Time-Variant RGB Model for Minimal Cell Allocation and Scheduling in 6TiSCH Networks

Alakesh Kalita, *Member, IEEE*, and Manas Khatua, *Member, IEEE*

***Abstract*—IEEE has standardized the 802.15.4e Time Slotted Channel Hopping (TSCH) mode to provide stringent latency, higher reliability, and low duty-cycle in various Internet of Things (IoT) applications. TSCH eliminates interference and multi-path fading on channels, but its channel hopping feature severely affects the 6TiSCH (IPv6 over IEEE 802.15.4e TSCH mode) network formation. Further, 6TiSCH Minimal Configuration standard does not provide sufficient bandwidth (i.e., minimal cell) for quick transmission of control packets required by the new nodes (i.e., pledges) during their network association. Many works have been proposed on 6TiSCH network formation as it has high impact on network performance and lifetime. However, the existing works either did not use all the available physical channels while allocating minimal cell(s) or are not stable with topology changes. Therefore, this work proposes a Time-Variant RGB (TRGB) model for minimal cell allocation and scheduling, which results in faster association of pledges and maintains network stability. We evaluate the TRGB using Markov Chain model and also on a real 60-node testbed in FIT IoT-LAB. Testbed results show that TRGB achieves 51% and 23% improvement over the state-of-the-art scheme in terms of joining time and energy consumption, respectively, while maintaining stability of the network.**

***Index Terms*—IEEE 802.15.4e, 6TiSCH, network bootstrapping, scheduling, IoT network protocol.**

I. Introduction

THE 6TiSCH network is mainly designed for providing reliable, deterministic, and delay bounded end-to-end IPv6 based communication in Industrial Internet of Things (IIoT) [1], [2], [3], [4], [5]. IEEE 802.15.4e TSCH MAC mode [6] allows the nodes to change their physical communication channels after each transmission to get rid of interference and multi-path fading on channels. On the other hand, pledges[1] do not have idea on which channel the Enhanced Beacon (EB) frames are being transmitted by the already joined nodes, this makes the initial network synchronization challenging for the pledges. Pledges start scanning on all the available channels one by one to receive EB frame for network synchronization, which results in huge energy consumption, and so reduces overall network lifetime [5]. Furthermore, faster joining/association of the pledges with the underlying 6TiSCH network allows them to transmit their data packet quickly, which in turn reduces energy consumption.

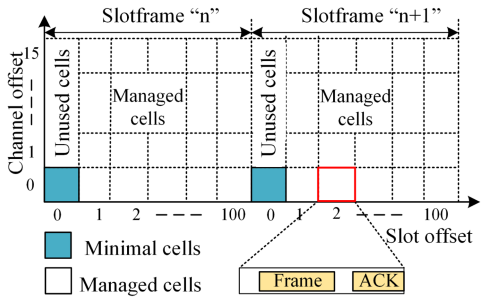


Fig. 1. Location of minimal cell in a slotframe in 6TiSCH-MC.

IETF released the 6TiSCH Minimal Configuration [7] (6TiSCH-MC) standard for describing minimal resource requirement in 6TiSCH network bootstrapping. 6TiSCH-MC mentioned that all the control packets should be transmitted in a single *minimal cell* of a *slotframe*[2] during or after 6TiSCH network bootstrapping and for maintaining stable network. However, existing works such as [8], [9], [10], [11], [12], [13], [14] have stated that the allocation made by 6TiSCH-MC is insufficient to transmit all the control packets during network bootstrap. Although the works [9], [10], [11], and [13] improved the 6TiSCH network formation performance, using the same resource allocation as 6TiSCH-MC, their improvements are not significant compared to the works in [8] and [12]. The work in [8] dynamically increased the number of minimal cells per slotframe horizontally (i.e., in different timeslots) depending on the number of generated control packets in the network, whereas work in [12] increased the number of minimal cells per slotframe vertically (i.e., in different channels) by autonomously allocating and scheduling them at the same time instant. In brief, [8] used different slot offsets and [12] used different channel offsets for allocating their additional minimal cells.

Fig. 1 shows the meaning of slot offset and channel offset, and default location of *minimal cell*, *unused cells* and *manageable cells* in a slotframe mentioned by 6TiSCH-MC. Manageable cells are used for sensory data transmission, which are scheduled by the distributed cell management protocols such as [15] and link scheduling schemes such as [16], [17], [18] after the

Manuscript received 13 December 2021; revised 30 December 2022; accepted 25 January 2023. Date of publication 31 January 2023; date of current version 8 January 2024. Recommended for acceptance by H. Liu. *(Corresponding author: Alakesh Kalita.)*

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

Digital Object Identifier 10.1109/TMC.2023.3241021

[1]IETF used the term *"pledge"* to designate a new node which wants to join a secure network by following the secure enrollment process.

[2]Slotframe is the collection of timeslots which repeats one after another, and one timeslot denotes the time duration (usually 10 ms) to transmit a frame and receive its acknowledgment.

network formation. Note that, except [12], none of the existing schemes used all the channel offsets at slot offset 0 for minimal cell allocation. Therefore, all the channel offsets, and so the physical channels remain unused except one, as shown in Fig. 1. Although [12] used all the channel offsets at the slot offset where the minimal cell resides, we find that the work [12] has several disadvantages, such as creates unstable network, nodes stuck with a single parent, and allows repeated control packet collision. In brief, using the scheme [12], a node might stick with a single parent as DIO packet is not transmitted in a common cell. So, the nodes do not know about their neighboring modes except their parents in [12], whereas the works in [8], [11], and [13] do not utilize all the physical channels at a time to transmit control packets. Furthermore, the works in [8] and [11], [12], [13] schedule to transmit all the control packets in the same shared cell, which affects the transmission of DIO packet, and thus degrade the performance of 6TiSCH network formation.

This work proposes to improve the pledges' joining/association process, and so, the performance of 6TiSCH network formation by efficiently utilizing the unused cells (shown in Fig. 1) as minimal cells. For doing this, at first we propose *time-variant autonomous allocation of minimal cell* (*INSTALL*) scheme, which autonomously allocates the minimal cell using the unique `EUI64` addresses of the nodes. This allocation is made time-variant so that nodes change their allocated minimal cells after each slotframe to reduce repeated control packet collision. *INSTALL* allocates at most four minimal cells to each node at the same slot offset without adding any signaling overhead in the network. As, *INSTALL* allocates different minimal cell(s) to different nodes at the same timeslot and a node can use only one minimal cell at a given slotframe, so, the question arises *which minimal cell will be used, when and by which node*? That means proper scheduling of the added minimal cells is required. Furthermore, the scheduling of the minimal cells should tightly interact with the RPL (Routing Protocol for Low-Power and Lossy Networks) [19] for both upward and downward routing. Therefore, we propose an autonomous *Red-Green-Blue* (*RGB*) scheduling algorithm for managing conflict-free scheduling of the allocated minimal cells of the nodes. *RGB* also schedules the different radio states i.e., transmission (`Tx`), reception (`Rx`) or `idle` of the parent-child pairs efficiently without any additional signaling overhead. Note that, both the *INSTALL* and *RGB* must run together by the already synchronized or joined nodes. On the other hand, pledges do not need to run these schemes as they only perform channel scanning at the initial stage of their joining process. In brief, the proposed TRGB scheme uses all the physical channels at a time by maintaining proper synchronization between the parent-child pairs, which significantly reduces congestion on a single channel, and it neither affects the manageable cells nor increases nodes' energy consumption, unlike the existing schemes on 6TiSCH network formation. Furthermore, TRGB separates the transmission of RPL traffic from other control traffic such as EB, keep-alive (discuss in Section IV-B3), which gives better flexibility in parent switching and network formation.

Both the proposed schemes i.e. *INSTALL* and *RGB* are implemented on Contiki-NG [20] and evaluated together. We named both the schemes together as Time-Variant RGB (TRGB) model. The performance of TRGB is evaluated using 60 nodes in FIT IoT-LAB [21], a public large scale open IoT testbed. Results show that *TRGB* outperforms all the benchmark schemes during 6TiSCH network formation. In brief, testbed results show *TRGB* performs 51% and 23% improvements over 6TiSCH-MC in terms of 6TiSCH joining time and energy (charge) consumption. The contributions of this paper are summarized as follows:

- A *time-variant autonomous allocation of minimal cell* (*INSTALL*) scheme is proposed for autonomously allocating the minimal cells to the nodes and changing the location of the allocated minimal cells periodically to aalleviate repeated collision.
- We propose *Red-Green-Blue* (*RGB*) scheme by which nodes can autonomously schedule different radio states such as `Tx`, `Rx` or `idle` of the allocated minimal cells.
- We design a Markov chain-based analytical model for multi-hop 6TiSCH network formation considering the proposed Time-Variant RGB (TRGB) model, which combines the *INSTALL* and *RGB* schemes together and show comparison based analytical results.
- We implement the *TRGB* model using FIT IoT-LAB real testbed for evaluating its performance. Results show that *TRGB* outperforms all the existing schemes in terms of joining time and energy consumption of the nodes.

II. Background

This section provides discussion about the IEEE 802.15.4e TSCH and formation process of 6TiSCH network in brief.

A. IEEE 802.15.4e TSCH

IEEE 802.15.4e includes the TSCH MAC behavior to provide dedicated channel access, reliability, low end-to-end packet delievery latency to the critical applications such as industrial automation, health-care system, process monitoring, oil and gas industry. TSCH divides time into small and fixed duration called timeslot that repeats over time. Every node of a network uses *cell*, which is a combination of a timeslot and transmission channel, for communication.

The parent-child pairs need to schedule their timeslot and transmission channel for communication. Basically, a parent-child pair computes the physical channel for communication by using the (1),

$$f = F[(ASN + channel\ offset)\ mod\ N_c] \tag{1}$$

where f denotes the calculated `channel index` associated with a particular frequency, F denotes the `lookup table` for channel hopping, ASN is the *absolute slot number* which denotes the number of total timeslots elapsed from the starting of the network. So, same ASN number is used by all the nodes present in a network and EB includes the current ASN value. One transmitter and receiver pair share the same channel offset for their communication. Hence, allocation of distinct channel offsets to different pairs of nodes allows them to communicate

using different physical channel at the same timeslot. The number of channels used in the network is indicated by N_c. Note that TSCH uses channel hopping concept utilizing the (1) by which a transmitting node can change its physical transmission channel to a destination node after every slotframe. This channel hopping feature helps the nodes to get rid of the impact of interference and multi-path fading on the communication channels, and thus the interference in 6TiSCH networks is very less.

B. 6TiSCH Network Formation

The formation of 6TiSCH network is initiated by the join registrar/coordinator (JRC) by broadcasting EB frame. Before receiving the EB frame, the pledges do not have idea about the channel where the transmission of control packet is taking place, and so they turn on their radios and start scanning on different random channels one by one for EB frame. This scanning process is repeated until the pledges receive valid EB. When a pledge receives an EB from any of the already joined nodes it becomes a TSCH synchronized node. The TSCH synchronized node knows the location of the control packet transmitting minimal cell, and so the channel in a slotframe from the received EB. So, it only activates its radio in the minimal cell during/after joining process for control packet transmission/reception to save energy. The TSCH synchronized node starts listening for the routing information carrying control packet to join the routing tree constructed by RPL i.e., RPL DODAG (Destination Oriented Directed Acyclic Graph). This routing packet is called DIO (DODAG Information Object) packet which is required by the TSCH synchronized nodes after completing their secure joining process by exchanging Join Request (JRQ) and Join Response (JRS) frames with the JRC via its parent (`join proxy`) node. When a valid DIO packet is received i.e., after joining the DODAG constructed by RPL., the TSCH synchronized node becomes a 6TiSCH joined node. Note that, only the 6TiSCH joined nodes are eligible to transmit their control packets for further expansion of stable network. Thus, when all the pledges become 6TiSCH joined nodes (or pledges join the DODAG), the formation of entire 6TiSCH network is considered to be completed. So, both the DODAG construction and 6TiSCH network formation mean the same event. In brief, the states followed by a pledge during its network association are pledge $\rightarrow$ TSCH synchronized node $\rightarrow$ TSCH secured joined node $\rightarrow$ 6TiSCH joined node.

III. Related Works

At the beginning, researchers proposed different approaches such as transmission of more number of EB frames in a slotframe [22], efficient selection of the collision-free EB broadcasting slots [23], [24], [25], [26] and active EB scanning [27], [28], [29] to reduce the initial channel scanning time of the pledges. However, their solutions are limited to TSCH synchronization and allow only one-hop communication, where only the transmission of EB frame is considered. To establish multi-hop communication, the TSCH synchronized node (one who has already received EB frame) should receive the routing information carrying DIO packet. Therefore, the 6TiSCH Minimal Configuration (6TiSCH-MC) [7] standard was published by IETF for multi-hop 6TiSCH network formation. This standard mentioned that only one *minimal cell* in a slotframe should be used for control packet transmission. The pledges should also receive both the EB and DIO control packets in order to fully join in the 6TiSCH network and transmit their own control packets for future multi-hop network extension.

Vallati et al. [8] revealed that the 6TiSCH-MC allocation is insufficient to transmit all the generated control packets. As a result of their work, the number of minimal cells per slotframe was dynamically increased, resulting in a significant performance improvement in terms of joining time of the pledges. However, this work has few drawbacks like it hinders the performance of network throughput and end-to-end packet delivery latency as data communication cells are converted into minimal cells. Furthermore, nodes consume more energy since they need to keep their radios turned on in the allocated minimal cell. Vucinic et al. [9] proposed a Bayesian based broadcasting strategy (BS), and Kalita and Khatua [10] proposed dynamic beacon generation (C2DBI) scheme to shorten the formation time of 6TiSCH networks by reducing congestion in the minimal cell. Further, Kalita and Khatua show the importance of routing information carrying DIO packet during the formation of 6TiSCH networks in [11] and studied the behavior of DIO packet generating Trickle algorithm in [13]. To achieve faster formation of 6TiSCH network, the works in [11] considered DIO as the highest priority packet during its requirement in the network instead of giving the highest priority to the EB frame all the time, unlike 6TiSCH-MC. Again, fairness in DIO transmission opportunity is ensured in [13]. Although these works improved the performance of 6TiSCH network during its formation compared to the 6TiSCH-MC, but failed to outperform the work in [8] because during the formation of 6TiSCH networks, they did not offer enough resource (i.e., minimal cell) to transmit all of the generated control packets.

All the above mentioned works related to multi-hop 6TiSCH network formation did not use all the channels (or channel offsets) at the slot offset(s) where the minimal cell(s) resides. The only work in [12] utilized these channel offsets by autonomously allocating, and then scheduling them as minimal cells at the slot offset 0. So, [12] increased the number of minimal cells per slotframe in the whole network, but, allows a node to use only one minimal cell in a given slotframe. Hence, this work achieved better performance compared to the existing schemes using similar low radio duty cycle as 6TiSCH-MC. However, this work did not maintain a common cell for all the nodes for exchanging routing information. This common cell is necessary for maintaining stable network and constructing efficient routing tree for both upward and downward routing. Furthermore, the nodes autonomously allocate their minimal cells only once, and then remain unchanged throughout the lifetime. Therefore, in this allocation, repeated collision of control packet transmission is possible when two or more nodes add the same minimal cell in every slotframe. Therefore, to deal with the above mentioned problems in the existing works and for efficient transmission of control packets in 6TiSCH networks, in this work, we increase

the number of minimal cell per slotframe without any additional impact on the energy consumption of the nodes.

IV. Design of TRGB Model

The works in [8] and [12] proved that the congestion in minimal cells gets reduced significantly when more number of minimal cells are used in a slotframe, which enables quick transmission of control packets, and so results in faster formation of 6TiSCH network. However, we do not use the manageable cells as minimal cells, unlike the work in [8], and therefore, the proposed approach *TRGB* does not hinder the throughput and end-to-end packet delivery of the networks. On the other hand, *TRGB* does not suffer from stability issue, unlike the work in [12] as it allocates dedicated shared cell for the transmission of DIO routing packets. *TRGB* has two sub-schemes, and both must be executed by all the joined and synchronized nodes. The first sub-scheme i.e., *time-variant autonomous allocation of minimal cell* (*INSTALL*) autonomously allocates minimal cells of the nodes whereas the other sub-scheme i.e., *Red-Green-Blue* (*RGB*) autonomously schedules the allocated minimal cells. In brief, *INSTALL* suffers from the synchronization issue for the parent-child pairs without the *RGB* scheme. Both these sub-schemes are described separately in the below subsections.

Algorithm 1: Time-Variant Autonomous Allocation of Minimal Cell.

if node is JRC **then**
  Add two minimal cells as follows:
  [0, 0] ▷Common cell
  [0, Own] ▷By (2)
else if node is only TSCH synchronized **then**
  if parent is JRC **then**
    Add two minimal cells as follows:
    [0,0] ▷Common cell
    [0, Parent] ▷By (3)
  else ▷i.e. parent is not JRC
    Add three minimal cells as follows:
    [0, 0] ▷Common cell
    [0, Parent] ▷By (3) [0, GrandPa] ▷By (4)
  end if
else if 6TiSCH joined node
  Add one more minimal cell as follows:
  [0, Own] ▷By (2)
else ▷It's a pledge
  Scan random channel for EB frame
end if

A. Time-Variant Autonomous Allocation of Minimal Cell

INSTALL uses the unique EUI64 addresses of the nodes to allocate minimal cells at slot offset 0. For the efficient construction of the routing tree i.e., RPL's DODAG, *INSTALL* adds a common cell using the slot offset 0 and channel offset 0 i.e., at cell [0,0] for all the nodes for exchanging routing information carrying DIO packet. Apart from this common minimal cell, *INSTALL* allows the nodes to add at most three more minimal cells at the same slot offset 0 but with different channel offsets depending on the position of the nodes in the DODAG tree. Let the different channel offsets x, y and z of three minimal cells are defined as follows:

$$x = mod(hash(\texttt{EUI64(Own)} + ASFC), N_c - 1) + 1 \quad (2)$$

$$y = mod(hash(\texttt{EUI64(Parent)} + ASFC), N_c - 1) + 1 \quad (3)$$

$$z = mod(hash(\texttt{EUI64(GrandPa)} + ASFC), N_c - 1) + 1 \quad (4)$$

where, hash[3] function is used to find a unique integer number from the EUI64 addresses of the nodes and current $ASFC$. The EUI64(own), EUI64(Parent), and EUI64(GrandPa) denote the EUI64 addresses of a node, its RPL routing parent and grandparent, respectively. Please note that according to 6TiSCH-MC standard, a node is allowed to transmit its own EB and DIO packet only after receiving both EB and DIO packet i.e., after becoming a complete 6TiSCH joined node. Hence, when a node transmits its EB frame, it has already found its parent (which may not be the best). When the 6TiSCH node transmits its own beacon, it also includes its parent address in the EB frame by adding an extra byte to the *information element* (IE) of an EB frame. Now, when a pledge receives the EB frame from the joined node, it automatically knows its parent and grandparent addresses from the received EB. After knowing these addresses, the pledge allocates its minimal cells based on the parent and grandparent addresses. $ASFC$ is the *absolute slotframe count* which is calculated as $\lfloor ASN/L \rfloor$. Here, ASN and L denote the *absolute slot number* and *slotframe length*, respectively. The function $\lfloor f \rfloor$ denotes the floor function and N_c denotes the number of used channels in the network. As the JRC does not have parent and grandparent, so JRC adds only one minimal cell corresponding to its own address. Similarly, the nodes that do not have grandparent i.e., one hop children of JRC, add two minimal cells considering the parent i.e., JRC's EUI64 address and their own EUI64 addresses. Otherwise, the nodes add three minimal cells at the slot offset 0 and the channel offsets x, y and z defined by the above equations.

Please note that at the beginning, the pledges do not know in which channel and at what time the already joined nodes transmit their control packets as mentioned in Section II-B. Thus, using the proposed scheme *INSTALL* also, at the beginning, the pledges do not have any idea about the channel and time where control packets are being transmitted by the joined nodes until they receive EB frame. Hence, only after receiving an EB frame, the pledges can determine all the R, G, and B slotframes. So, before receiving the EB, the pledges need to randomly scan all the channel one by one. Algorithm 1 describes the steps in *INSTALL* and following subsections describe it in more detail.

1) Alleviate Repeated Control Packet Collision: To reduce repeated usage of a set of minimal cells by more than one node,

[3]Network administrator can choose/design his/her own hash function which is implementation specific.

$ASFC$ is used in the above equations. After each slotframe, the value of $AFSC$ will be incremented by one i.e., $AFSC$ will vary over time. Therefore, $AFSC$ is used in the hashing equations so that the `hash` function returns different outputs after each slotframe. This prevents specific cells are being overlapped forever and alleviates repeated usage of the same set of minimal cells by more than one node. Ultimately, this reduces the repeated control packet collision. Therefore, $AFSC$ is included in the equations instead of allocating fixed channel offsets directly. Now, as x, y and z are used as channel offsets, so TSCH's channel hopping mechanism adds different transmission channels to the nodes over time. Otherwise, the nodes would always get the same physical channels if their allocated channel offsets are the same. Note that there will be `hash` collision when the number of nodes within a single hop network exceeds the total number of physical channels used in the network. `Hash` collision results in the assignment of the same minimal cell to the colliding nodes. However, packet collision is only possible when more than one node transmits its packets simultaneously using the same minimal cell.

2) Autonomous Allocation of Minimal Cell Without Signaling Overhead: A node can easily extract its parent and grandparent's `EUI64` addresses from the received EB frame. Hence, nodes can autonomously add their minimal cells at slot offset 0 without any signaling overhead. Furthermore, the `EUI64` addresses of the nodes are unique, so *INSTALL* allocates distinct minimal cells to the nodes. Please note that when a node changes its parent for any reason (such as based on the information received in DIO packet or the parent node is not reachable), it replaces the previously allocated minimal cells (based on the previous parent and grandparent `EUI64` addresses) by the minimal cells calculated based on the new parent and grandparent's `EUI64` addresses. The node changes its parent either by joining the network as a fresh node or being in a TSCH synchronized state. When the node joins as a fresh node, it deletes all the allocated minimal cells like the 6TiSCH-MC standard, waits for EB frame on a random channel, and subsequently allocates the minimal cells depending upon the next received EB frame. On the other hand, when the node changes its parent based on the information received in DIO packet (i.e., being in the TSCH synchronized state), it knows the parent `EUI64` address from the received DIO packet and subsequently comes to know about grandparent address to allocate the new minimal cells replacing the previously allocated minimal cells.

3) Leveraging Unused Cells: All the minimal cells of a node are added at the same slot offset but using different channel offsets, which vary with time. So, a node can use only one minimal cell per slotframe. Hence, *INSTALL* neither affects the existing data transmission schedule and manageable cells nor increases the radio duty cycles of the nodes. However, *INSTALL* increases the number of minimal cells per slotframe in the network without any signaling overhead.

4) An Example of INSTALL-Allocation: Fig. 2 shows an example of the allocation made by *INSTALL* in the first two consecutive slotframes i.e., $ASFC = 0$ and $ASFC = 1$. For example, node 5 allocates four minimal cells based on the

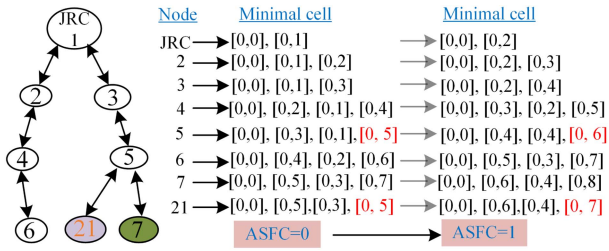


Fig. 2. Minimal cells allocation by *INSTALL*.

`EUI64` addresses of his own `[0,5]`, parent `[0,3]`, and grandparent `[0,1]` and including the common minimal cell `[0,0]` in $ASFC = 0$. However, in the next slotframe i.e., $ASFC = 1$, the positions of the added minimal cells of node 5 have changed to other locations except the common cell due to the used `hash` function and value of $ASFC$. All nodes change the location of autonomously allocated minimal cells after each slotframe as shown in Fig. 2. For example, node 5 and 21 use different minimal cells based on their respective `EUI64` addresses in $ASFC = 0$ and $ASFC = 1$. As mentioned before, nodes allocate their corresponding minimal cells depending on their parent and grandparent `EUI64` addresses at any given time. For example, if node 21 changes its parent to node 4, then node 21 reallocates minimal cells [0,0], [0,4], [0, 2], and [0, 7]. Thus, *INSTALL* alleviates repeated minimal cell overlapping, and so controls packet collision and supports node mobility.

B. Red-Green-Blue Scheduling

RGB is proposed to schedule the allocated minimal cells by *INSTALL* to avoid de-synchronization for the parent-child pairs. Furthermore, a node can be in any of the three different radio states in a given minimal cell of a slotframe `Tx`, `Rx` and `Idle`. *RGB* also schedules these radio states of a node effectively. For example, when the parent node is in `Tx` state, the child needs to be scheduled in `Rx` state, and vice versa. Algorithm 2 describes the steps of *RGB* and following subsections describe the proposed scheme *RGB* in more details.

Few important points about *RGB* scheduling are present in the following.

1) What These R, G, and B *Indicate?:* We consider a slotframe as *Red* (*R*) when the modulus of the ASN number of the starting timeslot of the slotframe to 3 is 0 i.e. $(starting\ ASN)\,\%\,3 = 0$. Similarly, we consider the slotframes *Green* (*G*) and *Blue* (*B*) when the modulus of the ASN number of the starting timeslot of the slotframes to 3 are 1 and 2, receptively. We consider the modulus of 3, so these R, G, and B slotframes are repeated after every 3 slotframes. Note that when the slotframe length is small, the R, G, and B slotframes appear more frequently and vice versa. It is also noteworthy that the same ASN is used by all the nodes in a network at a given timeslot. Hence, all the nodes consider a given slotframe either as R, G or B slotframe i.e., consideration of R, G, and B slotframes by the nodes do not vary from node to node. However,

Algorithm 2: Red-Green-Blue Scheduling.

```
JRC decides its Tx slotframe either at G or B
if a pledge receives EB in G then
    set flag = G
else
    set flag = B
end if
if current slotframe is R then
    Nodes set Tx/Rx at [0, 0]                          ▷Common cell
else if node is JRC then
    if current slotframe is decided Tx slotframe then
        Set Tx at [0,Own]                              ▷By (2)
    else
        Set Rx at [0,Own]                              ▷By (2)
    end if
else if current slotframe is flag then
    Set Rx at [0,Parent]                               ▷By (3)
else if current packet is for parent then
    if JRC is parent then
        Set Tx at [0,Parent]                           ▷By (3)
    else
        Set Tx at [0,GrandPa]                          ▷By (4)
    end if
else if packet is available then                       ▷Broadcast packet
    Set Tx at [0,Own]                                  ▷By (2)
else                                  ▷Own turn but no packet available
    Turn off the radio
end if
```

different nodes perform different radio activities in the R, G, and B, which are discussed in the below subsections.

2) Usage of R, G and B *Slotframes:* All the nodes reserve R slotframe for exchanging routing information carrying packets such as multicast/unicast DIS (DODAG Information Solicitation), DIO, whereas G and B slotframe are used either for Tx or Rx of EB, unicast DIO, keep-alive, unicast DIS, and DAO (Destination Advertisement Object) as described below.

3) Conflict-Free Radio States Scheduling: In *R* slotframe, all the nodes either transmit (Tx) or receive (Rx) routing information carrying packet using the common minimal cell i.e., [0, 0] depending on the availability of the routing packet in their transmission buffer. All the nodes are allowed to transmit their routing information using the common cell only so that the nodes can change different parents as the routing protocol discovers new neighbors over time via the reception of DIO packets. Therefore, if a node wants to change its parent, it should listen for DIO packet in the R slotframe. Again, assigning the ***R*slotframe** particularly for RPL traffic (excluding traffic such as EB, keep-alive, unicast RPL traffic) helps in efficient exchanging of routing packets by which nodes can easily discover better parent in less time and, consequently, find a better route to reach their destination.

On the other hand, in ***G*and *B*slotframes**, at the beginning, the JRC randomly chooses either G or B as its Tx slotframe and always uses that slotframe for uni-directional Tx slotframe while keeping the other slotframe as Rx slotframe. The other nodes perform opposite radio state (Tx/Rx) to their parent (Rx/Tx) in a given G or B slotframe. For example, if a node receives EBs from its parent at G that means the parent performs Tx at G, so the node (until the node performs re-association or changes its parent) considers G as Rx slotframe and B as Tx slotframe, otherwise, vice-versa. Hence, when a node is in Tx state, both the parent and child(s) are in Rx state, and vice versa. If we do not do this, both parent and child will be in the same radio state and create an unstable network.

Please note that when a node changes its parent, first it reallocates the minimal cells as per its new parent and grandparent EUI64 addresses (using *INSTALL*), and after that, the node configures its radio state opposite to the radio state of the new parent in G and B slotframes. However, the node needs to wait for one EB interval in worst case to decide its radio state in G and B slotframes. It is because the node would know about the radio state of its new parent in G and B slotframe only after receiving an EB frame. So the node has to be in *Rx* state after parent change in G and B slotframes. Please note that the node immediately transmits an updated DIO packet containing parent change information in the network [19]. Eventually, when the child nodes receive this updated DIO packet, they also look for reconfiguration of their radio states in G and B slotframes. The information about the radio states of a node in G and B slotframe can be transmitted in the *Optional Metric Container* of the DIO packets. However, in this work, we have not modified the DIO packet.

4) Which Minimal Cell Should be Used at What Time?: All the nodes (except JRC) always **listen** using their parent EUI64 addresses in either G or B slotframe. As JRC does not have a parent, so it listens on the minimal cell based on its own EUI64 address in its pre-decided Rx slotframe. Every node chooses one of its allocated minimal cells depending on the current control packet type to perform Tx operation. It works as follows, if the current control packet type is "*broadcast*" such as EB and for its children, then the node chooses the minimal cell based on its own EUI64 address. On the other hand, if the control packet type is "*unicast*" request packet to its parent such as unicast DIS or keep-alive, then the node chooses the minimal cell based on grandparent EUI64 address. It is because, at that moment, the parent would be listening using its own parent EUI64 address (i.e., grandparent address of the child node(s)). The nodes who do not have grandparent (i.e., their parent node is JRC), transmit their request control packets using the EUI64 address of the JRC. Thus, all nodes can autonomously schedule their minimal cells without any conflict with their parent and child node(s).

In brief, the common minimal cell is used for exchanging routing control packets i.e., for both Tx and Rx. The minimal cell based on node's own EUI64 address is used for broadcasting own control packets i.e., only for Tx; the minimal cell based on parent EUI64 address is used for listening control packets from parent i.e., only for Rx; and the minimal cell based on grandparent EUI64 address is used for transmitting unicast control packet to parent i.e., only for Tx.

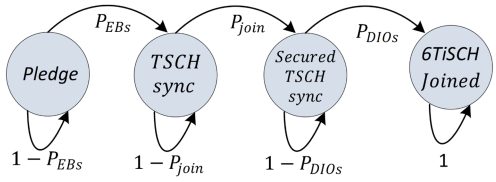


Fig. 3. Markov states of a pledge.

5) Energy Efficiency and Overhead in RGB: INSTALL already forces the nodes to use low *radio duty cycle* (RDC) as like 6TiSCH-MC. However, *RGB* further allows the nodes to turn off their radios when the nodes do not have any control packet to transmit in their Tx slotframes. It is because, at their Tx slotframe, the parent and child(s) of a node are in Rx state. So, there is no chance to receive control packets either from parent or child(s). Thus, *RGB* further reduces the radio duty cycles of the nodes, and so makes the scheduling more energy efficient.

Furthermore, the unused cells are converted into minimal cells without affecting the RDC and data communication cells, and both the cell allocation and scheduling are done autonomously without any additional signaling overhead. So, the proposed TRGB follows the 6TiSCH-MC standard without incurring any additional signaling overhead in the network.

V. Theoretical Study of TRGB

A. Theoretical Modeling of TRGB

In Section II-B, we discuss about the four states of the pledges i.e., new node/pledge, TSCH synchronized node, TSCH secured joined node, 6TiSCH joined node during their joining process. The *average joining time* (AJT) of a pledge to join a 6TiSCH network completely (to become a 6TiSCH joined node) depends on the average time spent in each individual states. As the probabilities of being in the intermediate states and the last state depend on the probabilities of being in their previous states, therefore, we model this four states of a pledge by the Markov Chain model as shown in Fig. 3. Note that using this analytical model, our aim is to show how increasing number of shared cells per slotframe helps in improving the performance of network formation, but not to offer a highly accurate model. Fig. 3 also shows the different successful control packets transmission probabilities of the nodes, which helps the pledges to move from one state to another state. Basically, P_{EB_S} denotes the successful EB transmission probability, P_{join} denotes the successful JRQ and JRS exchanging probabilities together i.e., P_{JRQ_S} and P_{JRS_S}, and P_{DIO_S} denotes the successful DIO packet transmission probability. The final state of the Markov model is absorbing state, where a pledge becomes a 6TiSCH joined node. As, the nodes are allowed to transmit their control packets once in a slotframe using the proposed scheme *TRGB*, therefore, the *average number of slotframes* (*ASF*) required to reach the final absorbing state by the pledges can be calculated as follows:

$$ASF = \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRQ_S}} + \frac{1}{P_{JRS_S}}\right) + \frac{1}{P_{DIO_S}} \quad (5)$$

TABLE I
List of Used Symbols

Symbol	Meaning
I_{eb}	Beacon generation interval
L	Slotframe length in timeslots
P_{eb}	EB transmission prob. in a minimal cell
P_{EB_S}	Prob. of successful EB trans. in a minimal cell
P_{jrq}	JRQ transmission prob. in a minimal cell
P_{JRQ_S}	Prob. of successful JRQ trans. in a minimal cell
P_{jrs}	JRS trans. prob. in a minimal cell
P_{JRS_S}	Successful JRS trans. prob. in a minimal cell
P_{dio}	DIO transmission prob. in a minimal cell
P_{DIO_S}	Prob. of successful DIO trans. in a minimal cell
N_c	Number of physical channels
AJT	Average joining time
ASF	Average number of slotframes

We summarize the frequently used symbols and their corresponding meaning in Table I. Multiplying the duration of a slotframe with *ASF* gives the AJT of a pledge to join a given multi-hop 6TiSCH network. So, to calculate the value of *ASF*, the values of P_{EB_S}, P_{JRS_S}, P_{JRQ_S}, and P_{DIO_S} need to be calculated. At first, the probability of successful EB transmission is computed as follows:

$$P_{EB_S} = \sum_{n}^{M-n} \frac{n}{N_c} \times n \times P_{eb}(1-P_{eb})^{\lfloor \frac{n}{N_c} \rfloor} \times (1-P_{jrs})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{loss})P(N=n) \quad (6)$$

where, n and M denote the number of already joined nodes (including JRC) and total number of nodes in the given multi-hop network, so $1 \leq n \leq M$. Different pledges can have different values of n in a given network. Therefore, the rest of the $(M-n)$ pledges need to join the network in presence of n joined nodes at any instant. The number of joined node at any instant is described by $P(N=n)$ and considering the uniform probability distribution, $P(N=n)$ can be written as $\frac{1}{M}$. Now, in a worst case network condition such as all the joined nodes are connected to JRC, when a 6TiSCH joined node transmits its EB in a slotframe with the probability P_{eb}, the other joined nodes $(n-1)$ should not transmit either EB or JRS frames i.e., denoted by $(1-P_{eb})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{jrs})^{\lfloor \frac{n}{N_c} \rfloor}$. Note that, here, other joined nodes means the nodes that are using the same minimal cell as the joined node and it happens when the number of joined surpluses the total number of channels used, N_c. The term $\lfloor \frac{n}{N_c} \rfloor$ describes this condition. Now, before getting synchronized with the TSCH network, the pledges do not in which channel and what time the already joined nodes transmit their control packet, so the entire probability value of P_{EB_S} is divided by the N_c. However, the EBs are broadcasted by the n joined nodes using different channels, and so n is multiplied with the total probability. Note that, *TRGB* allows the nodes to transmit DIO packet in R slotframe and JRQ frames are being transmitted by the pledges in parent Rx slotframe, so probabilities of transmitting these two control packets in a minimal cell i.e., P_{dio} and P_{jrs} are not considered in the above equation i.e., Equation (6). Only EB and JRS are transmitted either G or B slotframe depending on

the running RGB scheduling algorithm. In the equation, P_{loss} denotes the packet loss probability in the wireless medium.

Similarly, the successful transmission probabilities of different control packets such as P_{DIO_S}, P_{JRQ_S}, and P_{JRS_S} are calculated as follows:

$$P_{DIO_S} = \sum_{n}^{M-n} nP_{dio}(1-P_{dio})^{n-1} \times (1-P_{loss})P(N=n) \quad (7)$$

$$P_{JRQ_S} = \sum_{n}^{M-n} (M-n)P_{jrq}(1-P_{jrq})^{\lfloor \frac{M-n}{N_c} \rfloor} \times (1-P_{loss})P(N=n) \quad (8)$$

$$P_{JRS_S} = \sum_{n}^{M-n} n \times P_{jrs}(1-P_{eb})^{\lfloor \frac{n}{N_c} \rfloor} \times (1-P_{jrs})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{loss})P(N=n) \quad (9)$$

Pledges know the location of minimal cell after getting synchronized with the network, so, N_c is not used in the denominators of the (7), (8), and (9). Again, using *TRGB* joined nodes transmit their DIO packet in R slotframe, so, the transmission of DIO packet never collides with the transmission of other control packets such as EB, JRQ, JRS. These packets are transmitted either in G or B slotframe. Similarly, JRQ frame is transmitted in the parent `Rx` slotframe (i.e., transmitting node's `Tx` slotframe) using the minimal cell based on parent `EUI64` address. On the other hand, the joined nodes transmit their EBs using their own `EUI64` addresses, so collision of these packets is not possible. Therefore, in the (7), (8), and (9), collision among the packets is considered only when they are transmitted in the same slotframe. To get the final value of AJT, the EB and DIO transmission probabilities in a minimal cell by the joined nodes (P_{eb} and P_{dio}, respectively) are calculated as follows,

$$P_{eb} = L/I_{eb} \quad (10)$$

where, L and I_{eb} denote the slotframe length and beacon interval, respectively. And P_{dio} can be calculated as mentioned in [8] as follows,

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

where, I_{min} is the smallest DIO generation interval, P_r is the probability of Trickle reset, and N_D is the number of Trickle states. Finally, AJT of a pledge to join a multi-hop 6TiSCH network considering its *parent node joining time* (PJT) is calculated as follows,

$$AJT = PJT + ASF \times L \quad (11)$$

B. Theoretical Results of TRGB

1) Metrics: As both the joining time and energy consumption of the nodes are important for efficient data transmission and longer lifetime of 6TiSCH networks, therefore, we mainly considered three metrics to evaluate our proposed scheme *TRGB*. Before getting synchronized with the TSCH network, pledges need to scan for EB frame by keeping their radios active all the time. So, pledges consume their maximum energy during channel scanning. Hence, the time required by a pledge to be a TSCH synchronized node i.e., to reach the second state of the proposed Markov model is very much crucial. Therefore, we consider *TSCH synchronization time* of a pledge as our first metric to evaluate *TRGB*. The average TSCH synchronization time of a pledge for a given network is calculated by $\frac{1}{P_{EB_S}} \times L$.

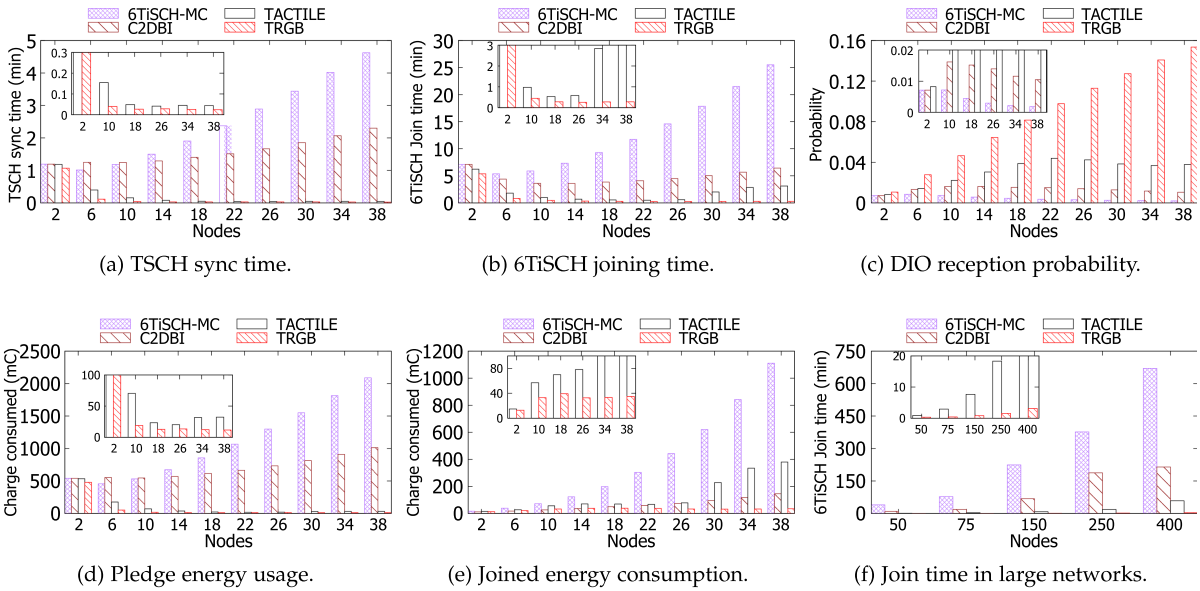
A pledge becomes a complete 6TiSCH joined node when it receives both the EB and DIO control packets and also successfully exchanges the JRQ and JRS frames with the JRC. In other way, it can be said that when a pledge reach the final absorbing state of the Markov model it becomes a 6TiSCH joined node. Only a 6TiSCH joined node is allowed to transmit its control packets and data packets. Hence, the time required to become a 6TiSCH joined node i.e., *6TiSCH joining time* should be minimum, which is considered as second metric for evaluation. The average 6TiSCH joining time of a pledge is calculated using the Equation (11).

As energy consumption of the nodes is important for longer network lifetime, therefore, we consider the *energy consumption* of the nodes during the formation of 6TiSCH network as the third metric for evaluation. The energy consumption (in terms of charge) of a pledge can be calculated by multiplying the total channel scanning time with the charged consumed by the pledge in unit time. More explanation about the charge consumption by both joined nodes and pledges is available in [10]. We followed the same approach to calculate the charge consumption of the nodes during network formation.

2) Parameter Details: To evaluate *TRGB*, we consider a congested topology where all the pledges join the JRC one by one. However, multi-hop topology can also be considered for evaluation using the proposed Markov model. The values of few parameters to evaluate the *TRGB* are taken as follows: $I_{eb} = 4sec$, $N_C = 16$, $I_{min} = 8ms$, $P_{loss} = 0.2$, $T_i = 10ms$, $L = 101timeslots$, $P_r = 0.2$, and $N_D = 16$.

3) Results and Discussion: We show the comparison based theoretical results on the above mentioned three metrics with the state-of-the-art schemes such as 6TiSCH-MC [7], C2DBI [10] and TACTILE [12] in Fig. 4. The complete theoretical analysis of TACTILE is given in [12], where as [10] contains the analysis of 6TiSCH-MC and C2DBI in details. We follow these works for getting the theoretical results of other schemes.

Fig. 4(a) and (b) show the TSCH synchronization time and 6TiSCH joining time of the pledges in different size networks, respectively. Both the joining times increase with the increasing number of nodes in the networks using the 6TiSCH-MC and C2DBI schemes. It is because both schemes transmit control packets using just one minimum cell per slotframe. As a result, as the number of nodes increases, so does the congestion



(a) TSCH sync time. (b) 6TiSCH joining time. (c) DIO reception probability.

(d) Pledge energy usage. (e) Joined energy consumption. (f) Join time in large networks.

Fig. 4. Comparison based theoretical results of TRGB. (a) TSCH sync time. (b) 6TiSCH joining time. (c) DIO reception probability. (d) Pledge energy usage. (e) Joined energy consumption. (f) Join time in large networks.

in the minimum cell, and hence the joining times. Although C2DBI reduces the congestion by dynamically varying the beacon generation interval, this modification is not sufficient for quick network formation. However, TACTILE reduces the congestion in minimal cell fully by allocating more minimal cell per slotframe using the unused cells. Hence, TACTILE improved both the joining times of the pledges significantly. However, using *TRGB*, the joining times are further improved compared to TACTILE. It is because, *TRGB* adds more minimal cells per slotframe and also separates the transmission of DIO packet from the other control packets. But, using TACTILE and other schemes, DIO packet gets less opportunity to get transmitted within a node because of the highest priority of EB frame.

Therefore, the pledges need to wait for more amount time to get the DIO packet compared to *TRGB*. To validate this claim, the values of successful DIO transmission probabilities P_{DIO_S} is shown in Fig. 4(c), where *TRGB* dominates all the other schemes. Furthermore, as the pledges do scan on random channel for EB frame, both the TACTILE and *TRGB* increase the EB reception probability of the pledges by transmitting more EB frames using different physical channels at the same time and it results in quick synchronization of the pledges with the TSCH network.

As the charge consumption of the pledges during their network joining process depends on the channel scanning time, therefore, using the *TRGB*, pledges consume less charge compared to the other schemes as shown in Fig. 4(d). It is because, using *TRGB*, pledges do not need to active their radios continuously for more amount of time, hence, it shows better results compared to the other schemes. Similarly, using *TRGB*, joined nodes do not need to transmit control packets (specially, DIO packet) more frequently as the pledges join the network quickly and start transmitting their control packets immediately. This improves the charge consumption of the joined nodes as shown in Fig. 4(e). Finally, Fig. 4(f) shows the performance of *TRGB* in large size networks. 6TiSCH joining time significantly increases with the increasing size of networks using 6TiSCH-MC because of the congestion in minimal cell. Although the congestion is reduced by C2DBI, TACTILE significantly reduces the congestion by employing more number of minimal cells per slotframe. However, *TRGB* outperforms all the schemes by using more number of minimal cell per slotframe and separating the transmission of DIO packet from the transmission of other control packets.

VI. Testbed Experiment

A. Testbed Setup

For the testbed experimental evaluation of *TRGB*, both the sub-schemes *INSTALL* and *RGB* have been implemented on Contiki-NG OS. The grandparents' `EUI64` addresses are added in the *information elements* (IE) of the EB frames and 32-bit integer mix function[4] is used for hashing. The binary file generated on Contiki-NG has been flashed in a static topology consisting of 32-bit ARM Cortex M3 micro-controller based 60 IoT nodes on the FIT IoT-LAB [21] in Strasbourg, France. The nodes are deployed in a 3D space as shown in Fig. 5, where Fig. 5(a) shows the physical deployment of the nodes and Fig. 5(b) shows the logical/routing tree ($1 \sim 2$ hops) constructed by the RPL. Nodes used all the 16 communication channels with -17 dBm packet transmission (`Tx`) power. The timeslot duration is taken as *10 ms* and RPL-Lite (storing mode) is used as network layer protocol. The other configuration settings used for testbed evaluation are mentioned in Table II.

[4]https://gist.github.com/badboy/6267743

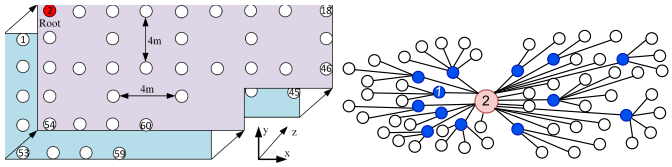


Fig. 5. Deployed 60 M3 nodes in Strasbourg FIT IoT-LAB (a) Physical topolgy. (b) Routing topology.

TABLE II
EXPERIMENTAL SETTINGS

Parameter	Value
Operating System	Contiki-NG
Mote	Cortex M3 IoT nodes
Testbed	FIT IoT-LAB
Timeslot length	10 ms
Slotframe length	37, 67, 101 timeslots
EB rate	2, 4, 16 slotframes
Number of channels	16
RPL version	RPL Lite
DRA's max. shared cells/SF	4
C2DBI's CBR measuring interval	8 seconds
Experiment duration	60 minutes

B. Metrics and Benchmark Schemes

The testbed experimental results are plotted with 95% confidence interval error bars by running each experiments 4-10 times for 60 minutes. The metrics described in Section V-B1 are used again to show the comparison-based testbed experimental results. Fig. 6(a), (b), and (c) show the TSCH synchronization time, 6TiSCH joining time, and charge consumption of the nodes, respectively. Apart from these metrics, the *radio duty cycle* (RDC) of the pledges and joined nodes, and the *number of parent switches* are also shown in Fig. 6(d), (e), and (f), receptively. Please note that we use the open-source *Energest module* of Contiki-NG to calculate the RDC of the nodes, and then using RDC, we calculate the charge consumption of the nodes. For this, the values of charge consumption for different radio activities of a node such as `Tx` and `Rx` are taken from from CC2650 datasheet.[5] The testbed results are obtained in different network conditions such as varied EB generation rate of the joined nodes and slotframe length, specifically, with varied congestion in minimal cell for better understanding of the schemes. Note that maximum 4 *minimal cells* per slotframe is taken for DRA and *8 seconds* is taken to measure the channel busy ration in C2DBI. *TRGB* is compared with the existing benchmark schemes such as 6TiSCH-MC [7], DRA [8], C2DBI [10], and TACTILE [12]. Note that we have not considered the scheme DRA [8] in our analytical analysis because of its dynamic behavior. The other parameters such as RPL traffic interval (e.g., minimum DIO interval, DODAG reset time), interval for keep-alive packet, and transmission and interference range are kept same for all the benchmark schemes for fair comparison among the schemes.

[5]https://tinyurl.com/8v7u7z3b

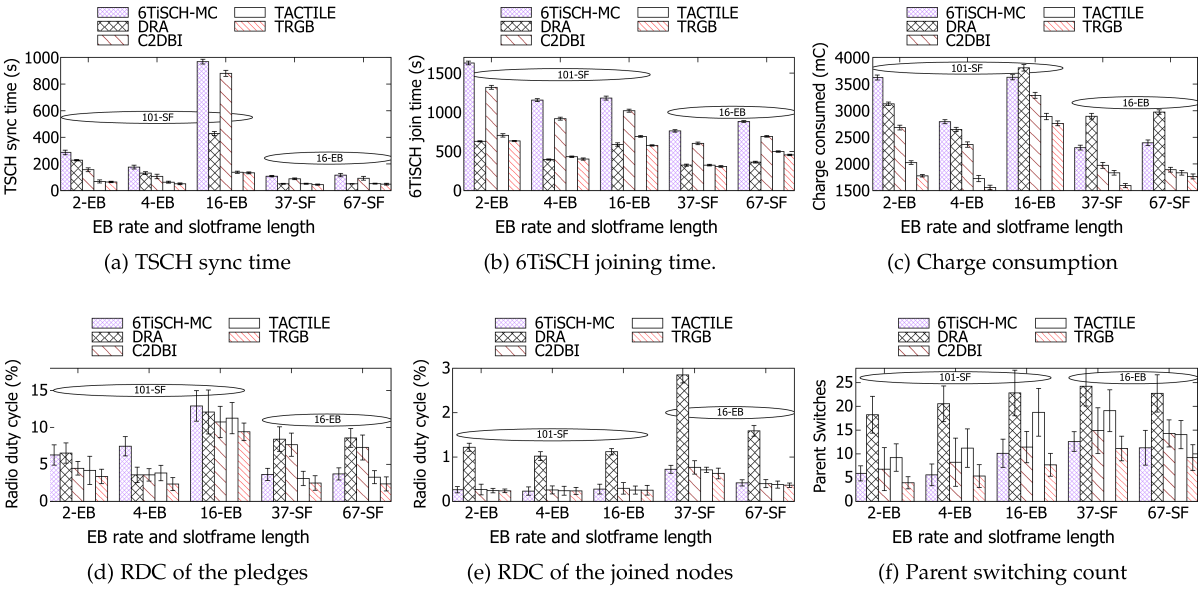
C. Testbed Experimental Results and Discussion

In Fig. 6, (b), (c), (d), (e), and (f), we show different testbed experimental results by varying the EB generation rate and slotframe length. Please note that, in the above mentioned figures, x-EB means an EB frame is generated by a joined node after x slotframes of size 101 timeslots and y-SF means a slotframe length equals y timeslots. We keep the EB generation interval fixed to 16 slotframes i.e. 16-EB with varied slotframe length. Testbed experimental results show that *TRGB* outperforms all the schemes (except DRA, in few cases) in terms of TSCH synchronization time and 6TiSCH joining time (Fig. 6(a) and (b)). The reason for this performance improvement is the usage of more number of minimal cells per slotframe which decreases the congestion in minimal cell during network formation. Although TACTILE increases the number of minimal cells per slotframe, it allows the nodes to transmit their DIO packet in the same minimal cell where they transmit their other control packets such as EB, unicast/multicast DIS, keep-alive, and DAO. This results in increasing waiting time for the pledges to receive DIO packets. On the other hand, *TRGB* separates the transmission of DIO packets from the transmission of other control packets, which results in faster joining of the pledges in both TSCH and 6TiSCH networks.

Please note that a pledge needs both EB frame and DIO packet to completely join in a 6TiSCH network. On the other hand, only EB frame is required by the pledge to get synchronized with the TSCH network. Furthermore, unlike EB generation interval, DIO packet generation interval is not periodic. Basically, DIO packet is generated using the Trickle algorithm [30] in 6TiSCH networks. It can be seen that the joining time of the nodes increase due to several reasons related to Trickle algorithm which are addressed in [11], [13], [31]. However, in this work, we consider the bandwidth issue (i.e., number of shared cells per slotframe) on the formation of 6TiSCH networks and we do not modify the default standard values (e.g., DIO generation rate) of different parameters.

All the existing schemes perform well when there is less congestion in the shared cell, for example, when the network is configured with the EB generation interval equals to 16 slotframes and slotframe length equal to 37 or 67 timeslots (i.e. 37-SF or 67-SF, as per the notation used in this paper). When nodes do not generate their EB frames frequently and the shared cell appears quickly, nodes get more opportunities to transmit their control packets. Therefore, all the schemes, including the proposed TRGB take less time for both TSCH synchronization (results are shown in Fig. 6(a) and 6TiSCH joining time (results are shown in Fig. 6(b)). So, TRGB does not show significant improvement in these network configurations compared to the benchmark schemes.

On the other hand, when the network is configured with limited bandwidth as slotframe length is set to 101 timeslots and EB packets are generated by the nodes very rapidly e.g., 2-EB, the nodes need to wait for a longer time to get DIO packet even though the nodes quickly receive EB frame and get synchronized with the TSCH network. Hence, 6TiSCH joining time of the nodes increases (results are shown in Fig. 6(b)) for the schemes C2DBI [10] and 6TiSCH-MC [7]. However, DRA [8],



(a) TSCH sync time (b) 6TiSCH joining time. (c) Charge consumption

(d) RDC of the pledges (e) RDC of the joined nodes (f) Parent switching count

Fig. 6. Comparison based testbed experiment results of TRGB. (a) TSCH sync time. (b) 6TiSCH joining time. (c) Charge consumption (d) RDC of the pledges (e) RDC of the joined nodes (f) Parent switching count.

TACTILE [12], and TRGB provides enough bandwidth to deal with the limited bandwidth allocation and higher EB transmission rate. Hence, all of these schemes show significant improvement in terms of joining time compared to 6TiSCH-MC and C2DBI. As DRA, TACTILE and TRGB use almost same number of shared cells per slotframe (i.e., allocated bandwidth for control packet transmission). Hence, all these schemes show almost similar results.

However, if we consider the energy consumption of DRA, it consumes highest energy compared to the TACTILE and TRGB, corresponding results are shown in Fig. 6(c), (d) and (e). As all the nodes need to keep their radios active during all shared cells for the scheme DRA, it results in increasing RDC of the joined nodes and pledges, and so, their charge consumption. Therefore, even though DRA exhibits almost similar performance in terms of 6TiSCH joining time, it consumes significantly more energy. On the other hand, TRGB consumes minimal energy compared to all the benchmark schemes as the nodes in TRGB need to keep their radios active only once in a slotframe. Please note that we consider the radio active time of a pledge till its TSCH synchronization completion time upon total experimental time to calculate the RDC of the pledge as a pledge changes its state after TSCH synchronization time. Similarly, RDC of a joined node is calculated based on the node's radio active time after it becomes a complete joined node till the total experimental time.

Furthermore, *TRGB* constructs more stable networks compared to the benchmark schemes. RPL tries to change the preferred parent when it does not get responses from the parent quickly. The default RPL assumes that responses are not received because of link failure, and so tries to change the parent, which creates unstable network. So, to get recovered from this issue, more control packets are transmitted by RPL, which again increases congestion in the minimal cell, and energy consumption of the nodes. Using 6TiSCH-MC, nodes change their parent more frequently as they do not get response from their recent parents because of the heavy congestion in the minimal cell. The parent switching results for all the schemes are shown in Fig. 6(f). One possible reason for more parent switching in DRA is EB's highest priority. Although nodes have sufficient resource to transmit their control packets, the other control packets need to wait longer time to get transmitted due to EB's highest priority, thus increases the waiting time of the nodes who have requested for control packets and so more parent switching occurs. However, by providing more number of minimal cells per slotframe and separating the transmission of DIO packet from other control packets, *TRGB* constructs more stable network. Please note that, during parent change a node can either be in TSCH synchronized state or rejoin the network as a fresh node. The node needs to keep its radio active all the time until it receives EB frame when it rejoins the network as a fresh node, and so energy consumption would increase further. However, we observe from our experimental results that most of the nodes change their parent they are in the TSCH synchronized state. Hence, there is no significant change in charge consumption and RDC (see Fig. 6(c), (d), and (e)) due to parent change (see Fig. 6(f)).

So, if we consider overall performance improvement considering all the three important metrics i.e., joining times, energy consumption and network stability, TRGB improves all the three metrics. In brief, TRGB takes less time to build more stable network by consuming less energy compared to all the benchmark schemes. For example, using the EB rate equals 16-EB and slotframe length equals 101-SF, *TRGB* achieves 51%, 2%, 43%, and 16% improvements in 6TiSCH formation time and 23%, 27%, 15%, and 4% improvements in charge consumption of the nodes compared to 6TiSCH-MC, DRA, C2DBI, and TACTILE, respectively.

VII. Conclusion

In this work, *TRGB* has been proposed to increase the number of minimum cells per slotframe for quick transmission of control packets, and so faster formation of 6TiSCH networks. *TRGB* efficiently leverages the unused cells of 6TiSCH-MC to use them as minimal cells and schedules them effectively for both collision free packet transmission and conflict-free radio state management of the nodes. *TRGB* also allows the nodes to change their allocated minimal cells after each slotframe to alleviate repeated control packet collision which occurs due to overlapping minimal cells. Apart from these, *TRGB* maintains a common cell for all the nodes to transmit only DIO packets so that it can tightly work with RPL routing protocol. Both the Markov Chain-based theoretical analysis and real testbed experiment on FIT IoT-LAB show that *TRGB* significantly improve the performance of 6TiSCH network formation. *TRGB* neither affects manageable cells, so the data transmission schedule, nor increases the radio duty cycles of the nodes, but maintains stable network.

References

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

[2] D. De Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4E: A survey," *Comput. Commun.*, vol. 88, pp. 1–24, 2016.

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

[4] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.

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

[6] *IEEE Standard for Local and Metropolitan Area Networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012 (Amendment to IEEE Standard 802.15.4–2011), pp. 1–225, Apr. 2012.

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

[9] M. Vucinic, T. Watteyne, and X. Vilajosana, "Broadcasting strategies in 6TiSCH networks," *Internet Technol. Lett.*, vol. 1, Dec. 2017, Art. no. e15. [Online]. Available: https://doi.org/10.1002/itl2.15

[10] A. Kalita and M. Khatua, "Channel condition based dynamic beacon interval for faster formation of 6TiSCH network," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2326–2337, Jul. 2021.

[11] A. Kalita and M. Khatua, "Opportunistic transmission of control packets for faster formation of 6TiSCH network," *ACM Trans. Internet Things*, vol. 2, no. 1, pp. 1–29, Jan. 2021.

[12] A. Kalita and M. Khatua, "Autonomous allocation and scheduling of minimal cell in 6TiSCH network," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12242–12250, 2021.

[13] A. Kalita and M. Khatua, "Adaptive control packet broadcasting scheme for faster 6TiSCH network bootstrapping," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17395–17402, Dec. 2021.

[14] A. Kalita and M. Khatua, "6TiSCH–IPv6 enabled open stack IoT network formation: A review," *ACM Trans. Internet Things*, vol. 3, no. 3, pp. 1–36, Jul. 2022.

[15] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH operation sublayer (6top) protocol (6P)," *Internet Eng. Task Force*, RFC8480, Nov. 2018.

[16] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, "6TiSCH minimal scheduling function (MSF)," RFC 9033, May 2021.

[17] S. Jeong, H.-S. Kim, J. Paek, and S. Bahk, "OST: On-demand TSCH scheduling with traffic-awareness," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 69–78.

[18] S. Kim, H.-S. Kim, and C. Kim, "ALICE: Autonomous link-based cell scheduling for TSCH," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, 2019, pp. 121–132.

[19] T. Winter et al., "RPL: IPv6 routing protocol for low-power and lossy networks," *Internet Eng. Task Force*, RFC6550, Mar. 2012.

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

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.*, 2016, pp. 1–9.

[26] 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.

[27] C. M. García Algora, V. Alfonso Reguera, E. M. García Fernández, and K. Steenhaut, "Parallel rendezvous-based association for IEEE 802.15.4 TSCH networks," *IEEE Sensors J.*, vol. 18, no. 21, pp. 9005–9020, Nov. 2018.

[28] B.-H. Bae and S.-H. Chung, "Fast synchronization scheme using 2-Way parallel rendezvous in IEEE 802.15.4 TSCH," *Sensors*, vol. 20, no. 5, 2020, Art. no. 1303.

[29] M. Mohamadi, B. Djamaa, M. R. Senouci, and A. Mellouk, "FAN: Fast and active network formation in IEEE 802.15.4 TSCH networks," *J. Netw. Comput. Appl.*, vol. 183/184, 2021, Art. no. 103026.

[30] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle algorithm," *Internet Eng. Task Force*, RFC6206, Mar. 2011.

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

Alakesh Kalita (Member, IEEE) received the PhD degree in computer science and engineering from the Indian Institute of Technology Guwahati, India, in 2022. He is currently a postdoctoral research fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include Internet of Things and edge/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 with SUTD, Singapore, from 2015 to 2016. His research interests include Internet of Things, wireless LANs, sensor networks, and network security.