Adaptive Control Packet Broadcasting Scheme for Faster 6TiSCH Network Bootstrapping

Alakesh Kalita and Manas Khatua, *Member, IEEE*

***Abstract*—The IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) Working Group released 6TiSCH minimal configuration (6TiSCH-MC) standard for 6TiSCH network bootstrapping. 6TiSCH-MC allocates only one shared cell per slotframe, known as a minimal cell, to transmit all types of control packets. Our Markov Chain-based probabilistic analysis reveals that few network parameters have a high impact on the performance of 6TiSCH network formation due to increasing congestion in the minimal cell with the increased number of nodes. This work aims to improve the 6TiSCH network formation process by reducing congestion in the minimal cell. For this, the Trickle algorithm, which is used for controlling the rate of routing information-carrying packet transmission is modified so that sufficient routing information can be provided without congesting the minimal cell. To reduce the congestion further, a slotframe window (SW)-based adaptive scheme is proposed by which nodes are restricted to transmit their control packets frequently. The proposed dynamic Trickle algorithm and SW-based scheme are implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed. The experimental results show that both the proposed schemes together improve the joining time and energy consumption of the pledges compared to the state-of-the-art schemes. Additionally, both the proposed schemes provide fair control packet transmission opportunities among the nodes in a network.**

***Index Terms*—Industrial Internet of Things (IIoT), IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH), network formation, RPL, Trickle algorithm.**

I. Introduction

IETF formed IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) Working Group (WG) to provide interoperability between IEEE 802.15.4e [1] standardized time-slotted channel hopping (TSCH) medium access control (MAC) behavior and IETF's upper layer protocols [2], [3]. This WG deals with the network bootstrapping and scheduling of data transmission cell in the 6TiSCH network. The overall 6TiSCH communication architecture, rooted on the top of the IEEE 802.15.4e TSCH link layer, is designed to provide reliable and timed data delivery in large-scale multihop Industrial Internet of Things (IIoT) networks using the existing IPv6 infrastructure [3], [4]. Before proceeding further, frequently used abbreviated words are summarized in Table I.

TABLE I
Frequently Used Abbreviated Words

Abbreviation	Meaning
TSCH	Time Slotted Channel Hopping
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e
EB	Enhanced Beacon
RPL	Routing Protocol for Low-Power & Lossy Networks
DODAG	Destination Oriented Directed Acyclic Graphs
DIO	DODAG Information Object
6TiSCH-MC	6TiSCH Minimal Configuration
SW	Slotframe window

For the effective and resource-efficient network bootstrapping, 6TiSCH minimal configuration (6TiSCH-MC) [5] was published by the WG. 6TiSCH-MC mentioned that a pledge (i.e., new node) should get both the network information-carrying EB frame and routing layer protocol, i.e., RPL [6] information-carrying DIO packet to join a 6TiSCH network. 6TiSCH-MC allocates only one shared cell (also called a minimal cell) per slotframe for the transmission of all types of control packets, such as join request (JRQ) and join response (JRS) frames for secure enrollment [7], [8], DODAG information solicitation (DIS), and keep-alive in addition to EB and DIO packets.

Motivation: A pledge is allowed to transmit its control packets and sensory data packet only after joining a network. Therefore, faster formation of the 6TiSCH network is essential. However, there exist few problems that play the pivot role for this work. First, the works in [9]–[12] proved that joining time and energy consumption of a pledge increase due to the increasing congestion in the minimal cell when more number of nodes join the network. Although the existing works [10], [11], [13] studied the impact of fixed EB rate on congestion in the minimal cell, and thus on network formation, the impact of DIO transmission rate is still unexplored. However, our Markov chain-based analysis reveals that the DIO transmission rate affects 6TiSCH network formation. Second, few existing works [14]–[18] have shown that the default DIO rate-controlling Trickle algorithm [19] cannot provide fair DIO transmission opportunity among the nodes, which in turn results in longer RPL's DODAG construction time. However, their solutions consider the previous version of the IEEE 802.15.4 MAC protocol in which the transmission of control packets was not performed in the minimal cell. Hence, the Trickle algorithm needs to be modified considering the transmission of DIO packet happens in the minimal cell. Third, the EB transmission rate is set by the network administrator at the beginning and remains fixed. However, this fixed rate creates congestion in the minimal cell when the number of nodes increases in the network [10], [11]. Hence, the EB rate needs to be modified adaptively in order to reduce

Manuscript received January 6, 2021; revised April 7, 2021; accepted May 3, 2021. Date of publication May 17, 2021; date of current version December 7, 2021. *(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.2021.3080735

congestion. Again, there is no limitation on the number of transmitted control packets by a node. This in turn results in severe congestion in the minimal cell when all the nodes transmit their control packets more frequently. Hence, there should be some mechanism by which congestion in the minimal cell can be reduced in such scenario.

Contribution: The main goal of this work is to reduce the joining time and energy consumption of the pledges during 6TiSCH network formation by reducing congestion in shared minimal cell. The contribution of this work is twofold. First, a dynamic Trickle algorithm is proposed to reduce congestion in the minimal cell. It is done by skipping the intermediate Trickle states which cause unnecessary DIO transmission during network consistency. To minimize the congestion further, one Trickle parameter (namely, redundancy constant k) is set dynamically instead of using a fixed value all the time. Furthermore, the proposed dynamic Trickle algorithm allows a node to adjust its Trickle listen-only period based on its previous transmission records to achieve a fair DIO transmission opportunity which helps in the faster formation of RPL's DODAG.

In the second scheme, a slotframe window (SW)-based adaptive scheme is proposed to restrict a node on generating and transmitting more control packets within a short period of time. This restriction reduces congestion in the minimal cell and also provides fair control packet transmission opportunities among the nodes. Nodes calculate their SWs depending on the number of nodes and generated DIO packets in their surroundings. So, it does not require additional signaling overhead to compute the SW. In brief, the contributions are as follows.

1) A Markov chain-based analytical model is proposed for analyzing the effect of Trickle parameters on 6TiSCH network formation.
2) A dynamic Trickle algorithm is proposed to reduce congestion in the minimal cell and to provide fair DIO transmission opportunity to all the nodes.
3) An SW-based adaptive scheme is proposed to restrict the nodes from transmitting several control packets within a short period of time in order to reduce congestion in minimal cell.
4) The proposed schemes are implemented on Contiki-NG OS and evaluated using a real FIT IoT-LAB testbed.

Article Organization: The remainder of this article is organized as follows. Section II summarizes the existing works related to the 6TiSCH network formation and Trickle algorithm. Section III presents an analytical model to show the impact of various Trickle parameters on 6TiSCH network formation. The proposed schemes are briefly described in Section IV, and in Section V, their experimental evaluation is provided. Finally, the conclusion of this work is drawn in Section VI.

II. Related Works

The formation of the 6TiSCH network gains attention among the researchers because of its channel hopping feature and resource-constrained nodes. Nodes change their physical transmission channel in every communication timeslot irrespective of the type (i.e., data or control) of a packet. Initially, the pledges do not know about exact the physical channel in which EB transmission happens. So, they need to keep their radios active all the time to receive transmitted EB frames which cause high energy consumption. Although, few works, such as [20] and [21], increased the transmission rate of EB to maximize the EB reception probability of the pledges, but their solutions are limited to single-hop node association and consume more energy. Later on, IETF published 6TiSCH-MC [5] for network bootstrapping in which it is mentioned that only one shared cell per slotframe should be used to transmit all types of control packets by all the nodes present in a network. However, Vallatti *et al.* [9] have proved that the resource allocated by 6TiSCH-MC for bootstrapping is not sufficient to transmit all the generated control packets. Therefore, the works [9] and [12] increased the number of shared cells per slotframe for quick transmission of control packets. As a result, both works significantly improve the formation time. However, the work [9] consumes more energy and a lot of cells remain unused in each slotframe. Further, it affects in data transmission schedule as many slots are assigned as shared slots in a slotframe. On the other hand, although the work [12] does not increase the duty cycle of a node, and so energy consumption, collision is still possible in congested networks. Later, Vučinić *et al.* [13] mentioned that EB transmission probability should be 0.1 per slotframe irrespective of the number of nodes present in a network from their simulation analysis. However, a pledge might need to wait for more time to get an EB if it has only a few joined neighbor nodes. Kalita and Khatua [10], [11] varied the EB generation interval based on the shared cell congestion status to reduce congestion in a shared cell. However, the authors did not consider the impact of the DIO rate on channel congestion. The same authors highlighted the problem of EB's highest priority and insufficient transmission of DIO packet during network formation in their other works [22]. As a solution, the authors dynamically change the priority of a control packet and the behavior of the Trickle algorithm depending on network requirement. However, the authors did not study about the congestion in a shared cell due to transmission of both EB and DIO packets together.

Many works [14]–[18] mentioned that the default Trickle algorithm creates unfairness among the nodes during DIO transmission which results in longer RPL's DODAG construction time. It happens due to the use of the listen-only period for the suppression mechanism of the Trickle algorithm. In every Trickle interval, a node randomly selects one time instant between the half of the interval to the end of the interval. The node listens for consistent DIO packets from the starting of the interval to the selected random time. If it receives consistent DIOs more than a threshold called redundancy constant k, then it suppresses its own DIO; otherwise, transmits. Fig. 1 shows few examples of the Trickle operation considering $k = 1$ where nodes ② and ③ always suppress their DIOs because of the improper selection of the random time. This results in longer DODAG construction time. This fairness issue also causes quick energy draining for the frequently transmitting node ①.

The works [14]–[18] tried to solve these problems in various ways, such as by varying the listen-only period depending on previous suppression and transmission counts of DIO, neighbor nodes. However, their solutions consider the previous

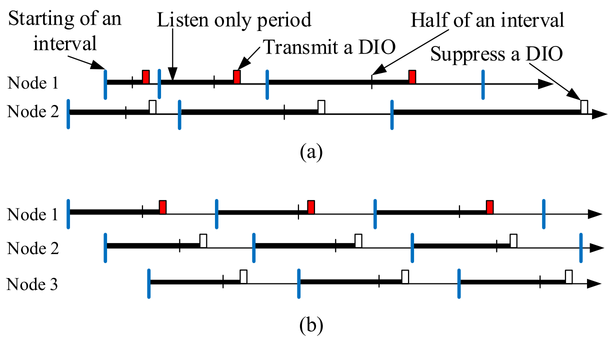

Fig. 1. Example of Trickle operation in different scenarios. (a) Nodes 1 and 2 are in different Trickle states and starting points. (b) All nodes are in same Trickle states but different starting points.

TABLE II
LIST OF FREQUENTLY USED NOTATIONS

Variable	Meaning
I_{min}	Initial/minimum DIO generation interval
I_{max}	Maximum DIO generation interval
I	Current Trickle (or DIO generation) interval
c	Counter for received consistent DIOs
k	Redundancy constant
j	Current Trickle state number
N_{s_cell}	Number of shared cell in a DIO interval
N_{nbr}	Number of neighboring nodes
L	Slotframe length in timeslots
T	Timeslot length in milliseconds
Tr	Number of transmitted DIOs
S	Number of suppressed DIOs
N_D	Number of Trickle states

version of IEEE 802.15.4, where the transmission of DIO packets was not performed in a shared cell. A node may need to wait for a longer time due to the congestion in a shared cell. Hence, the performance obtained by these works may not be the same in the 6TiSCH network as sufficient bandwidth was used in their solutions. On the other hand, Kalita and Khatua [22] did not consider the listen-only problem in their work. Hence, the Trickle algorithm should be modified so that it can provide fairness among the nodes considering DIO transmission happens only in a shared cell.

III. ANALYSIS OF TRICKLE ALGORITHM

In this section, the impact of Trickle parameters on 6TiSCH network formation is described briefly. To the best of our knowledge, the Trickle algorithm is studied on the 6TiSCH network for the first time. Although the transmission rate of the DIO packet is governed by the Trickle algorithm [19] but it is worth mentioning that the DIO transmission rate depends on few Trickle parameters, such as DIO minimum interval $I_{\min}$ and redundancy constant k. These parameters are set by the network administrator at the beginning and remain unchanged. Here, $I_{\min}$ is the smallest DIO generation interval and k is the fixed integer used by the suppression mechanism of the Trickle algorithm. The smaller value of $I_{\min}$ and bigger value of k allow more DIO transmission in a network, and *vice versa* [19]. Fig. 2 shows the flowchart of the Trickle algorithm which is explained briefly in the next section.

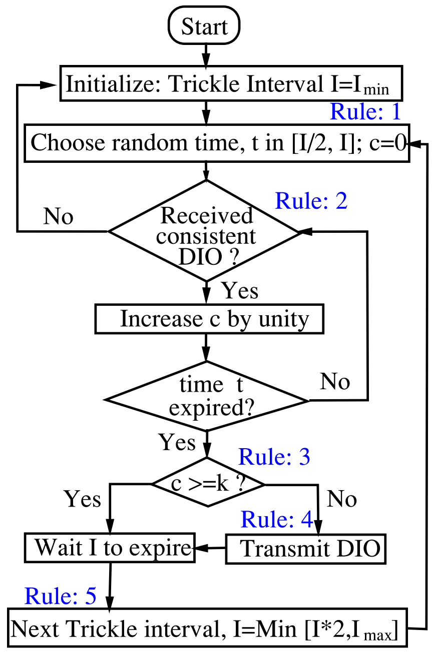

Fig. 2. Flowchart of the Trickle algorithm.

A. Modeling the Pledge Joining Process

Let us assume a 2-D area of size A, in which N nodes are deployed randomly. Let the coverage radius of each node is r. The other used variables with their corresponding meaning are explained in Table II. All nodes run the RPL protocol and thus the Trickle algorithm in it. This setup is used to calculate the average joining time (AJT) of the deployed pledges. The AJT of a pledge denotes the time taken by the pledge to receive one valid EB frame followed by its secure enrollment (i.e., exchanging of JRQ and JRS frames) and reception of a valid DIO packet. Therefore, AJT mainly depends on the transmission probabilities of EB and DIO packets in a shared cell. The trickle algorithm doubles the DIO generation interval after each interval, until maximum Trickle interval $I_{\max}$ is reached (Fig. 2, Rule: 5). The value of the current interval I reset to $I_{\min}$ when network inconsistency occurs (Fig. 2, Rule: 2). Therefore, the next Trickle state depends on the current Trickle state and network condition. Hence, the average probability of being in a Trickle state can be calculated using a semi-Markov model as shown in Fig. 3(a). Note that the DIOs are generated irregularly, i.e., the DIO generation rate does not follow any constant average rate, and the next DIO generating state depends on how long the current state has lasted. Therefore, the semi-Markov model is used to find the total time spend by a node in each Trickle state, i.e., sojourn time. Fig. 3(a) shows how a node changes its Trickle state starting from the first state $I_{\min}$ to the last state $I_{\max}$. Each Trickle state I_j represent $2^j \times I_{\min}$ duration of Trickle interval. There are maximum N_D Trickle states. The state transition probabilities P_r and $(1-P_r)$ are shown in the figure where P_r is the Trickle reset probability. Now, the probability of generating a DIO packet in a shared cell being in Trickle State I_j can be calculated as

$$P_{\text{dio}}^{j} = \sum_{j=1}^{N_D} \binom{N_D}{j} x(1-x)^{(N_D-1-j)} \min\left(1, \frac{L}{2^{j-2} I_{\min}}\right) \quad (1)$$

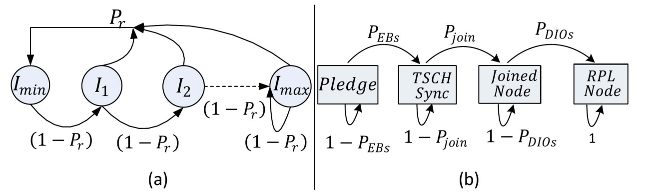

Fig. 3. (a) Semi-Markov model for Trickle states. (b) Markov chain model of node joining time [11].

where $x = ([P_r(1-P_r)^{j-1}2^{j-1}]/[\sum_{j=1}^{N_D} P_r(1-P_r)^{j-1}2^{j-1}])$ implies the probability of being in the Trickle state I_j following the sojourn time of the semi-Markov process. The term $\min(1, [L/(2^{j-2}I_{\min})])$ denotes the probability of generating a DIO in a slotframe within the interval $[2^{j-2}I_{\min}, 2^{j-1}I_{\min}]$ for the I_j Trickle state, where L is the slotframe length. This is because, a node generates a DIO at randomly selected time t in between $[I/2, I]$ when the interval length is I (Fig. 2, Rule: 1). Now, the transmission of a generated DIO packet follows the suppression mechanism of the Trickle algorithm. A node suppresses its generated DIO when it receives greater than or equal to k number of consistent DIO packets till a randomly selected time t of an interval (Fig. 2, Rule: 3, c denotes the number of received consistent DIOs). Otherwise, the node transmits its generated DIO packet (Fig. 2, Rule: 4). Therefore, when all the neighbor joined nodes (say jn) of a node generate and transmit their DIOs, the node can surely transmit its own DIO if $jn < k$. It is because less than k number of DIOs would be transmitted by its surrounding nodes. On the other hand, when $jn \geq k$, the node can be able to transmit only when it generates DIO within the first k transmissions of DIOs. Thus, DIO transmission probability in a shared cell P_{dioTx} depends on its generation probability P^j_{dio} and suppression conditions as follows:

$$P_{\text{dioTx}} = P^j_{\text{dio}} P(jn < k) + P^j_{\text{dio}} P(jn \geq k) P(f) \quad (2)$$

where $P(jn < k)$ and $P(jn \geq k)$ denote the probabilities of satisfying the $jn < k$ and $jn \geq k$ conditions, respectively, and $P(f)$ denotes the probability that a node generates its DIO within the first k transmissions of DIOs. These probabilities can be calculated as

$$P(jn < k) = (k-1)/M, \;\; P(jn \geq k) = (M-k+1)/M$$
$$P(f) = k/(jn+1) \quad (3)$$

where $M = ([N/A]\pi r^2)$ is the total number of neighbors of a node. Both the pledges and joined nodes are included in those M neighbors. We assume that initially all M nodes are pledges. They join the network one by one. So, at any instant, the probability of having any number of joined nodes equals to $(1/M)$ considering uniform probability distribution. Therefore, $(k-1)/M$ denotes the probability that at any instant of time at most $(k-1)$ joined nodes are available. Similarly, we compute the other probabilities.

We use the Markov chain model [shown in Fig. 3(b)] described in [11] to compute the AJT of the pledges as follows:

$$\text{AJT} = \text{PJT} + \text{ASF} \times L \quad (4)$$

where PJT is the AJT of the parent node of the pledge in multihop network, and ASF is the average number of slotframes that is required for a pledge to join the network, which is calculated as follows:

$$\text{ASF} = \frac{1}{P_{\text{EB}_S}} + \left(\frac{1}{P_{\text{JRQ}_S}} + \frac{1}{P_{\text{JRS}_S}}\right) + \frac{1}{P_{\text{DIO}_S}} \quad (5)$$

where P_{EB_S}, P_{DIO_S}, P_{JRQ_S}, and P_{JRS_S} are the probabilities of successfully transmitting an EB, DIO, JRQ, and JRS frames/packets in a shared cell, respectively. The values of these probabilities can be calculated using the value of P_{dioTx} [from (2)] and following the method as described in [11]. For the brevity of this article, that method is not discussed here. However, interested readers can follow the work in [11] for getting the detailed procedure.

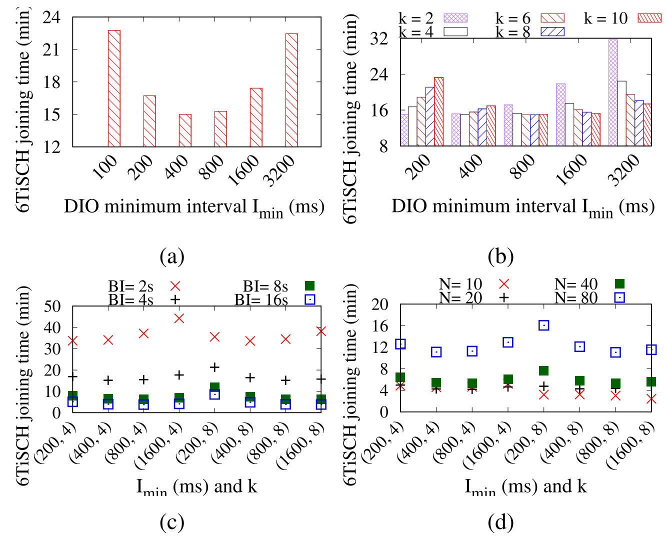

Fig. 4. Analytical results on joining time under varied (a) $I_{\min}$, (b) $I_{\min}$, k, (c) $I_{\min}$, k, BI, and (d) $I_{\min}$, k, N.

TABLE III
LIST OF USED PARAMETERS AND THEIR VALUES

Parameter	Value
Number of channels, N_C	16
Packet loss probability, P_{loss}	0.2
Beacon interval, BI	$4sec$
Timeslot duration, T	$10ms$
Slotframe length, L	101 $timeslots$
Maximum Trickle states, N_D	8
Trickle reset probability, P_r	0.2

B. Analytical Results and Discussion

For performing theoretical analysis, N nodes are deployed at an area of $A = 100 \times 10$ m^2, where the radio range of each node is $r = 10$ m. The default values of other variables are mentioned in Table III. Fig. 4 shows the AJT against the varied $I_{\min}$, k, BI, and N.

Fig. 4(a) shows the joining time for different values of $I_{\min}$ with $N = 100$ and $k = 4$. When the values of $I_{\min}$ are less, the frequent transmission of DIO packet congests the shared cell which results in higher joining time of the pledges. On the other hand, using a higher value of $I_{\min}$, pledges need to wait a longer time for receiving the DIO packet, hence, their joining times are high again. It indicates that neither higher nor lower values of $I_{\min}$ are optimal with respect to the performance of network joining time. In Fig. 4(b), along with $I_{\min}$, k is also varied. Results show that some correlation exists between these two parameters $I_{\min}$ and k. When the values of $I_{\min}$ are either very low or very high, the values of k need to be very low and very high, respectively, to achieve less joining time.

In Fig. 4(c), the effect of EB generation rate combining with $I_{\min}$ and k are shown. The frequent transmission of the EB frame severely congests the shared cell. Again, because of the highest priority of the EB frame, other control packets are replaced by the EB from the transmission buffer. Hence, the joining time of the pledges increase. So, a longer EB generation interval, helps in reducing congestion in the shared cell, and thus, yields lesser joining time irrespective of the values of $I_{\min}$ and k as shown in the figure. It is further observed that congestion in the shared cell increases with the increasing node densities, and so the joining time. However, the value of increment varies with the values of $I_{\min}$ and k as shown in Fig. 4(d).

Analytical results show that the same values of $I_{\min}$ and k show different results under different network conditions, such as network size and beacon generation interval. It is mainly due to the congestion in a shared cell. So, it can be concluded that the Trickle algorithm should dynamically change its values depending on the network condition, rather than being fixed, in order to reduce congestion in a shared cell.

IV. Proposed Scheme

In this section, the proposed dynamic Trickle algorithm and the SW-based adaptive control packet transmission scheme are described briefly. The variables used in the proposed schemes are already explained in Table II.

A. Dynamic Trickle Algorithm

The above-mentioned analytical results show that the improper selection of Trickle parameters, such as $I_{\min}$ and k, results in more DIO transmission in the network, which increases the congestion in the shared cell, and so 6TiSCH formation time. When the Trickle algorithm is reset, it increases the transmission of DIO packets. In Fig. 2, Rule: 2 shows one of such Trickle resetting conditions that is when a joined node receives an inconsistent DIO packet. Note that there are a few more Trickle resetting conditions, such as reception of multicast DIS, resetting of DODAG, etc. A single multicast DIS request forces all the recipient nodes to reset their Trickle algorithms. This results in burst transmission of DIO packets in the network for a long time which creates severe congestion in a shared cell. So, in order to reduce such burst transmission, the proposed scheme allows the receiving node to generate only a single DIO packet in the consistent networks. This is done by allowing the receiving node to directly jump to the Trickle state where it has received the multicast DIS after generating one DIO packet. Hence, it reduces the number of transmitted DIOs by skipping the intermediate Trickle state(s). Thus, this modification reduces congestion as well as energy consumption of the nodes. This modification is done at Fig. 2, Rule: 2 and the steps are mentioned in Algorithm 1.

To reduce the transmission of unnecessary DIO packet further, the redundancy constant k is set based on the number of neighbors and current Trickle state instead of using a fixed value all the time. All the nodes are permitted to transmit their DIOs at the initial Trickle state as a node would be in that state only when there is a strong requirement of DIO in

Algorithm 1: Skipping Intermediate Trickle State

```
if Received consistent multicast DIS then
  | if I is not I_min then
  |   | flag = j /*save the Trickle state*/
  |   | Reset I, j, S and T_r to their default values
  | end
else if flag then
  | I = 2^(flag-1) I_min /*jump to saved state*/
  | Reset flag
end
```

Algorithm 2: Dynamic Redundancy Constant k

```
if I=I_min then
  | k = min(N_nbr + 1, 10)/*all nodes are allowed*/
else if j > I_min && j ≤ N_D/2 then
  | k = min(⌈(N_nbr + 1)/2⌉, 10)
else
  | k = min(N_nbr + 1, 10)
end
```

the network. However, after this initial state to the half of the total Trickle states, i.e., $N_D/2$, k is set to half of the neighbors so that frequent transmission of DIO can be reduced. And, when the nodes generate their DIOs less frequently, i.e., their current Trickle state is greater than $N_D/2$, again k is set to neighbor node count. However, the maximum value of k is set to 10 (specified by RPL). The k is calculated before the condition at Fig. 2, Rule: 3 and Algorithm 2 describes the steps.

Now, to deal with the fairness issue of the Trickle algorithm, the proposed dynamic Trickle algorithm allows a node to calculate its DIO transmitting shared cell based on its previous transmission record and number of neighbors. When a node suppresses its DIOs continuously in the previous intervals, it gets more opportunity to transmit in the next interval. This is done by selecting the random time t within a short interval (started from the beginning of the interval) depending on the number of neighbors and total suppression count, i.e., $t = [0, \lceil[(N_{s\_\text{cell}})/2]\rceil - ([(N_{s\_\text{cell}}/2)/(N_{nbr}+1)] \times S)]$. When the listen-only period of a node is small, less number of consistent DIOs will be received by the node and so it can transmit its generated DIO immediately (at the beginning of the interval). Furthermore, the node does not need to wait more than half of the interval like the default Trickle algorithm. On the other hand, a frequent DIO transmitting node calculates its listen-only period depending on the number of already transmitted DIOs, i.e., $t = [\lceil([N_{s\_\text{cell}}]/2)\rceil + ([(N_{nbr}+1)/(N_{s\_\text{cell}}/2)] \times Tr), N_{s\_\text{cell}}]$. This allows the node to choose its transmitting cell from the end of the interval. Thus, the proposed scheme maintains fairness among the nodes. This modification is done at Fig. 2, Rule: 1 and steps are mentioned in Algorithm 3.

B. Slotframe Window-Based Adaptive Transmission Scheme

Apart from the dynamic Trickle algorithm, this article also proposed one SW-based adaptive control packet transmission scheme to reduce congestion in a shared cell. This is done by restricting the joined nodes to transmit their control packets more frequently when the number of neighbor nodes and generated control packets are more in the network. To understand the proposed scheme, let us assume I_{eb} is the fixed EB generation rate for all the nodes. So, the probability of generating

Algorithm 3: Providing Fair DIO Transmission Opportunity

Calculate: N_{s_cell}= $\left\lfloor \frac{2^{m-1} I_{\min}}{L \times T} \right\rfloor$; `/* count the number of shared cells present in a Trickle state` m `*/`
if I = $I_{\min}$ *OR S != 0* **then**
  $t = \left[0, \left\lceil \frac{N_{s\_cell}}{2} \right\rceil - (\frac{N_{s\_cell}/2}{N_{nbr}+1} \times S)\right]$
else if *Tr* != 0 **then**
  $t = \left[\left\lceil \frac{N_{s\_cell}}{2} \right\rceil + (\frac{N_{nbr}+1}{N_{s\_cell}/2} \times Tr), N_{s\_cell}\right]$
end

an EB per shared cell by a node is $P_{\text{eb}} = (L \times T)/I_{\text{eb}}$. Now, if N_{nbr} is the number of neighbors, then $P_{\text{eb}} \times (N_{\text{nbr}} + 1) < 1$; otherwise, when $P_{\text{eb}} \times (N_{\text{nbr}} + 1) > 1$ then at least one node will not be able to transmit its EB in every slotframe by *Pigeon hole theory*. Therefore, the proposed scheme allows a node to generate its EB depending on its number of neighbors rather than using a fixed rate as follows:

$$I_{\text{eb}} \geq \text{Min}\left\{I_{\text{eb}}^{\max}, \text{Max}\left\{I_{\text{eb}}^{\min}, L \times T \times (N_{\text{nbr}} + 1)\right\}\right\} \quad (6)$$

where $I_{\text{eb}}^{\min}$ and $I_{\text{eb}}^{\max}$ are the minimum and maximum beacon generation interval. Here, a node waits at least $(N_{\text{nbr}}+1)$ shared cells before transmitting its next EB frame. However, the same node also has other sporadic control packets, such as DIO, DIS, and keep-alive to transmit. For this, the new term SW is calculated as follows:

$$\text{SW} = L \times T \times (N_{\text{nbr}} + 1) + \lceil x \ \% \ ofL \times T(N_{\text{nbr}} + 1)\rceil \quad (7)$$

where the value of x defines the percentage of extra shared cells allocated for other control packets. We consider two different values of x. By default, the value of x is set to 50 but when a multicast DIS packet is broadcasted, all the receiving nodes will generate one new DIO. Therefore, in such cases, the value of x is set to 100, so that additional $(N_{\text{nbr}} + 1)$ number of shared cells can be used for the transmission of newly generated DIOs.

After calculating the value of SW, a node is restricted to broadcast more than two control packets (i.e., one EB frame and one DIO packet) within the SW amount of time. Therefore, when there is a situation like more number of control packets need to be transmitted, a node is restricted to transmit its control packets more frequently. This reduces congestion in the shared cell and also provides equal opportunity to all the nodes to transmit their control packets.

However, we observe that the nodes who has more neighbors get less opportunity to transmit their packets compared to the nodes who has less neighbors. Fig. 5(a) depicts one such type of scenario. When the nodes calculate their SW following (7) and $x = 0$, node Ⓐ gets less opportunity to transmit its packets compared to the nodes Ⓑ, Ⓒ, and Ⓓ. Furthermore, collision is possible and the common neighbor node Ⓟ does not get any free shared cell to transmit. To deal with this problem, the final value of SW is calculated as follows:

$$\text{SW}_f = \text{Max}\left\{\text{SW}, \ \text{SW}_i^{\max}\right\} \quad (8)$$

where SW is calculated by a node itself (7) and $\text{SW}_i^{\max}$ is the maximum value of SW among its neighboring nodes. The improvement of this step is shown in Fig. 5(b) where all nodes

Algorithm 4: SW-Based Adaptive Transmission Scheme

Initialize: SW_f = $I_{\text{eb}}^{\min}$, Transmitted packet *tp*=0, *x*=50
if *Received consistent multicast DIS* **then**
  if SW_f *time is over* **then**
    Calculate: I_{eb} using Equation 6; $tp = 0$; $x = 50$;
    Calculate: *SW* using Equation 7 & 8
  end
end
if *the current* $Link_{type}$ *is Shared* **then**
  if *tp* < *2 && channel is free* **then**
    Transmit a packet from broadcasting buffer; tp++;
  else if *received multicast DIS* **then**
    x = 100 `/*to increase SW length*/`
  end
end

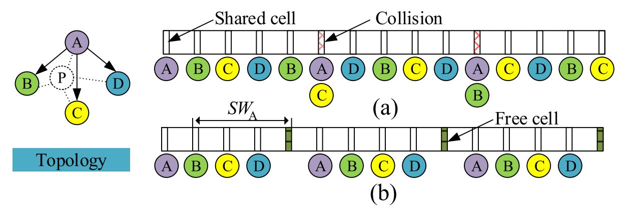

Fig. 5. Transmission of control packet by the nodes following (a) 6TiSCH-MC and (b) proposed SW-based scheme.

TABLE IV
EXPERIMENTAL SETTINGS

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

[including Ⓟ] can transmit their packets without any collision. In the figure, SW_A denotes the calculated value of SW by node Ⓐ. The overall steps of this SW-based scheme are mentioned in Algorithm 4.

V. PERFORMANCE EVALUATION

The proposed schemes are implemented on Contiki-NG OS [23] and evaluated using FIT IoT-LAB [24] testbed experiments. The performance of both the proposed schemes together is compared with two benchmark schemes, i.e., 6TiSCH-MC [5] and BS [13]. Every node running the proposed schemes includes its calculated SW in the information element (IE) of the EB frame. Thus, all the nodes exchange their calculated SW without any signaling overhead. The testbed experimental settings are mentioned in Table IV and two different topologies are used for evaluation in which the first one is a Tree topology (Fig. 6) and the other one is 6 × 6 Grid topology. The node density of the Grid topology is more than the Tree topology.

For evaluation, mainly four metrics are considered. The first metric is *TSCH joining time*. It denotes the time when a pledge receives its first valid EB from an already joined node. Before TSCH joining time, the pledges keep on scanning for the EB frame by activating their radios all the time,

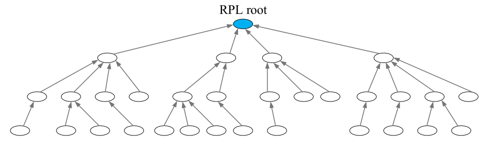

Fig. 6. Tree topology used in testbed experiments.

which causes maximum energy consumption. However, after receiving an EB, the pledges come to know about the location of the shared cell [`SlotOffset, ChannelOffset`] and then, keep their radios active only in the shared cell, which saves energy. The second metric 6TiSCH *joining time* denotes the time when a TSCH joined node receives its first valid DIO packet, i.e., TSCH joined node joins the topology created by routing protocol RPL. Now, the 6TiSCH joined node can transmit its control packets for further expansion of the 6TiSCH networks.

The `motes` used in IoT network constrained in terms of processing capacity, memory, and energy. As mentioned above, pledges consume huge energy before their TSCH joining. Therefore, *energy consumption* by pledge during its network joining is considered as our third metric for evaluation.

Apart from this, to prove that the proposed dynamic Trickle algorithm gives equal opportunity to all the nodes to transmit their DIOs (i.e., provides fairness), *Jain's fairness index* [25] is used, which is defined as follows:

$$F(x) = \frac{\left(\sum_{i=1}^{z} x_i\right)^2}{z \sum_{i=1}^{z} {x_i}^2}, \quad x_i \geq 0 \tag{9}$$

where z denotes the number of users (here, `motes`) contending for the resource (here, shared cell) and the ith user receives an allocation x_i. The value of $F(x)$ becomes 1 when the resource is equally distributed among the z users.

The received results from the testbed experiments are shown using 95% confidence interval in Fig. 7. Fig. 7(a) and (c) shows the TSCH joining time and Fig. 7(b) and (d) shows the 6TiSCH joining time of the nodes in Tree and Grid topologies, respectively. The results are shown by varying the DIO minimum interval $I_{\min}$ and redundancy constant k in the 6TiSCH-MC and BS schemes. As the proposed scheme dynamically sets the value of k, so results are obtained only varying the value of $I_{\min}$. From the received result, it can be seen that the proposed scheme outperforms both 6TiSCH-MC and BS under all the configured settings. It is mainly because of the congestion reduction in shared cell and providing equal transmission opportunity to all the nodes by both the proposed schemes. The effect of the dynamic Trickle algorithm on the average number of transmitted and suppressed DIO packets is shown in Table V for both the topologies. By skipping the intermediate Trickle states and dynamically setting k, the number of transmitted DIOs has been reduced. Furthermore, the SW-based scheme also helps in reducing this count. Similarly, Algorithm 3 helps in providing fairness in DIO transmission among the nodes, i.e., reduces suppression count. On the other hand, 6TiSCH-MC suffers from heavy congestion in a shared cell in Grid topology as the nodes are densely deployed, and so experience more joining time and energy consumption. Although the suppression mechanism is used to save node's energy and network bandwidth, the suppression of more number of DIOs affects on routing operation as shown by [14].

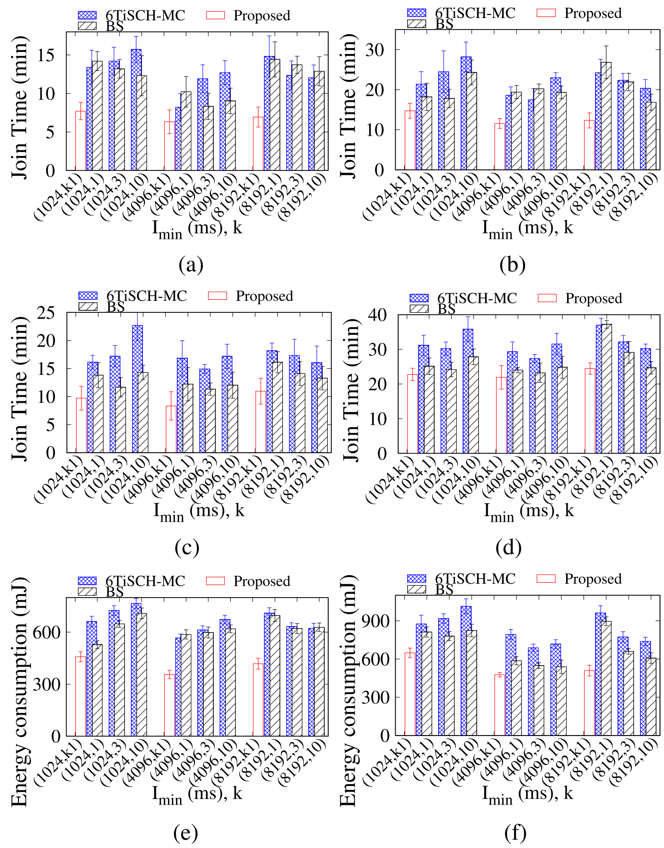

Fig. 7. Testbed experimental results with respect to I_{min}, and k or $k1$, where k and $k1$ denote static and dynamic redundancy constants, respectively. (a) TSCH join time in Tree. (b) 6TiSCH join time in Tree. (c) TSCH join time in Grid. (d) 6TiSCH join time in Grid. (e) Energy consumption in Tree. (f) Energy consumption in Grid.

So, the proposed scheme does not affect routing operation as it suppresses less number of DIO transmission.

Nodes can save their energy by quickly joining the TSCH network and transmitting less number of control packets. As both the proposed schemes together reduce the joining time as well as the number of transmitted control packets, the energy consumption by the nodes also gets reduced using the proposed schemes together compared to the benchmark schemes. The energy consumption results are shown in Fig. 7(e) and (f) for both the experimental topologies. From the results, it can be claimed that the proposed schemes together improve the lifetime of the network.

Now, coming to Jain's fairness index, when all the nodes transmit their DIOs without any suppression, $F(x)$ becomes 1. However, the transmission of more number of DIOs increases the load in the shared cell and so the congestion. Therefore, the results on fairness are shown along with transmission load in the shared cell in Table VI. The transmission load in the shared cell is calculated by taking the ratio of the number of transmitted DIOs to the number of occurring shared cells during the entire experimental time. The results clearly show that the proposed schemes improve the performance compared to the benchmark schemes. Though in some cases, 6TiSCH-MC and BS give better fairness (specifically when, $k = 10$), their transmission load is also more. But, the proposed schemes are able to maintain the balance between the fairness and transmission load.

TABLE V
RESULTS ON AVERAGE NUMBER OF TRANSMITTED AND SUPPRESSED DIOS SHOWN AS TUPLE (TRANSMITTED DIOS AND SUPPRESSED DIOS) FOR BOTH THE TOPOLOGIES

Topo-logy	I_{min}	6TiSCH-MC			BS			Proposed
	(ms)	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$	Dynamic
Tree	1024	521, 82	568, 15	467, 0	469, 94	483, 18	599, 0	169, 1
	4096	397, 67	403, 9	490, 0	488, 123	483, 15	468, 0	152, 4
	8192	335, 57	379, 4	443, 0	568, 79	534, 2	423, 0	158, 3
Grid	1024	658, 102	692, 22	513, 0	602, 137	587, 94	667, 0	207, 3
	4096	479, 76	468, 17	459, 0	529, 54	561, 14	580, 0	187, 2
	8132	435, 67	462, 12	507, 0	524, 45	471, 9	453, 0	193, 3

TABLE VI
RESULTS ON FAIRNESS AND TRANSMISSION LOAD SHOWN AS TUPLE (FAIRNESS AND TRANSMISSION LOAD) FOR BOTH THE TOPOLOGIES

Topo-logy	I_{min}	6TiSCH-MC			BS			Proposed
	(ms)	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$	Dynamic
Tree	1024	.87, .14	.98, .15	1, .13	.83, .13	.96, .14	1, .17	.99, .05
	4096	.86, .11	.98, .11	1, .13	.80, .13	.97, .13	1, .13	.97, .04
	8192	.86, .09	.99, .1	1, .13	.88, .15	.98, .15	1, .12	.98, .04
Grid	1024	.87, .18	.97, .19	1, .14	.81, .17	.86, .16	1, .19	.99, .06
	4096	.86, .13	.96, .13	1, .13	.90, .15	.98, .16	1, .16	.99, .05
	8192	.87, .12	.97, .13	1, .14	.92, .15	.98, .13	1, .13	.98, .05

VI. CONCLUSION

It is observed that the existing works did not consider the effect of DIO transmission rate on 6TiSCH network formation. Initially, our Markov chain-based probabilistic analysis reveals that the improper selection of various Tickle parameter values can increase the 6TiSCH network formation time due to the increased congestion in a shared cell. Therefore, a dynamic Trickle algorithm is proposed to reduce this congestion. The proposed scheme skips some intermediate DIO generation Trickle states during network consistency and varies one Trickle parameter, i.e., redundancy constant dynamically. Further, it provides fair DIO transmission opportunity to all the nodes by varying the Trickle listen-only period of the nodes depending on their previous DIO transmission history. Apart from this, the SW-based adaptive control packet transmission scheme is proposed to further reduce the congestion in a shared cell. This scheme restricts the nodes to transmit their control packets more frequently. Both the proposed schemes are implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed. Testbed results show that both the proposed schemes together improve the joining time and energy consumption of the nodes significantly compared to the state-of-the-art schemes while providing fair DIO transmission opportunity to all the nodes.

REFERENCES

[1] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2015, Apr. 2016.

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

[3] 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 Quat., 2020.

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

[5] 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.

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

[7] M. Richardson, "6TiSCH zero-touch secure join protocol," Internet Eng. Task Force, Internet-Draft draft-ietf-6tisch-dtsecurity-zerotouch-join-04, Jul. 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-6tisch-dtsecurity-zerotouch-join/04/bibtex/

[8] M. Vucinic, J. Simon, K. Pister, and M. Richardson, "Constrained join protocol (CoJP) for 6TiSCH," Internet Eng. Task Force, Internet-Draft draft-ietf-6tisch-minimal-security-15, Dec. 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-6tisch-minimal-security/15/bibtex/

[9] 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.

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

[11] 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

[12] A. Kalita and M. Khatua, "Autonomous allocation and scheduling of minimal cell in 6TiSCH network," *IEEE Internet Things J.*, early access, Feb. 25, 2021. [Online]. Available: https://doi.org/10.1109/JIOT.2021.3062115

[13] M. Vučinić, T. Watteyne, and X. Vilajosana, "Broadcasting strategies in 6TiSCH networks," *Internet Technol. Lett.*, vol. 1, p. e15, Jan./Feb. 2018. [Online]. Available: https://doi.org/10.1002/itl2.15

[14] C. Vallati and E. Mingozzi, "Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol," in *Proc. Sustain. Internet ICT Sustain.*, Palermo, Italy, 2013, pp. 1–9.

[15] M. Vučinić, M. Król B. Jonglez, T. Coladon, and B. Tourancheau, "Trickle-D: High fairness and low transmission load with dynamic redundancy," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1477–1488, Oct. 2017.

[16] M. B. Yassein, S. Aljawarneh, and E. Masa'deh, "A new elastic trickle timer algorithm for Internet of Things," *J. Netw. Comput. Appl.*, vol. 89, pp. 38–47, Jul. 2017.

[17] B. Ghaleb, A. Y. Al-Dubai, E. Ekonomou, I. Romdhani, Y. Nasser, and A. Boukerche, "A novel adaptive and efficient routing update scheme for low-power lossy networks in IoT," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5177–5189, Dec. 2018.

[18] T. M. M. Meyfroyt, M. Stolikj, and J. J. Lukkien, "Adaptive broadcast suppression for trickle-based protocols," in *Proc. IEEE 16th Int. Symp. World Wireless Mobile Multimedia Netw.*, Boston, MA, USA, 2015, pp. 1–9.

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

[20] 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.

[21] 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. World Wireless Mobile Multimedia Netw.*, Coimbra, Portugal, Jun. 2016, pp. 1–9.

[22] A. Kalita and M. Khatua, "Opportunistic transmission of control packets for faster formation of 6TiSCH network," *ACM Trans. Internet Things*, vol. 2, no. 1, p. 5, Jan. 2021.

[23] 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.*, Tampa, FL, USA, 2004, pp. 455–462.

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

[25] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Eastern Res. Lab., Lexington, MA, USA and Digit. Equip. Corp., Maynard, MA, USA, Rep. DEC-TR-301, vol. 38, 1984.

Alakesh Kalita is currently pursuing the Doctoral degree with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India.

His research interests include IoT and WSN.

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

He was Postdoctoral Research Fellow with SUTD, Singapore, from 2015 to 2016. He also worked with TCS, Mumbai, India, for two and half year. He is an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India. His research interests include performance evaluation of communication protocols, IoT, WSN, and network security.

Dr. Khatua is a member of ACM.