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,

$$AJT = PJT + ASF \times L, \quad (8)$$

where, $PJT$ is the average joining time of the parent node of a pledge in multihop network. For example, in Fig. 3, joining time of node $JN2$ is the summation of its own joining process time and its parent node's joining time (joining time of $JN1$ or $X$ whichever $JN2$ selects as its parent).

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

## 4.2 Analytical Model for Energy Consumption

During network formation process, two different types of nodes are present in the network. One is joined node, i.e., the nodes which have already joined in the network, and the other is pledge (new node) who wants to join in the network. During network formation, these types of node consume different amount of energy because of their different activities. This is because a joined node needs to broadcast or receive control packet once in each slotframe or multi-slotframe, whereas a pledge needs to remain active until it receives a valid EB.

In this analytical model, we do not consider the microcontroller states of a node to keep the analytical model as simple as possible. Only the radio state of a node during network formation is considered. The following subsections briefly describe the computation method of $E_{newNode}$ and $E_{joinedNode}$ respectively.

### 4.2.1 Energy Consumption by a Pledge

Referring to the Markov Chain model shown in Fig. 2, the average number of slotframes (ASF) required to receive an EB frame successfully can be calculated as, $ASF_{EB} = 1/P_{EB_S}$ During these $(1/P_{EB_S})$ slotframes, a pledge needs to keep active its radio to get a valid EB frame. Therefore, the average time a pledge needs to keep active its radio is,

$$Average\ Time_{EB} = PJT + L/P_{EB_S}. \quad (9)$$

Note that in the above equation parent node joining time is also added. It is because the parent node can send EB frame only after finishing its joining process. Converting this time duration into number of timeslots, we get,

$$Average\ Slot_{EB} = \frac{Average\ Time_{EB}}{timeslot\ duration}. \quad (10)$$

Therefore, average energy consumption by a pledge for successfully getting an EB frame is computed as,

$$Average\ Energy_{EB} = \frac{PJT \times P_{EB_S} + L}{T \times P_{EB_S}}\ (E_{RxData}), \quad (11)$$

where, $E_{RxData}$ is the energy consumption for receiving a frame or packet in a timeslot, and $T$ is the duration of each timeslot. Similarly, average energy consumption for successfully receiving a DIO packet is,

$$Average\ Energy_{DIO} = \frac{1}{P_{DIO_S}}(E_{RxData}). \quad (12)$$

Note that the variables $L$ and $T$ are not used in Equation (12) because a pledge is already synchronized with the network after getting a valid EB. Likewise, the average energy consumption by a pledge to send a JRQ and to receive its corresponding JRS frame successfully can be calculated as,

$$Average\ Energy_{JRQ} = \frac{1}{P_{JRQ_S}}(E_{TxData}) \quad (13)$$

$$Average\ Energy_{JRS} = \frac{1}{P_{JRS_S}}(E_{RxData}), \quad (14)$$

where, $E_{TxData}$ is the amount of energy consumed to transmit a packet. Finally, the average energy consumption by a pledge during its network admission process is,

$$\begin{aligned}E_{newNode} =& Average\ Energy_{EB} + Average\ Energy_{DIO}\\ &+ Average\ Energy_{JRQ} + Average\ Energy_{JRS}.\end{aligned} \quad (15)$$

### 4.2.2 Energy Consumption by a Joined Node

Average energy consumption by a joined node is the summation of energy consumption when it was a pledge and after

becoming a joined node. A joined node needs to transmit different control packets for forming the rest of the network. For example, in Fig. 3, $JN2$ needs to broadcast different control packets (e.g., EB, DIO, JRQ, JRS) so that $JN3$ can join the network. The number of such control packets transmitted by any joined node can be estimated as follows,

$$C_{EB} = \frac{ASF_{multihop}}{EB\ transmitting\ interval} = ASF_{multihop} \times P_{eb} \tag{16}$$

$$C_{DIO} = ASF_{multihop} \times P_{dio}, \tag{17}$$

where, $ASF_{multihop}$ denotes the average number of slotframes required to form the rest of the network, which can be calculated by summing all hop-by-hop ASF values. Again, the total number of JRQ frames received by a joined node can be calculated as,

$$C_{JRQ} = \frac{ASF_{multihop}}{JRQ\ transmitting\ interval} \times No\ of\ new\ nodes = ASF_{multihop} \times P_{JRQ_S} \times K, \tag{18}$$

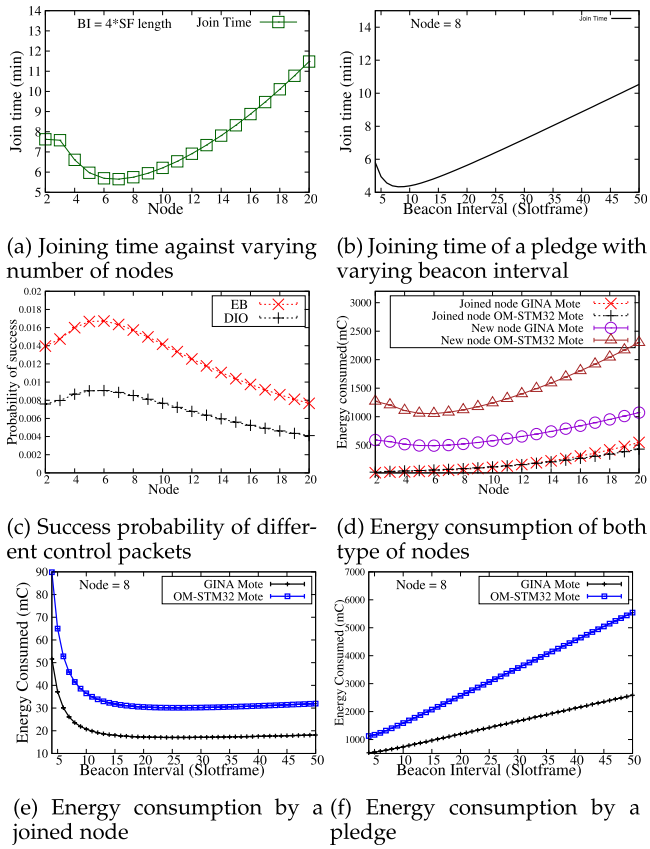
where, $K$ is the number of pledges to be admitted in the network. As a response to JRQ frames, same number of JRS frames also need to be transmitted. Therefore, $C_{JRS} = C_{JRQ}$, where, $C_{JRS}$ is the total number of JRS frames. Now, we compute the total energy consumption by a joined node or PC as follows,

$$E_{joinedNode} = E_{newNode} + (C_{EB})E_{TxData} + (C_{DIO})E_{TxData} + (C_{JRQ})E_{RxData} + (C_{JRS})E_{TxData} \tag{19}$$

where, $E_{TxData}$ and $E_{RxData}$ are the amount of energy consumed by a joined node during transmission and reception of a packet in a slotframe, respectively. For the PC, the value of $E_{newNode}$ is 0, and for the other joined nodes, $E_{newNode}$ depends on their joining times.

### 4.3 Analytical Results

In this section, we graphically present the above analysis and show the limitations of existing 6TiSCH network formation procedure. Let us consider a network with the following given values: $N_C = 16$, $I_{eb} = 4 * L$, $P_{loss} = 0.2$, $D_{min} = 8\ ms$, $L = 101\ timeslots$, $T_i = 10\ ms$, $N_D = 16$ and $P_r = 0.2$. We compute the average joining time (AJT) and energy consumption of nodes during node joining process using the above analytical models. The received results are plotted in Fig. 4. Fig. 4a depicts that, for a fixed beacon interval, the joining time of a pledge increases gradually with the increased of number nodes after a certain threshold. In other words, the performance of the network degrades once it allows to join a pledge with it. It can be seen that, initially, increasing value of nodes helps the pledges to join in less time, because more number of EB frames are transmitted by several joined nodes. But when the number of nodes increases further, due to the limited number of shared slots, contention among the participating nodes also increases. This, in turn, results in higher joining time. Fig. 4b shows the results of variable beacon interval with fixed number of nodes, i.e., 8 nodes. The figure shows that joining time is high in case of low beacon intervals as well as at high beacon intervals. In low beacon intervals, contention among the joined nodes increases as all the joined nodes want to send their beacons frequently. This, in turn, increases the time




(a) Joining time against varying number of nodes

(b) Joining time of a pledge with varying beacon interval

(c) Success probability of different control packets

(d) Energy consumption of both type of nodes

(e) Energy consumption by a joined node

(f) Energy consumption by a pledge

Fig. 4. Numerical results on average joining time and energy consumption of nodes during 6TiSCH network formation with increasing values of joined node and beacon interval.

period to get a valid EB by a pledge. In high beacon intervals, a pledge has to wait for long time to get a beacon. Therefore, once again the joining time increases with the increase in beacon interval. Fig. 4c shows the change in success probabilities of EB, DIO packets with the varied number of nodes. Because of the increasing number of joined nodes, congestion also increases, which results in low success probabilities. For calculating the energy consumption of the joined nodes and pledges, two different types of wireless motes, GINA and OM-STM32 [24], are considered. These two types of nodes are considered to show the effect of fixed beacon interval scheme in energy consumption. Both the GINA and OM-STM32 motes consume 69.6 and 119.2 micro coulombs, respectively, for transmitting a packets. Again, the same motes consume 72.1 and 154.8 micro coulombs for receiving a packet, respectively.

Considering this energy consumption data, we plot the average energy consumption graph of a joined node and a pledge by varying the number of nodes and beacon intervals in Fig. 4d, 4e, and 4f, respectively. It can be observed in Fig. 4d that the energy consumption by a joined node almost remains unchanged with respect to varying number of nodes. This is because the sending rate of control packets by a joined node is not significantly changed when a pledge joins into the network in fixed beacon interval scheme. On the other hand, energy consumption by a pledge increases with the increased number of nodes. It is because the contention in shared slots increases with the increased number of joined nodes. A pledge needs to wait for a long time to get a valid EB. Again, it can be seen that a joined node consumes less energy than a pledge.

This is because a pledge has to actively scan entire slotframes duration to get a valid EB, whereas a joined node sends/receives only in a single timeslot in a slotframe/multi-slotframe. Fig. 4e shows the energy consumption of a joined node with varying beacon interval. It can be observed that energy consumption by a joined node almost remains same with the increase in beacon interval. Whereas a different scenario is observed for the pledges as shown in Fig. 4f. Here, energy consumption increases with the increase of beacon interval. During high beacon interval, though the congestion in shared slot is reduced but a pledge has to wait for a long time to get a valid EB. During this entire time period the pledge needs to keep active its radio all the time which consumes more energy, and thus, energy consumption increases again.

From these analytical results, it can be established that the performance of a network decreases with the increased number of joined nodes, i.e., when a pledge joins in the network. Further, it is also important to keep beacon interval in optimal range to maintain the joining time and energy consumption as low as possible.

## 5 Proposed Methodology

From the previous analysis, it can be seen that joining time of a pledge increases with the increasing number of nodes in a network that follows fixed beacon interval scheme. The number of nodes has a significant impact on the energy consumption of a pledge. To reduce both the joining time and energy consumption during network formation, we propose a dynamic beacon interval scheme.

### 5.1 Dynamic Beacon Interval

As the congestion in shared slot increases with the increasing number of joined nodes, it is important to reduce congestion in shared slots while the pledges are joining one by one. For this purpose, we propose a scheme to dynamically adjust the beacon interval depending upon the congestion status in shared slots. When congestion is increased in the shared slots, the proposed method increases the beacon interval so that the number of beacons generated by joined nodes decreases. This decreasing rate of beacon generation reduces the congestion, and thus gives opportunities to other control packets to be transmitted in the network. On the other hand, when the congestion is low in the shared slots, the beacon interval is reduced so that a pledge can quickly get a valid beacon. This in turn reduces the energy consumption of a pledge as the pledge needs to keep active its radio until it synchronizes with the TSCH network. Now the congestion estimation can be done using different methods considering many parameters such as number of neighboring nodes, packet loss, queue length, and channel load. In this work, channel busyness ratio (CBR) [25] is used for congestion estimation as it gives better results in both the static and dynamic networks. Static network means when the nodes are in fixed position, but the nodes are mobile in dynamic network. The CBR is measured for a fixed interval, and based on the measured CBR value, the beacon interval of a joined node is calculated and assigned. The assigned beacon interval will be used in the next time period. The CBR for a particular period of time is calculated as,

$$CBR = \frac{Busy\ shared\ slots}{Busy\ shared\ slots + Empty\ shared\ slots}. \tag{20}$$

A shared slot is considered to be busy if the measured signal strength in that slot is higher than the clear channel assessment (CCA). Otherwise, a shared slot is considered as an empty slot. Note that the CCA based channel quality estimation is less affected by the overcrowded 2.4 GHz frequency band technologies as CCA is performed in different physical channel in every shared slot according to channel hopping feature of TSCH. After computing the CBR for a particular time interval, a joined node computes its own beacon interval for the next time as follows,

$$I_{eb} = \begin{cases} I_{eb}^{min} & \text{if CBR} = 0 \\ I_{eb}^{min} + (I_{eb}^{max} - I_{eb}^{min})^{CBR} & \text{otherwise} \end{cases}. \tag{21}$$

Using the above equation, a joined node always choose its beacon interval in between the minimum beacon interval $I_{eb}^{min}$ and maximum beacon interval $I_{eb}^{max}$ as the values of CBR varies between 0 to 1. Algorithm 1 describes the method of adjusting beacon interval dynamically based on present channel congestion status.

**Algorithm 1.** Channel Condition Based Dynamic Beacon Interval Scheme (C2DBI)

**Input**: $N_i$ : Node i; $W$ : CBR period; $T_t$ : Current time; $T_{cbr}$ : Time instant of last CBR calculation
**Output**: $I_{eb}$ for the next period

**for** *each timeslot* $T_t$ **do**
  **if** *the current* $Link_{type}$ *is Shared* **then**
    increment $totalSharedSlot$ variable by unity
    **if** *current* $CCA_{status}$ *is busy* **then**
      increment $busySlot$ variable by unity
    **end**
  **end**
  **if** *the difference between* $T_t$ *and* $T_{cbr}$ *is greater than or equal to* $W$ **then**
    $CBR_{(N_i,T_{cbr},W)} = busySlot/totalSharedSlot$
    **if** *the CBR is not equal to 0* **then**
      $I_{eb} = I_{eb}^{min} + (I_{eb}^{max} - I_{eb}^{min})^{CBR}$
    **else**
      $I_{eb} = I_{eb}^{min}$
    **end**
    Update $T_{cbr}$ by $T_t$
    Reset $busySlot$ and $totalSharedSlot$ to 0
  **end**
**end**

### 5.2 Theoretical Modeling of Dynamic Beacon Interval

Let us consider that $P_{jns}$ is the probability of an already joined node transmits a packet in a shared slot and $P_{nns}$ is the probability that a pledge transmits a packet in a shared slot. If $P_{ts}$ denotes that there is a transmission in a shared slot, then $P_{ts}$ can be written as,

$$P_{ts} = P_{jns} + P_{nns}. \tag{22}$$

Now, an already joined node transmits all the EB, DIO, JRS control packets. So, we can write $P_{jns}$ as follows,

$$P_{jns} = P_{eb} + P_{dio} + P_{jrs}, \tag{23}$$

where, $P_{eb}$ and $P_{dio}$ are the transmission probabilities of sending an EB frame and a DIO packet in a slotframe, respectively. And $P_{jrs}$ is the probability of transmitting JRS frame after getting valid JRQ frame. On the other hand, a pledge only sends JRQ frames during network formation. Hence, $P_{nns}$ can be computed as,

$$P_{nns} = P_{jrq}, \tag{24}$$

where, $P_{jrq}$ is the transmission probability of a JRQ frame in a slotframe. Now, considering both the $P_{jns}$ and $P_{nns}$, the Equation (22) can be rewritten as,

$$P_{ts} = P_{eb} + P_{dio} + P_{JRQ_S} + P_{EB_S}, \tag{25}$$

where, $P_{jrq} = P_{EB_S}$ as a pledge sends JRQ frame only after receiving a valid EB frame, and $P_{jrs} = P_{JRQ_S}$ as a joined node sends JRS frame only after getting a valid JRQ frame from a pledge.

The same multihop network model is considered, as shown in Fig. 3, for evaluating the proposed scheme. As different joined nodes may have different number of neighbor nodes, the calculated CBR values of the joined nodes are also different. Let's assume that joined node $k$ has $nn$ number of pledges and $jn$ number of joined nodes in its communication range. These nodes use the same shared slot for transmitting their control packets. Sometimes, it is also possible that there is no transmission in a shared slot. These types of slots are known as empty slots. The probability of such an empty shared slot is,

$$P_{Empty} = (1 - P_{ts})^{nn+jn} \tag{26}$$

And the probability of having at least one transmission in a shared slot is,

$$P_T = 1 - (1 - P_{ts})^{nn+jn} \tag{27}$$

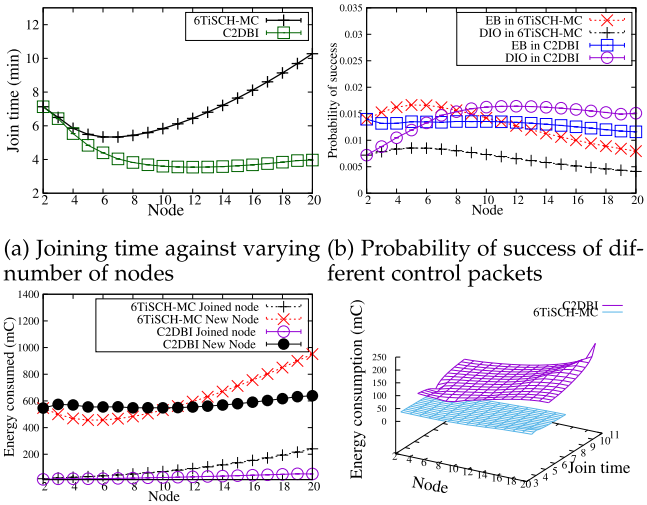
Now, the CBR at a particular time interval for node $k$ can be calculated as follows,

$$\begin{aligned} CBR_k &= \frac{Busy\ shared\ slots}{Busy\ shared\ slots + Empty\ shared\ slots} \\ &= \frac{P_T}{P_T + P_{Empty}}. \end{aligned} \tag{28}$$

Note that $P_T$ and $P_{Empty}$ are computed with respect to node $k$. This is because a CBR time interval consists of several shared timeslots. In each timeslot, either a transmission occurs or there is no transmission. Now, if we consider that the CBR calculation time interval consists of $W$ shared slots, then the Equation (28) can be written as follows,

$$CBR_k = \frac{\sum_{i=0}^{W}(1 - (1 - P_{ts})^{nn+jn}) * T_i}{\sum_{i=0}^{W}\left(1 - (1 - P_{ts})^{nn+jn} + (1 - P_{ts})^{nn+jn}\right) * T_i}, \tag{29}$$

where, $T_i$ denotes the shared slot duration in $i$th slotframe. As all the shared slot durations are same, we can simplify the above equation as follows,




(a) Joining time against varying number of nodes

(b) Probability of success of different control packets

(c) Energy consumption of both type of nodes

(d) Join time and energy consumption of a pledge

Fig. 5. Comparing numerical results of the proposed scheme (C2DBI) with 6TiSCH-MC.

$$CBR_k = 1 - (1 - P_{ts})^{nn+jn} = P_T \tag{30}$$

Now, the calculated CBR value can be directly used in the dynamic beacon interval formulation as shown in Equation (21) to calculate next beacon interval of node $k$ i.e., $I_{eb}^k$.

When different joined nodes have different number of neighbors, then their calculated CBR value will also be different. This results in varied beacon interval of each joined node. Now, with the proposed dynamic beacon interval approach, the successful EB frame receiving probability can be calculated as follows,

$$P_{EB_S} = \frac{\sum_{n}^{M-n} nP_{eb}^k \prod_{i=1,i\neq k}^{n-1}(1 - P_{eb}^i)(1 - P_{dio})^{n-1}}{(1 - P_{jrs})^{n-1}(1 - P_{jrq})^{M-n}(1 - P_{loss})P(N = n),} \tag{31}$$

where, $P_{eb}^i$ denotes the EB frame transmitting probability in a shared slot by the joined node $i$, which is calculated by $\frac{L}{I_{eb}^i}$. Similarly, we can calculate the successful packet transmission probabilities in a slotframe for other control packets such as $P_{DIO_S}$, $P_{JRQ_S}$, $P_{JRS_S}$ using varied beacon interval, i.e., $I_{eb}^i$ of each joined node $i$.

### 5.3 Analytical Results of Dynamic Beacon Interval

Assigning similar values to the variables as mentioned in Section 4, few graphs are plotted for the proposed method and compared them with the fixed beacon interval based 6TiSCH-MC scheme. Fig. 5a depicts the join time with respect to the number of nodes for the proposed method, and also compares it with the 6TiSCH-MC method. Here, we take $I_{eb}^{min} = 404\ milliseconds$, which is the same value used in 6TiSCH-MC scheme. Let $I_{eb}^{max} = 1010\ milliseconds$. These values are taken randomly, however any other value can also be considered. It is observed in Fig. 5a that the proposed model outperforms the 6TiSCH-MC model. Fig. 5b depicts the reason behind it. This graph shows the probabilities of successfully transmitting different control packets during network formation in both the schemes. When the number of joined nodes

TABLE 3
Simulation Parameters

| Parameter | Value |
|---|---|
| Operating System | Contiki 4.4 |
| MAC protocol | 802.15.4e TSCH |
| Number of channels | 16 |
| Timeslot length | 10 ms |
| Slotframe length | 33,67,101 timeslots |
| RPL version | RPL Lite |
| RPL DIO interval | Trickle Algo. |
| Keep-alive timeout | 30 secs |
| Mote type | Cooja Mote |
| Propagation model | MRM |
| Simulation duration | 120 mins |

increases, congestion in the shared slot also increases. This increasing congestion reduces the success probabilities of different control packets in 6TiSCH-MC. Whereas, in C2DBI, congestion is reduced by increasing the beacon generation interval, which in turn improves the success probabilities of different control packets. The energy consumption graph of GINA motes for both the 6TiSCH-MC and the proposed C2DBI schemes is shown in Fig. 5c. The plotted graph shows how the proposed scheme maintains almost stable energy consumption during heavy network congestion. It can be observed in the graph that the PC/joined node consumes almost same energy in both the approaches. This is because, in both the approaches, the PC/joined nodes use similar duty cycles for transmitting their control packets. A small difference is observed because less number of EB frames are transmitted when the congestion is high in the proposed scheme. But a significant difference in energy consumption can be observed for the pledges when the number of nodes varies. This can be explained by the higher congestion in 6TiSCH-MC when the number of joined nodes increases. Higher congestion forces the pledges to wait for more amount of time to get a valid EB. This increasing amount of time causes more energy consumption as the pledges need to keep their radio active. Whereas in our proposed scheme C2DBI, congestion is reduced by increasing the beacon intervals of the already joined nodes. Therefore, almost equal energy consumption can be seen in our proposed scheme even in the presence of more number of nodes. Fig. 5d shows the energy consumption and joining time of a pledge with respect to the number of nodes for both the schemes using GINA motes.

## 6 Performance Evaluation by Simulation

In this section, the performance of the proposed scheme is analyzed using the Cooja network simulator on Contiki 4.4 operating system [26]. In our evaluation, the nodes are deployed in a fixed square ($6 \times 6$) size grid area where the PC (RPL root) is placed at the top left corner. And the considered grid topology is a multihop topology.

The simulation parameters and their corresponding values are mentioned in Table 3. In 6TiSCH-MC, the considered beacon interval is equal to $4$ $seconds$. In the proposed scheme, we take the same value for minimum beacon interval $I_{eb}^{min}$, and value of the maximum beacon interval $I_{eb}^{max}$ equals to $12$ $seconds$. The CBR calculation interval is considered to be $8$ $seconds$. For DRA [16], the limit of maximum shared slots is set to $8$ $timeslots$ within a slotframe. The slotframe length is varied from 33 to 101 *timeslots* in order to obtain different results with different network configurations. We run each simulation for $120$ $minutes$ using a realistic Multipath Raytracer Medium (MRM) channel model. This model provides various propagation effects such as multi-path, refraction, and diffraction. During simulation, keepalive packet is used to keep the communication between a node and its parent active. When a node does not hear from its parent for $30s$, it initiates a keepalive packet.

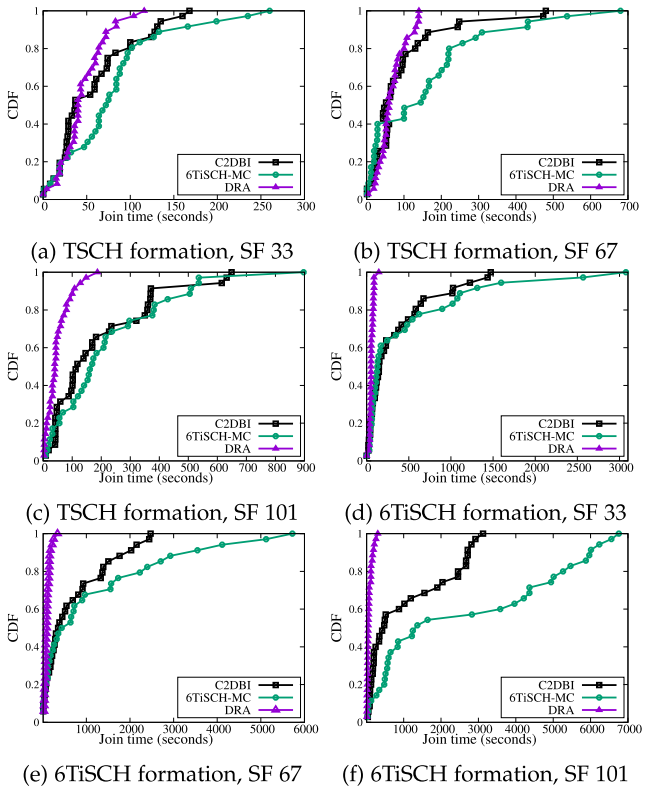



Fig. 6. Simulation results of TSCH and 6TiSCH network formation time using different schemes for different slotframe size (= 33, 67, 101 *timeslots*).

### 6.1 TSCH Network Formation Time

The performance of TSCH network formation is evaluated by means of total time required by the nodes to join in the TSCH network. A pledge scans for EB frame to join the network. So, TSCH network formation time for a pledge is the time required by the pledge to get a valid EB.

Fig. 6a, 6b, and 6c show the cumulative distribution function (CDF) of TSCH formation time with different slotframe lengths such as 33, 67, and 101 *timeslots*. It can be observed in all the three figures that DRA takes less amount of time than our proposed method (C2DBI) and 6TiSCH-MC. The reason for this is the use of multiple shared slots per slotframe by the DRA. DRA dynamically increases the number of shared slots per slotframe when there are more number of control packets available in the network. However, in the C2DBI and 6TiSCH-MC, only one shared slot is used, which delayed the transmission of already generated control packets. Although both the C2DBI and 6TiSCH-MC use single shared slot in each slotframe/multi-slotframe, the C2DBI takes less amount of time

to form a TSCH network compared to that in 6TiSCH-MC because of the reduction of congestion in the shared slots using dynamic beacon interval.

### 6.2 6TiSCH Network Formation Time

To expand the network further, it is necessary for a pledge to join the network fully. That means the pledge should also get the routing information of the network along with EB frame to transmit its own beacon frame. The Fig. 6d, 6e, and 6f show the time required by a newly joined node to send its first EB after completely joining the network. Once again, it is observed that DRA performs better than C2DBI and 6TiSCH-MC. Here also, the reason is same, i.e., multiple shared slots are allocated within a slotframe by the nodes in DRA scheme. On the other hand, dynamic beacon interval reduces congestion in shared slots, and, thus, it performs better than 6TiSCH-MC. Note that C2DBI and 6TiSCH-MC use a single shared slot in a slotframe. Again, it can be observed that, though the pledges join in the TSCH network quickly, but they take longer time to send their beacons. This is because of the delay in getting DIO packets when congestion in shared slot increases due to the limited available resources. Further, the joining time increases with the increasing values of slotframe length. This is because, the frequency of the occurring shared slots decreases with the increasing slotframe length within a time period. Hence, less number of control packets are transmitted because of high congestion in less number of shared slots. However, in DRA, because of the usages of multiple shared slots in a slotframe, the congestion problem affects less. Hence, DRA gives better results with respect to joining time but not for energy consumption.

## 7 Testbed Experiments

The implementation of C2DBI in Contiki-4.4 has been flashed in a real testbed at FIT IoT-LAB [18]. The IoT-Lab M3 nodes are deployed in a $(5 \times 5)$ grid topology and a $(2 \times 12)$ linear topology in Lille and Grenoble locations, respectively, for testing our proposed scheme in real testbed. The used M3 node is an STM32 (ARM Cortex M3) micro-controller based node which supports FreeRTOS, Contiki, and RIOT operating systems. We consider similar configuration as it is used in simulation in terms of beacon interval for all the schemes. Here, the slotframe length is set to $101$ $timeslots$. The received results are plotted in Fig. 7. Fig. 7a and 7c show the time required by a pledge to receive the first EB frame in $(5 \times 5)$ and $(2 \times 12)$ topologies, respectively. Similarly, Fig. 7b and 7d show the time required by a newly joined node to broadcast its first EB frame in $(5 \times 5)$ and $(2 \times 12)$ topologies, respectively. The plotted results show that the DRA performs better than the proposed scheme C2DBI as well as 6TiSCH-MC in terms of both TSCH joining time and 6TiSCH joining time. This is because more number of shared slots are used in a single slotframe by DRA. On the other hand, the proposed scheme and 6TiSCH-MC use single shared slot in a slotframe. In this regard, the proposed scheme performs better than the 6TiSCH-MC because of the reduction of congestion in the shared slot using dynamic beacon interval.

Even though DRA performs better with respect to joining time, very high duty cycle is observed in DRA as shown in Fig. 7e and 7f. Results in the Fig. 7e and 7f show the average duty cycles of all nodes present in the network in their

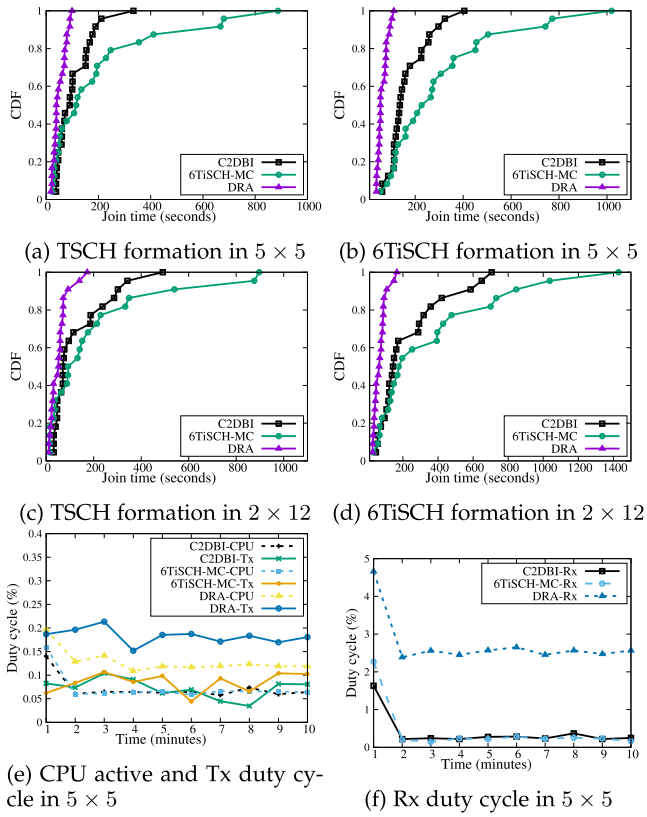



Fig. 7. Testbed results corresponding to different topologies (i.e., $5 \times 5,\ 2 \times 12$) using different formation schemes.

three different states ($CPU_{active}$, $Tx$, $Rx$) during the initial $10$ $minutes$ of network formation. In this initial $10$ $minutes$ of the network formation, most of the nodes join the network. After this time interval, there is no such significant change in average radio duty cycles. Here, the duty cycle of a micro-controller state is the ratio of time spend in the micro-controller state (e.g., Rx) in every $60$ $seconds$. As DRA uses more number of shared slots, the nodes need to keep their radio active for more amount of time than the single shared slot based schemes such as C2DBI and 6TiSCH-MC. Therefore, it results in higher duty cycle. Again, higher duty cycle increases the energy consumption of a node. So, the nodes in DRA consume more energy than that in C2DBI and 6TiSCH-MC. Thus channel condition based dynamic beacon interval significantly improves the network formation performance with respect to joining time and energy consumption.

## 8 Conclusion

In this paper, a channel condition based dynamic beacon interval is proposed for faster association of nodes in 6TiSCH networks. At first, using a Markov chain based analytical model it is shown that the performance of a network degrades with the increased number of joined nodes due to fixed beacon interval in 6TiSCH-MC. To overcome this problem, we propose a channel condition based dynamic beacon interval scheme (C2DBI) which dynamically adjust the beacon interval of a joined node depending on its present channel congestion status measured by the CBR parameter. Theoretical analysis of the proposed scheme is done to show how the performance of network formation is improved with varied beacon interval

compared to the fixed beacon interval scheme. Furthermore, the proposed scheme is also evaluated using simulation in Cooja simulator and using real testbed experiments in FIT IoT-LAB. The received results are compared with the existing benchmark protocols. Though the proposed scheme does not perform better than the DRA with respect to network formation time, but from the energy consumption perspective, C2DBI is better than all the existing works. Further to note that DRA uses multiple shared slots per slotframe whereas C2DBI uses only one. It is observed in our simulation that around 8 percent of the total slotframe of size 101 is used as shared slots in DRA, whereas our proposed method uses only 0.99 percent during network formation. This observation signifies that sensory data packet transmissions are less affected by the proposed method C2DBI compared to that in DRA. From the obtained results, we claim that C2DBI scheme could be more suitable for the networks where energy consumption is an important constraint along with network formation time.

## REFERENCES


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

[2] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

[3] *IEEE Standard for Local and Metropolitan area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4–2011, Sep. 2011.

[4] *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 Standard 802.15.4–2011), Apr. 2012.

[5] X. Vilajosana, T. Watteyne, T. Chang, M. VučiniS. Duquennoy, and P. Thubert, "IETF 6TiSCH: A Tutorial," *IEEE Commun. Surv. Tuts.*, vol. 22, no. 1, pp. 595–615, First Quarter 2020.

[6] M. R. Palattella *et al.*, "Standardized protocol stack for the Internet of (important) Things," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 3, pp. 1389–1406, Mar. 2013.

[7] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," *Internet Eng. Task Force*, RFC 8180, May 2017.

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

[9] M. Vucinic, J. Simon, K. Pister, and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH," *Internet Eng. Task Force*, Dec. 2019.

[10] M. Richardson, "6TiSCH zero-touch secure join protocol," *Internet Eng. Task Force*, Jul. 2019.

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

[12] D. D. 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.*, 2014, pp. 1–6.

[13] 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, 2018.

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

[15] T. P. Duy, T. Dinh, and Y. Kim, "A rapid joining scheme based on fuzzy logic for highly dynamic IEEE 802.15. 4e time-slotted channel hopping networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 8, pp. 1–10, 2016.

[16] 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.

[17] M. Vucinic, T. Watteyne, and X. Vilajosana, "Broadcasting Strategies in 6TiSCH Networks," *Internet Technol. Lett.*, Dec. 2017. [Online]. Available: https://doi.org/10.1002/itl2.15

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

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

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

[21] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," *Internet Eng. Task Force*, RFC 6206, Mar. 2011.

[22] 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. Aerosp. Sci.*, 2017, pp. 1–15.

[23] 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. Sensor Syst.*, 2015, pp. 337–350.

[24] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, "A realistic energy consumption model for TSCH networks," *IEEE Sensors J.*, vol. 14, no. 2, pp. 482–489, Feb. 2014.

[25] M. Khatua and S. Misra, "D2D: Delay-aware distributed dynamic adaptation of contention window in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 322–335, Feb. 2016.

[26] 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.


**Alakesh Kalita** received the BTech degree in computer science and engineering from Assam Don Bosco University, India, in 2012, and the MTech degree in computer science and engineering from Assam University, India, in 2016. He is currently a doctoral researcher with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India, since January 2018. His research interests include Internet of Things, Wireless Sensor Network, and Cloud Computing.

**Manas Khatua** (Member, IEEE) received the PhD degree from the Indian Institute of Technology (IIT) Kharagpur, India, in 2015. He is currently an assistant professor with the Department of Computer Science and Engineering, IIT Guwahati, India, since May 2018. Prior to that, he was an assistant professor in IIT Jodhpur, from 2016 to 2018, and was a postdoctoral research fellow at SUTD, Singapore, from 2015 to 2016. He was associated with Tata Consultancy Services (India) from 2008 to 2010. His research interests include Performance Evaluation of Communication Protocols, Internet of Things, Wireless LANs, Sensor Networks, and Network Security. He is an author of many reputed international journal and conference publications. He is also a member of ACM.