Ad Hoc Networks 115 (2021) 102435

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

# Quick and efficient network access schemes for IoT devices

Kaumudi Singh [*], Vasanth Kumar T., T.V. Prabhakar, Joy Kuri

*Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru, India*

ARTICLE INFO

*Keywords:*
IEEE 802.15.4
IEEE 802.15.4e-TSCH
Guaranteed Time Slots
Joining time
Device Registration
Sparse Beacon Advertisement

ABSTRACT

In an IoT (Internet of Things) network, nodes need first to access the network resources before they can transmit data. If the data to be transmitted require real-time guarantees, then the node should not be stuck in the process of accessing resources. Quick and efficient access to the network resources is desirable for timely data communication. To this end, we have studied the IEEE 802.15.4 and IEEE 802.15.4e-TSCH standards and have proposed energy-efficient algorithms to ensure that the nodes can access the resources at the earliest. We have proposed a "Device Registration" algorithm for IEEE 802.15.4 networks, which aims at increasing accessibility of the Guaranteed Time Slots (GTSs) available in the superframe. The algorithm can be implemented by making minor modifications to the parameters within the existing MAC framework. Similarly, we have proposed a "Sparse Beacon Advertisement", a beacon scheduling algorithm, for IEEE 802.15.4e-TSCH networks that aims at reducing the wait time for a new node before it joins the network, even when there are few beacons being advertised. Both these algorithms have been evaluated extensively using simulations and experiments on a testbed. Our results show that in an IEEE 802.15.4 network, with the proposed algorithm, nodes are twice as successful in accessing the GTS resources as they were before. Similarly, Sparse Beacon Advertisement reduces the joining times by at least 60% in an IEEE 802.15.4e-TSCH network.

## 1. Introduction

Internet of Things (IoT) consists of a large number of interconnected sensor and actuator nodes that communicate with each other to facilitate intelligent and autonomous applications such as safety, security, environment monitoring, asset management, healthcare, etc. The application of energy harvesting technologies due to advancements in low power process technologies [1] has further accelerated untethered operations of these nodes. For example, state-of-art technologies can achieve an active current of 20 μA/MHz and a deep standby current of 150 nA [2]. Integrating these low power operational capabilities with low power communication technologies like Bluetooth Low Energy (BLE) [3], Z-Wave [4], IEEE 802.15.4 [5], and IEEE 802.15.4e [6] hold the promise of extended lifetimes for mission-critical condition monitoring applications. These applications generate sporadic data, which is mostly event triggered and must be communicated across the network with near-real-time guarantees. In these applications, the nodes usually stay in a low power mode and communicate only when events are detected. We also observe that the epicenter of such events is restricted to a subset of "active" nodes. For example, in an intrusion detection system, only the nodes at violated entry points will be triggered while detecting an intrusion. In contrast, the rest of the nodes can still maintain their regular duty cycle.

Any network communication comprises of a handshake over the control channel and an eventual data transmission on the data channel. There are many low power MAC schemes well suited for data channels. In contrast, the energy consumption of control channels is often ignored with the assumption that it is mostly setup related and will be incurred "only once". For instance, in the beacon enabled mode of IEEE 802.15.4 technology, nodes must contend to send out their access requests for the Guaranteed Time Slots (GTSs) in the Contention Access Period. This method not only suffers from nondeterministic delays to obtain resources, but it also exhausts a significant amount of energy. Even though the standard was amended in 2015 and two MAC behaviors, *viz.*, Deterministic and Synchronous Multichannel Extension (DSME) and Time Slotted Channel Hopping (TSCH), were incorporated in the IEEE 802.15.4e-2015 revision [6] to support real time guarantees, robustness, reliability and flexibility in data transmission in the emerging Industrial IoT (IIoT), assumptions regarding energy consumption in control channel remain consistently in place. The IEEE 802.15.4 technology has a massive market footprint. Currently, half a billion chipsets are available, and this number is likely to touch 4.5 billion by the year 2023 [7]. Additionally, manufacturers have also released combo chipsets. For instance, nRF52840 from Nordic Semiconductor is an advanced Bluetooth 5, Thread, and Zigbee multiprotocol SoC [8].

---

[*] Corresponding author.
*E-mail address:* kaumudi@iisc.ac.in (K. Singh).

https://doi.org/10.1016/j.adhoc.2021.102435
Received 15 May 2020; Received in revised form 4 January 2021; Accepted 19 January 2021

Moreover, the IEEE 802.15.4e-2015 amendments are limited to the MAC sublayer of the IEEE 802.15.4 while the physical layer of the standard is kept intact. Therefore, any node that supports the IEEE 802.15.4 radio can be upgraded to support TSCH based MAC as well. It is, therefore, important to study the energy consumption of communication protocols in a holistic manner by including control channels to complete the picture. In this work, we study the IEEE 802.15.4 and IEEE 802.15.4e-TSCH protocols and show how delays in control handshakes can be avoided to ensure ultra-fast access to resources and prompt data transmissions.

### *1.1. Motivation*

Although the beacon enabled mode of networking in IEEE 802.15.4 is a mature technology, it cannot support time-critical sporadic data transmission in its current form. Fig. 3 shows that the performance of slotted CSMA/CA, *i.e.*, the probability with which nodes can access the GTSs, is low due to two major shortcomings:

1. Channel assessments and backoffs are required to perform the control handshake for accessing the GTS resources, due to which the node may not be able to support real-time guarantees.
2. GTS resources, once acquired, get reserved, which might not be suitable when working with a network where the number of active nodes is dynamic. Reservation might lead to starvation of another node and poor utilization of these resources.

To this end, we propose a "Device Registration" algorithm, which acts as a precursor to the rich literature around the efficient GTS resource usage schemes. Our algorithm provides nodes with efficient access to GTS resources. To deal with the first concern, we restrict the transmission of request packets in the Contention Access Period (CAP) without any channel assessments and backoffs. Also, CAP is divided into "microslots" so that more number of request packets can be transmitted. In addition, we propose eliminating the reservation of GTS resources so that they can be appropriately utilized by the active nodes to mitigate the second concern. The proposed algorithm benefits networks that are powered by battery and harvested energy alike.

TSCH can support large networks that need strict real time guarantees [9]. It combines frequency hopping capabilities with Time Division Multiple Access (TDMA) and relies on Enhanced Beacons (EBs) for the formation of the network. These are broadcasted at regular intervals in the network on different channels according to the channel hopping rules to synchronize and advertise the network. A new node that wishes to join the network gets associated with the network once it receives one of these beacons. However, such a node is unaware of the channel hopping rules and might spend a large amount of time and energy waiting for the reception of an EB before it can sync with the network and start communicating. This can be seen from the entries corresponding to the Minimal Configuration in Fig. 6 and Table 2. This wait time is termed association time or the joining time. It depends upon the number of EBs being transmitted and their frequency. On the one hand, transmitting fewer EBs or transmitting them less often can lead to long joining times. However, transmitting large number of EBs or frequent EB transmission might lead to EB collisions which again will lead to long joining times. To address this, we have devised a beacon scheduling algorithm called the "Sparse Beacon Advertisement" that reduces the joining times in scenarios when there are only a few beacons being advertised in the network. Since we are able to achieve low joining times with few beacons, we are now not required to increase the number and frequency of beacon transmission. Hence, we can altogether avoid scenarios where the joining process is delayed due to beacon collisions. Reduced joining times also result in significant energy savings for the node trying to join the network.

The following are the major contributions of the paper:

1. Proposed a novel "Device Registration" algorithm for improving the real-time support of IEEE 802.15.4 and demonstrated that the proposed algorithm can be implemented with minor modifications to the parameters without violating the existing MAC framework of the beacon enabled mode of IEEE 802.15.4.
2. Evaluated the proposed "Device Registration" algorithm thoroughly using a testbed of 7 nodes, all of which support IEEE 802.15.4. The nodes form a star topology around the PAN coordinator. The transmission power is set at 0 dBm, and Channel 26 has been used for communication. Comprehensive MATLAB simulations were also performed to study the scalability of the proposed algorithm.
3. Proposed a "Sparse Beacon Advertisement" for scheduling Enhanced Beacons (EB) in an IEEE 802.15.4e-TSCH network to facilitate quick and efficient association of nodes with the network even when there are few EBs being broadcasted in the network.
4. Evaluated and compared the performance of the proposed "Sparse Beacon Advertisement" algorithm with different beacon scheduling algorithms with the help of extensive MATLAB simulations. A TSCH network was also implemented with the help of nodes that support IEEE 802.15.4. The network emulates the process of beacon advertisements in TSCH by combining beacon enabled mode of IEEE 802.15.4 with channel hopping capabilities. The network does not support data transmissions and is constructed solely to validate the reduction in joining times as observed during simulations and to calculate the reduction in the joining node's energy expenditure.

The rest of the paper has been organized as follows: Section 2 gives a small overview of the IEEE 802.15.4 and IEEE 802.15.4e-TSCH standards. Section 3 discusses the research works that have been conducted with respect to both these standards. Section 4 gives the details about the proposed "Device Registration" algorithm for the IEEE 802.15.4 networks along with the details of its implementation, evaluation, and a discussion on the obtained results. Details for "Sparse Beacon Advertisement" are provided in Section 5 along with the details of its implementation, evaluation, and insights from the results. The paper is concluded in Section 6.

## 2. Background

In this section, we give an overview of the MAC layer of IEEE 802.15.4 and IEEE 802.15.4e-TSCH.

### *2.1. IEEE 802.15.4 – Beacon-enabled mode of communication*

Network communication, with the beacon-enabled mode of IEEE 802.15.4, is structured using superframes, which are delimited by beacons broadcasted periodically by a PAN coordinator (henceforth called base station in this paper). The beacon is followed by an active period and an inactive period. The base station goes to a low power mode in the inactive period, and hence all the communication occurs within the active period. The active period is further divided into 16 equal slots, 9 of which make up the Contention Access Period (CAP), and the rest 7 constitute the Contention Free Period (CFP). In CAP, nodes contend among themselves to access the channel using slotted CSMA/CA. The slots available in CFP are called Guaranteed Time Slots (GTSs). In these slots, nodes have exclusive rights to use the channel, and hence, these slots can provide support for time-critical traffic.

When a node wishes to use the GTSs, it must transmit an allocation request to the base station, which essentially has information about the data length that the node is expected to transmit so that adequate GTSs can be allocated. The node sends this packet in CAP using slotted CSMA/CA for channel access. Once the base station receives the packet, it checks the requirement of the node and the available GTS resources.

If adequate GTS resources are available, the base station allocates the resource to the node and lets it know about the same with the next beacon. A successful node can keep using this resource until it sends an explicit deallocation request to the base station or until the base station notices a period of inactivity for the allocated resources and deallocates them.

### 2.2. IEEE 802.15.4e-TSCH – Time slotted channel hopping

TSCH combines frequency hopping capabilities with Time Division Multiple Access (TDMA) to reduce unwanted packet collision and low duty cycles. TSCH maintains a rigid time schedule by dividing the time into fixed length timeslots. A timeslot can accommodate a maximum size data packet and its associated acknowledgment. A collection of these timeslots forms a slotframe, and these slotframes repeat periodically. Each of the timeslots inside a slotframe is indexed, and the indices repeat with the periodicity of the slotframe.

Network formation in a TSCH network begins when a PAN coordinator wakes up and starts transmitting EBs. The frequency of EB transmissions increases progressively as more nodes successfully join the network. These beacons announce the presence of the network as well as maintain time synchronization in the network. Along with the slot index, each timeslot also has an associated Absolute Slot Number (ASN), which increments after every slot. $ASN0$ indicates the first ever timeslot in the network. PAN coordinator keeps track of this number and advertises the current value with every EB.

A node gets associated with the network when it receives an EB. The EBs provide the node with time and security related information, which synchronizes it with the network. Upon synchronization, the node can transmit its own EB and eventually start communicating with other nodes in the network. For communication, a link is provided to the node. The link is defined by a tuple - $[t_s, ch_{of}]$, where $t_s$ indicates the timeslot index in a slotframe and $ch_{of}$ is the channel offset to be used for the communication. Channel offset indicates the logical channel that is to be used, and its value belongs to the set $[0, N_c - 1]$, where $N_c$ is the total number of channels allowed for channel hopping. Technically, all 16 channels can be used but the user can blacklist specific channels based on their channel quality. The logical channel offset can be mapped to a physical channel ($ch_{ph}$) using a truth table, $F$, defined as [9]:

$$ch_{ph} = F[(ASN + ch_{of})\%N_c] \tag{1}$$

A node, upon waking up, randomly selects one of the physical channels being used for communication and stays on the same physical channel with its receiver on until and unless it receives an EB. Once the EB is received, it performs a handshake with the transmitter of the EB, thereby synchronizing with the network.

## 3. Related works

After having discussed some background about the protocols, we now discuss recent works in the field of IEEE 802.15.4 and IEEE 802.15.4e-TSCH.

### 3.1. IEEE 802.15.4 – Related works

Several works in the literature have studied the IEEE 802.15.4 MAC layer to enhance its performance [10]. Given the billions of devices in future IoT networks, successful access to GTS resources is essential and has not been addressed directly in the literature. In several GTS-assisted applications where there are hard deadlines to be met, a significant number of research works describe schemes and improvements in the manner GTS can be utilized. However, the mechanism and its issues, through which the base station realizes that there are nodes that have requested access to GTS resources, is not paid much attention. All the works assume that the base station already knows about the devices that need GTSs. [11] and [12] aim to improve GTS scheduling by assigning priorities to different nodes and then allocating the GTS resources. [13] analyzes GTS allocation for applications with deadlines and models the stability, delay, and throughput from a GTS allocation using Markov Chains. [14] has proposed an implicit mechanism using which the GTS can be allocated to different nodes in a time-sensitive network. Several investigations have been conducted into bandwidth utilization schemes for CFP. [15,16] address the fragmentation problem in GTS and allow multiple nodes to share GTSs. They have also divided GTSs into even smaller slots so that more nodes can be accommodated. Similarly, [17] allows adjusting the size of GTS according to the data size demand of the node which has requested to access the resource. [18] attempts to modify the active period in accordance with the number of nodes that are requesting to access GTS resource. They have also proposed that reducing the size of a single GTS can help accommodate data from a larger number of nodes than before. [19] proposes a resource management scheme for improving the bandwidth utilization and meeting delay requirements in a home automation system by modifying GTS command frame, implementing two different queues for GTS allocation and deallocation, scheduling GTS resources in an optimal manner and operating sensors with power saving algorithm. [20] proposes an advanced GTS scheduling which eliminates the GTS request packets during CAP and allocates GTSs to nodes in order of their discovery. However, even with the proposal, a maximum of 7 nodes can be scheduled in the CAP period.

To summarize, most works focus on strategies to allocate GTS resources among the requesting nodes. They assume that the devices *have completed registration and are just waiting to be allocated a GTS*. In contrast, we address the problem related to standard access mechanism and aim towards *improving device registration performance* so that the available literature on GTS utilization can be used to their full potential.

### 3.2. IEEE 802.15.4e - TSCH – Related works

As discussed in Introduction 1, a node needs to receive an EB before it can join the network, and the joining time can vary depending on the manner the beacons are scheduled in the network.

TSCH does not provide a beacon scheduling algorithm, but Internet Engineering Task Force (IETF) has defined a Minimal Configuration that states the minimal set of rules for bootstrapping a TSCH network [21]. Minimal scheduling algorithm schedules one slot every slotframe for the transmission of beacons. This configuration is well suited when the network has reached its steady-state operation, and there are very few new nodes waiting to join the network. However, one slot per slotframe is not sufficient to ensure quick association during network bootstrap. Consequently, several research studies have been conducted to evaluate the network association process and investigate avenues for further improvement [22,23]. Markov Chain based analysis has been performed for studying the network formation [24, 25]. [24] proposes a random advertisement algorithm for transmissions of EBs, and the formulation concludes that each node which is a part of the network should transmit EBs with a probability that is a function of the total number of nodes in the network. [25] calculates the average association time and formulates an optimal schedule for beacon advertisement that minimizes the association time. Furthermore, a near-optimal schedule, *viz.*, Model Based Schedule (MBS) has been devised. Various beacon scheduling algorithms, *viz.*, Random Vertical (RV) Filling, Random Horizontal (RH) Filling, Enhanced Coordinated Vertical (ECV) Filling, and Enhanced Coordinated Horizontal (ECH) Filling have been proposed [26,27]. ECV and ECH are extensions of RV and RH, respectively, and are more complex and energy-hungry. These algorithms allocate the different timeslots and channel offset for advertising the beacons. Average association time has been modeled as a function of node density, reliability, and frequency of beacon transmission. Experiments are also performed to evaluate the performance of the proposed algorithm. Deterministic Beacon Algorithm (DBA) has been proposed and shown to perform better than RV and RH algorithms [28]. Enhanced DBA (EDBA) is proposed in [29]. EDBA is also modeled based on Markov Chain, and it aims towards a collision-free beacon advertisement that minimizes the average association time. The algorithm is evaluated using NS3 simulations and outperforms the MBS algorithm. Advertisement Timeslot Partitioning (ATP) [30] increases EB transmission rate by sending multiple EBs in the same timeslot as EBs are much shorter than the maximum size data packet that can be accommodated in a timeslot. Another autonomous EB scheduling policy named Collision Free Advertisement Scheduling (CFAS) has been proposed [31]. CFAS claims that collisions of EBs are completely avoided as the nodes pick slots for EB transmissions based on their unique identifiers, and its performance is shown to be better than that of ECV and ECH algorithms.

As can be seen from the literature, almost all research works aim towards allocating sufficient resources for beacon transmission. However, the benefits of these algorithms can be reaped once multiple EBs are being transmitted in the network. The performance does not improve a lot when very few beacons are being transmitted in the network. In this paper, we aim towards scheduling the beacons so that even with few beacons being advertised in the network, the nodes can quickly join the network.

## 4. Adapting IEEE 802.15.4 for improved device registration probability

To implement the proposed "Device Registration" algorithm for IEEE 802.15.4 networks, we have considered a network that consists of energy-constrained nodes, out of which $n$ nodes are active at any given time, and a grid-powered base station (Fig. 1(a)). The nodes and the base station are arranged in a star topology. To support real-time data, the network implements the beacon-enabled mode of IEEE 802.15.4. Data transmission is prohibited during CAP and allowed only using the GTSs. To gain access to the GTSs, active nodes need to transmit their registration requests during CAP. The registration request packet is a standard IEEE 802.15.4 packet with one-byte payload. If the request packet of a node reaches the base station without any collision or corruption, the node is said to have "registered" successfully with the base station. As indicated in Section 2.1, 9 slots of CAP are significantly larger than the transmission duration of a single registration request. With these bigger slots, only a few registration requests can be accommodated within the CAP, and this reduces the chances of registration. Therefore, we have divided the CAP into much smaller slots, each of which is adequate for transmitting a single registration request. These smaller slots are termed "microslots", and the nodes choose one of these microslots for transmitting their request. With smaller microslots, the nodes can choose from a larger collection, and hence the chances of successful registration increase.

A deadline, $D$, is set by the base station to complete device registration, and nodes that can successfully register within this time are allotted a GTS. The deadline is announced by the base station as a part of the beacon payload and is set to support the real-time guarantees. The basic idea is that all active nodes (ideally) should register within the set deadline. The CAP is divided into $m$ microslots (Fig. 1(b)), each of duration $(D_m)$. The number of microslots $(N_m)$ available within the deadline is given by Eq. (2).

$$N_m = \lfloor \frac{D}{D_m} \rfloor \tag{2}$$

The nodes choose one of these available microslots randomly and independently for transmitting their requests. We can list the key points of the proposed device registration algorithm as follows:

1. Eliminate clear channel assessment and multiple backoffs,
2. Divide the CAP into multiple smaller microslots,
3. Fix deadlines in CAP to support real-time guarantees,
4. Prohibit nodes from transmitting data in the CAP,
5. Free GTS resources after every superframe and therefore eliminating unnecessary reservation of GTS across superframes.

The pseudocode of the proposed algorithm is shown in Algorithm 1. Once a node has calculated $N_m$ and picked a random microslot $(m_{picked})$ from the available ones for transmitting the registration request, it backs off till time $T$, given by Eq. (3), where $T_B$ is the time when the beacon arrived. At time $T$, it transmits the registration request without any clear channel assessment and goes to low power mode again. When the deadline $D$ gets over, the node wakes up to receive a transmission from the base station to determine if they have successfully registered. The successful nodes are allocated GTSs in the order of their discovery and can now transmit data in their respective slot.

$$T = T_B + D_m \times m_{picked} \tag{3}$$

**Algorithm 1** Device Registration for Star Network

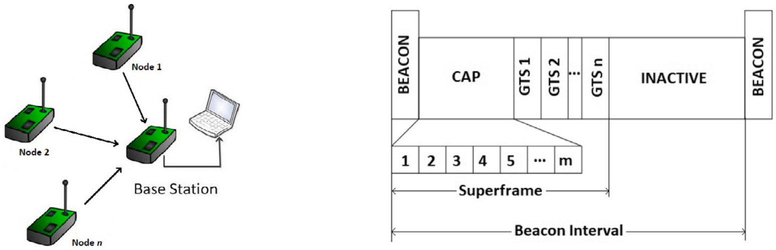
```
procedure RegisterDevice(D)
    Calculate $N_m$ using Equation (2).
    Pick a microslot randomly between 1 and $N_m$.
    Calculate $T$ using Equation (3) and take backoff.
    Transmit registration request at $T$.
    Go to low power mode till $D$.
    Receive transmission from the base station at $D$.
    if Registration == Successful then
        Transmit data in GTS.
    else
        Go to low power mode till next beacon
```

The number of nodes registering successfully within the deadline D is, of course, a random variable. A detailed study of its corresponding probability distribution is available in [32].

### 4.1. Implementation

The wireless sensor node platform uses CC2530 system-on-chip solution from Texas Instruments [33], which supports IEEE 802.15.4 radio with a receiver sensitivity of −97dBm. There are seven nodes in the network along with a base station, and they transmit packets at 0 dBm. Channel 26 has been used for conducting the experiments. The hardware linear forward shift register pseudorandom number generator in-built in the SoC has been used for generating the random numbers for the algorithm. The registration packet, which is 13 bytes long, takes a total of about 500 μs for transmission. Along with this, around 400 μs is required for radio turn around (switching of the radio from reception mode to transmission mode and vice versa). From this result, it was concluded that the duration of the microslot, *i.e.*, $D_m$, can be fixed at 1 ms. The algorithm is implemented using the following modifications to various parameters of IEEE 802.15.4 MAC:

1. The standard framework of a superframe has been kept intact. The base station broadcasts beacons regularly to maintain synchronization across the network. We, however, have confined data transmissions to CFP, and only the registration requests are transmitted in CAP. Deadline has been introduced in CAP to ensure that the nodes can inform the base station regarding their transmission requirements before the CFP starts. This provides the base station with enough time to recognize the nodes that have successfully registered and, thus, allocate appropriate resources.
2. The standard suggests that the basic slot period should be at least $aUnitBackoffPeriod$ long, which is about 20 symbol periods or 320 μs [34]. In our implementation, following the standard norms, we have considered this basic period as the duration of a microslot which is $D_m = 1$ ms long.



(a) System model for the network. Nodes communicate directly with the base station but not among themselves.

(b) Modified superframe structure which shows the division of CAP into microslots.

**Fig. 1.** System configurations — Topology and Communication.

3. After a node receives a beacon; if it wishes to communicate data, the standard suggests that the node should take a random backoff. The backoff exponent is then initialized as $BE = macMinBE$, whose default value is 3. Then the number of backoff slots is given as $BS = rand(0, 2^{BE} - 1)$, each of period $aUnitBackoffPeriod$ [34]. In our implementation, we have modified the number of backoff slots as $BS = rand(1, N_m)$, each of duration $D_m$.
4. According to the standard, the channel assessments performed after any backoff is controlled by the Contention Window ($CW$) parameter. The standard initializes its value to 2 [34] so that two channel assessments are performed before the nodes transmit data. The channel assessments are performed until and unless $CW \neq 0$. We eliminate the channel assessment before transmission of the request packet by fixing $CW = 0$.
5. According to the standard, the originator of a message may or may not request the recipient for an associated acknowledgment [34]. If the acknowledgment is not requested, the originator assumes that the transmission was successful. In our implementation, the nodes do not ask for an immediate acknowledgment from the base station after they transmit the registration request and go to a low power mode following the transmission. In addition, the nodes are not allowed to retransmit the request packets, and thus, we initialize, $aMaxFrameRetries = 0$.
6. After the registration deadline $D$ gets over, the base station transmits a broadcast containing the IDs of the nodes that have successfully registered along with the sequence in which they can transmit their data in the CFP. This implies that the broadcast also contains information about the resources that have been allocated to different nodes. This transmission is similar to GTS descriptors [14] that are added to beacons.
7. The nodes that have successfully registered can use the GTS resources available after the registration period. However, once the superframe is over, the resources are no longer available to the same set of nodes. In the next superframe, nodes with data to send must again attempt to reserve GTS slots. Therefore, in our proposed algorithm, there is no announcement of GTS reservations using the GTS descriptors of a beacon, and therefore, GTS descriptors are not added to the beacons.

### *4.2. Evaluation*

The proposed algorithm has been evaluated with the help of extensive simulations and experiments. The experimental testbed is shown in Fig. 2. To emulate that out of all the nodes in a network, only a few are 'active' at any given time, nodes are powered using solar panels and are placed under solar emulators. When the solar emulators were switched off, the associated nodes go to a low power mode. The nodes are activated when their associated emulator is turned on. The base station, on the other hand, is powered using batteries to ensure that the availability of energy does not constrain its operation.

#### *4.2.1. Experimental results*

To conduct the experiments, a Beacon Interval of 2 s has been set up. The active period is set to be about 123 ms long. This is achieved by setting the values of Beacon Order, $BO = 7$ and Superframe Order, $SO = 3$ [10]. A deadline of 49 ms has been set to schedule the transmission from the base station, which informs the nodes if their registration is successful at the beginning of CFP. There are 7 active nodes ($N$) in the network. The experiments capture how successful the nodes are in registering with the base station when deadlines are specified. The chosen metric is success probability. To calculate this, the experiment is run continuously for 1000 superframes and this exercise is repeated 10 times. Fig. 3 shows a comparison between the results obtained using the proposed algorithm and the IEEE 802.15.4 standard on a network with 7 nodes. The $x$-axis depicts the number of nodes, $x$, and the y-axis represents the tail probability of $x$ nodes registering within the deadline. For the comparison, the CAP follows the slotted CSMA/CA. To maintain fairness among both the implementations, we maintain a deadline of 49 ms for both of them. In the standard implementation, to ensure that the deadline coincides with the slot boundary of one of the 9 slots of CAP, CAP is divided into 7 ms slots, and the first 7 slots are used for registration. The results depict that the performance of our algorithm shows 2x improvement as compared to the standard slotted CSMA/CA. The experimental results also illustrate that the results were obtained with a confidence interval of $\pm 0.025$ with a confidence of 0.9.

#### *4.2.2. Simulation study*

We have carried out extensive simulations of the proposed system model [32] to obtain insight into the scalability of the algorithm. MATLAB simulations are performed to figure out the efficacy of the algorithm when we consider a large number of nodes as with large node populations, performing experiments is a daunting task. The network maintains a star topology, and the number of nodes in the network is varied from 10 - 30 for the simulations. The registration deadline is kept fixed at 49 ms. All the nodes are considered to be within the communication range of the base station. Interference and capture effects have not been considered while performing the simulations. Therefore, a packet is considered to be lost only when it collides with another packet. In addition, we also simulate slotted CSMA/CA, where nodes take backoffs and perform channel assessment before transmitting the registration packets. However, in situations that leads to the collision of two or more registration packets, retransmissions are not allowed so that a fair comparison between the two algorithms can be attempted. To calculate the success probabilities, 100,000 superframes are simulated. Fig. 4 compares the performance of the proposed algorithm with that of slotted CSMA/CA for large networks. The nomenclature of this figure is similar to that of Fig. 3.

It can be observed from the comparison that the proposed "Device Registration" algorithm provides better success probabilities for registration packets than those offered by the standard setup of slotted CSMA/CA. We can see from the figure that in a situation where $N = 10$

**Fig. 2.** Testbed Setup. The setup is shown on the left and a single sensor node is shown on the right.

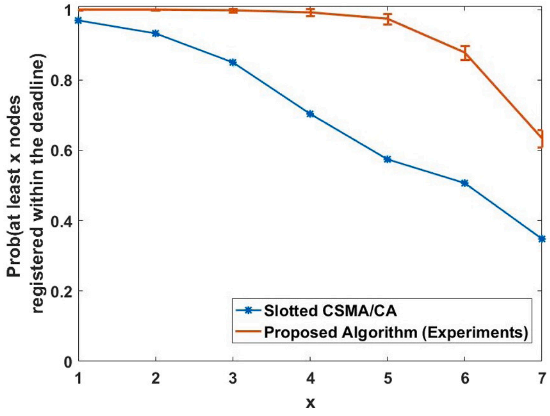


**Fig. 3.** Comparison between the proposed Device Registration Algorithm and slotted CSMA/CA with 7 nodes in the network and a deadline of 49 ms.

nodes are active in the network, the probability that all of them register successfully is about 0.4 when the proposed algorithm is employed. On the other hand, when slotted CSMA/CA is employed, the same probability drops to less than 0.05. Despite the superior performance of the proposed algorithm, it does not scale well, and the probabilities of successful registration drop significantly with the increase in the number of active nodes in the network, as depicted in the figure. If we increase the deadline, a larger number of nodes can be accommodated. However, in doing so, we may also have to extend the active period of the superframe so that there are enough CFP resources to accommodate data from all the nodes.

### 4.2.3. Energy savings

Having demonstrated the ability of the algorithm to improve the device registration success, we calculate the energy efficiency of the algorithm. The results are compared with the scheme when slotted CSMA/CA is used for transmitting registration requests. To calculate the energy consumed by a node during its operation, we measure the current it is drawing [35] and the voltage it is operating at. The product of these two quantities, along with the time for which the node is in operation, delivers the energy consumed by the node. In our proposal, the nodes switch on their radios only when they wish to transmit the registration request and keep them switched off for the rest of the period. However, the timers must stay active so that proper duty cycles can be maintained. The node takes one microslot for transmission and maintains low power for the rest of the registration period. This is a significant factor that leads to energy saving in the proposed algorithm as the channel assessment and backoffs have been eliminated.

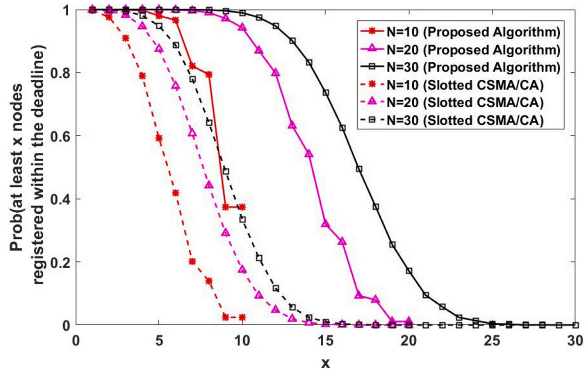


**Fig. 4.** Comparison of success probabilities, obtained with the proposed algorithm and slotted CSMA/CA, with different number of nodes in the network. The deadline is set to 49 ms.

**Table 1**
Comparison of energy consumed while accessing the GTSs using the slotted CSMA/CA $E_{total}(CSMA)$ and the Proposed Algorithm (PA) $E_{total}(PA)$ (Energy in mJ).

| Deadline | $E_{total}(CSMA)$ | $E_{total}(PA)$ | Percentage improvement |
|---|---|---|---|
| 49 ms | 1.52 | 1.38 | 9.2 |
| 56 ms | 1.71 | 1.54 | 9.9 |

The energy comparison can be made for the best and the worst transmission scenarios for a node. The best transmission scenario assumes that the node gets a transmission slot at the beginning of the registration period. The energy consumed under the best transmission scenario is almost the same with both slotted CSMA/CA and the proposed algorithm. The worst transmission scenario, however, assumes that the node does not get any transmission slot until the very end of the registration period. This implies that when the proposed Device Registration algorithm has been employed, the node stays in a low power mode for the entire duration of the registration period and wakes up at the last slot to transmit its registration request. On the other hand, if a node is using slotted CSMA/CA for transmitting the registration packet, it means that the node has performed several channel assessments, encountered a busy channel in each of these assessments, has taken appropriate backoffs and has found a free channel only at the end. In such situations, the node spends a considerable amount of energy performing channel assessments apart from the energy consumed in the transmission of the registration request packet. If the deadlines are large, the nodes might have to perform more channel assessments than what would have sufficed for scenarios with small deadlines. Similarly, with the proposed algorithm, with longer deadlines, the nodes stay in low power mode for long periods, and this leads to a slight increase in energy consumption. For example, for a deadline of 49 ms, the node stays in low power mode for 48 ms, and for a deadline of 56 ms, this time increases to 55 ms. Table 1 tabulates the energy consumed by a node with both the slotted CSMA/CA standard and the proposed algorithm under this worst-case scenario. The energy comparison shows that for every node, the energy consumption reduces by 10% when the proposed algorithm is used.

### 4.3. Device registration – Discussion and insights

Several insights can be gathered from the results obtained after evaluating the proposed device registration algorithm for IEEE 802.15.4.

First and foremost, we can note from Fig. 3 that the proposed Device Registration algorithm shows almost a two-fold (2x) increase in the success probabilities for registration compared to the slotted CSMA/CA scheme. We can observe from the same figure that with the proposed scheme, 5 out of 7 nodes are able to register successfully with a probability of 0.9. Whereas, in the case of the slotted CSMA/CA scheme, this event has a probability of 0.57. Only 5 nodes register with the base station when the registration requests of the other two nodes have collided. The chances of such collisions have been brought down to just 10% with the proposed scheme.

A significant step that we have implemented to achieve this improvement is to eliminate channel assessments and backoffs. As reported in Table 1, this choice leads to an energy savings of about 10%. Eliminating channel assessments and backoffs imply that mechanisms to avoid collisions have been removed. To reduce the collisions of the registration packets, the division of CAP into microslots comes to our aid. This division provides the nodes with ample slots to choose from for transmitting their registration requests, thereby, reducing the chances of their collisions. In spite of this, the registration packets do experience collisions, and since retransmissions are not allowed, the nodes whose registration packets have collided are unable to register successfully with the base station. To avoid such situations, we can extend the algorithm and allow multiple transmissions of the registration requests. [32] discusses this in more detail. It should be noted that eliminating channel assessment and backoffs, along with the prohibition of data transmissions in CAP, does not hinder the eventual data transmission. The nodes in our network have to transmit data with real-time guarantees, and in such a situation, transmitting data packets in CAP will not be advisable as contention-based access cannot guarantee timely delivery of data. Therefore, the data must be transmitted data during CFP and not during CAP.

The IEEE 802.15.4 allows only 7 slots in CFP. According to our implementation, where the active period is 123 ms (Section 4.2.1), CFP is 53 ms long and a total of 1656 bytes can be transmitted in this duration at a data rate of 250 kbps. However, this number is not absolute and can be varied by varying the duration of the active period. However, the standard suggests that a maximum of 7 nodes can be accommodated. This is definitely not enough to satisfy the requirements of a network if there are more than 7 nodes requesting the usage of GTS. Therefore, several works in the literature attempt to improve the GTS resources utilization (Section 3.1). For example, let us assume that we have an application that demands that nodes send 50 bytes long packets. If we implement the scheme where the size of the GTSs is fixed based on the node's bandwidth requirement [17], then we can accommodate transmissions of up to 33 nodes in CFP. If, however, this arrangement does not suit us, we can increase the superframe duration such that more nodes can be accommodated. However, improving GTS utilization can be futile if nodes are unable to access them as their requests did not reach the base station due to issues with channel contention. Therefore, improving the utilization of GTS alone does not help unless nodes can access these resources. Using the Device Registration algorithm, we can ensure that multiple nodes are able to access the resources efficiently, thereby making efficient use of the CFP resources that have been made available.

As explained in Section 4.1, we have implemented our proposed algorithm with minor parameter modifications in the existing MAC of IEEE 802.15.4. Our implementation follows the basic framework of beacon enabled mode of communication described in the standard. We have not proposed a new MAC for improving the real-time support using IEEE 802.15.4. This way, our proposed algorithm can be adopted by the existing IEEE 802.15.4 network without going through an extensive upgrade.

The scalability analysis for the proposed algorithm can be seen in Fig. 4. These results show that the probability of successful registration decreases when the number of active nodes in the network increases. Even in such cases, we cannot go back to IEEE 802.15.4 with standard CAP and CFP, as can be seen from Fig. 4. Success probabilities obtained when slotted CSMA/CA is employed for transmission of registration requests are worse than those obtained using the proposed algorithm. Channel assessment and backoffs suggested by the standard makes it unsuitable for supporting real-time guarantees. When the number of nodes requiring real-time guarantees for data transmission increases, it is better to employ a deterministic manner of scheduling the transmissions rather than relying on a probabilistic approach for channel access. Therefore, in such situations, it should be advisable to deploy a network supporting IEEE 802.15.4e-TSCH, which can easily support traffic that requires real-time guarantees. However, this would mean a daunting task of upgrading all the existing IEEE 802.15.4 networks to IEEE 802.15.4e-TSCH.

## 5. Sparse Beacon Advertisements for faster node associations in IEEE 802.15.4e-TSCH networks

From the previous sections, we have gathered that for large networks that require real-time guarantees, employing the IEEE 802.15.4e-TSCH standard is beneficial. However, as explained in Introduction 1, TSCH is a relatively new MAC standard and suffers from long joining times before a new node can join the network, especially if there are very few beacon advertisements. Hence, in this paper, we have attempted to tackle beacon scheduling in a network with a limited number of beacon advertisements.

To implement the proposed “Sparse Beacon Advertisement” for scheduling the beacons, we have considered a network that has $n$ nodes, one of which is the PAN coordinator, that starts the network and is the first one to begin the beacon advertisements. As the nodes join the network, they all begin advertising the beacons, thereby increasing the number of beacons being advertised in the network. We also assume that only one node wants to associate with the network at any given time. The ‘new’, unsynchronized node can join the network by receiving a beacon from any advertising node. Once a beacon is received, the node can synchronize with the network and start transmitting its own beacon. We have not applied any topology restriction over the network, as TSCH can handle any topology. Also, we have not paid attention to data transmission and have focused only on the control handling involved while bootstrapping the network and establishing the synchronization over the network.

In our proposed algorithm, the beacons are not transmitted on all the channels that have been configured for data transmission. Instead, we restrict the transmission of beacons on only a subset of these configured channels. Before a ‘new’ node is deployed to join the network, details about basic parameters such as node IDs, security keys, slot timing information, the total number of channels being used for communication, *etc.*, must be configured into the node. Along with these parameters, we can also configure the channels that are being used for beacon advertisements. According to the standard, the first slot of every slotframe is reserved for the transmission of beacons. Hence, as shown in Fig. 5, beacons can be transmitted only on the slots depicted in green.

Even though the number of channels for beacon transmissions is restricted, there is no need to implement a separate truth table that elaborates the channel hopping sequence for the beacons. Instead, a single truth table can be implemented to describe the channel hopping sequence of beacons as well as data transmissions. Assuming that a total of $N_d$ channels are being used for data transmission, the channel hopping sequence will have $N_d$ entries. Channels offsets $[0, N_d - 1]$ can be mapped to their respective entries in this hopping sequence using Eq. (1) with $N_c = N_d$. If out of $N_d$ channels, $N_b$ channels are used for beacon transmissions, the channel offsets $[0, N_b - 1]$ can be used for beacon transmission. These offsets can be mapped to the first $N_b$ entries of the same channel hopping sequence using Eq. (1) with $N_c = N_b$.

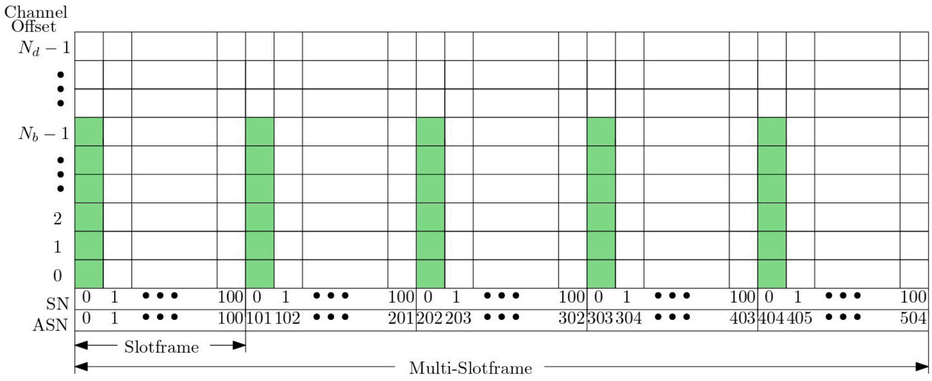


**Fig. 5.** Slotframe structure for the TSCH setup. There are 101 slots in a slotframe and a multi-slotframe consists of 5 slotframes. There are a total of $N_d$ channels for data transmission and out of these, only $N_b$ channels are used for beacon transmission. SN is the Slot Number associated with every slot in a slotframe and ASN denotes the Absolute Slot Number.

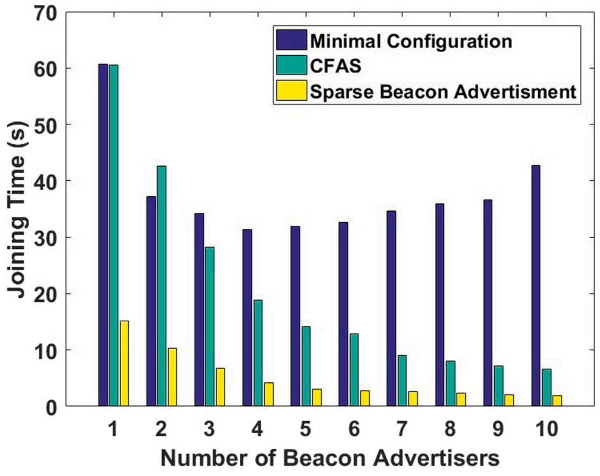


**Fig. 6.** Comparing joining times as obtained with the different beacon scheduling algorithms as the number of beacon advertisers are varied in the TSCH network. Beacon advertisements are restricted on 4 channels whereas data communication can go on all 16 channels.

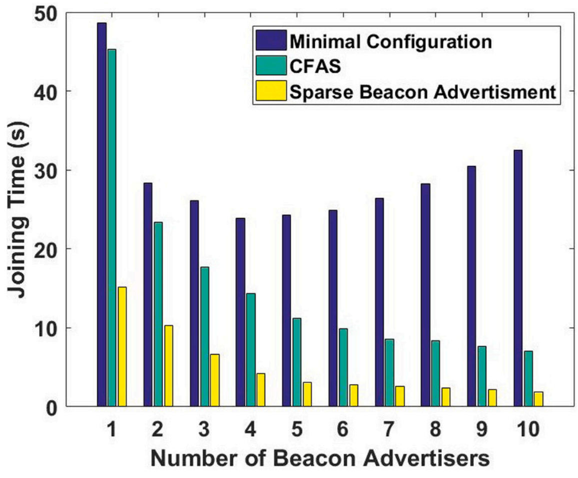


**Fig. 7.** Comparing joining times as obtained with the different beacon scheduling algorithms as the number of beacon advertisers are varied in the TSCH network. Beacon advertisements are restricted on 4 channels whereas data communication can go on 12 channels.

### *5.1. Sparse Beacon Advertisement – Simulation study*

To evaluate the performance of the proposed "Sparse Beacon Advertisement", we compare it with the Minimal configuration [21] and the CFAS [31]. We have chosen the Minimal Configuration as it is the most basic configuration for beacon scheduling and CFAS as it claims to outperform most of the beacon scheduling algorithms.

We have considered a TSCH setup that has 101 slots, each 15 ms long, in every slotframe. Five such slotframes constitute a multi-slotframe (Fig. 5), and every node transmits a beacon once every multi-slotframe. The beacon transmission schedule repeats every multi-slotframe, and hence a node should transmit a beacon every 505 slots. This structure was chosen as it has been used in multiple literature works [27,31]. Keeping the structure similar to other research works makes it straightforward for us to compare our proposal with theirs. For this setup, we have simulated Minimal, CFAS, and "Sparse Beacon Advertisement" algorithms using MATLAB and have compared the average joining times encountered with each of these algorithms when the number of beacon advertisers in the network is varied. The average joining time is calculated by running 150,000 iterations. In our simulations, the node can wake up anytime during the first multi-slotframe of the simulation. It is assumed that the rest of the network is synchronized, and every node transmits its beacon at its respective slot from the beginning of the simulations. The time it takes for a new node to receive a beacon is calculated as the difference between the time instant a beacon was received and the time instant the node woke up.

To implement our proposed algorithm, we have restricted the number of channels over which the beacon can be transmitted to be only 4. Therefore, channel offsets $[0, 1, 2, 3]$ can be used for assigning links for beacon transmission. These channel offsets will be translated to physical channel based on Eq. (1) with $N_c = N_b = 4$. Accordingly, the truth table will always pick one of the first four channels from the channel hopping sequence. Also, in a multi-slotframe structure, the slots corresponding to the ASN values $[0, 101, 202, 303, 404]$ can be used for transmitting beacons (Fig. 5). Alongside, we have also varied the total number of channels configured for data transmission, and we configured a total of 16, 12, and 8 channels for data transmissions.

Fig. 6 shows a comparison between the joining times observed with Minimal configuration, CFAS, and the proposed algorithm when a total of 16 channels are configured for communication and beacon transmissions are restricted on only 4 of them. Similarly, Fig. 7 and Fig. 8 depict the comparison when a total of only 12 and 8 channels are configured for data transmission, respectively. We can observe that even though CFAS performs better than the Minimal configuration when there are multiple advertisers, it offers limited improvement in the joining times when there are very few beacon advertisements. On the other hand, the proposed algorithm performs much better than

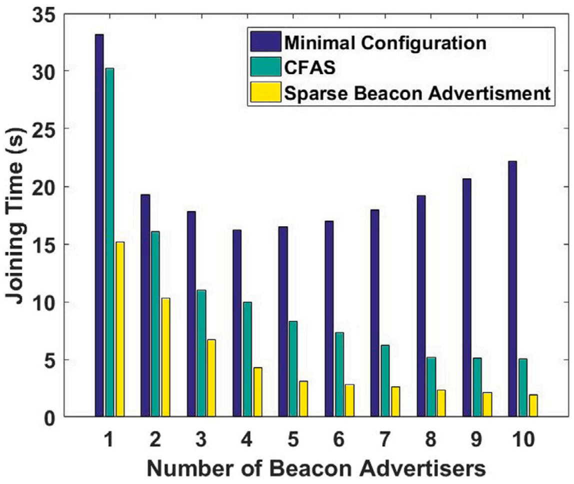


**Fig. 8.** Comparing joining times as obtained with the different beacon scheduling algorithms as the number of beacon advertisers are varied in the TSCH network. Beacon advertisements are restricted on 4 channels whereas data communication can go on 8 channels.

the other two beacon scheduling algorithms, especially when there are fewer beacon advertisers.

### 5.2. Implementating a IEEE 802.15.4e-TSCH network

We have implemented a TSCH network using the $nRF52840$ SoC from Nordic Semiconductor [8] to validate the performance of our algorithm. The output transmission power of the SoC can be varied from −20 dBm to +8 dBm, and its radio sensitivity is −100 dBm at 250 kbps. These boards provide support for multiple low power communication protocols, including the IEEE 802.15.4 standard, and can be used to implement the MAC for IEEE 802.15.4e-TSCH as the physical layer requirements for both standards are the same. We have implemented the process of beacon advertisements in TSCH using the parameters of the beacon enabled mode of IEEE 802.15.4 MAC. Since IEEE 802.15.4 already has provisions for beacon transmissions, we added the concept of channel hopping to the same to emulate the way beacons will be transmitted in a TSCH network. Our implementation concentrates only on beacon advertisements and does not focus on data transmission or network topology formation. This has enabled us to validate the performance of our algorithm without implementing a complete TSCH MAC on our node. We have emulated the same TSCH setup, including the slotframe and multi-slotframe structure, which was used for simulations (Fig. 5).

Another objective of implementing the TSCH network was to measure the energy consumed by a node while it waits for an EB. Existing implementations of TSCH are built on top of Contiki OS, which might not give a proper indication of energy consumption. At any given time, the OS might be running multiple background tasks that will then add to the energy consumption of the node. Hence, we decided to implement the beacon advertisements on IEEE 802.15.4 compatible nodes to better understand the node's energy consumption.

#### 5.2.1. Details of implementation

To implement TSCH slots, beacon transmissions, and the sleep cycles for the nodes, we have taken advantage of the superframe structure already available in the beacon enabled mode of IEEE 802.15.4. For every node, the active period of the beacon interval is restricted to the duration of a single TSCH slot. The inactive period of the beacon interval is calculated depending upon the period after which the beacon needs to be transmitted again.

The active period, also called the superframe duration, of the beacon interval, can be calculated as $SD = aBaseSuperframeDuration \times 2^{SO} symbols$, where $0 \leq SO \leq 14$ [10]. $aBaseSuperframeDuration$ is 960 symbols long, and if we fix $SO = 0$, we get $SD = 15.36$ ms at 250 kbps. This is the closest we can get to the 15 ms slots of the TSCH setup. In this manner, we can achieve uniform 15.36 ms slots for all the nodes. Next, we know that the node should sleep for a duration of 505 slots, *i.e.*, $7.75s$. This can be achieved by setting an appropriate beacon interval using $BI = aBaseSuperframeDuration \times 2^{BO} symbols$, where $0 \leq BO \leq 14$. If we set $BO = 8$, we can get $BI = 7.86s$ at 250 kbps. With this setting, the node wakes up every $7.86s$ for a duration of 15.36 ms to transmit the beacon. The beacon contains information about the current ASN value, the sleep duration of the node, and information regarding channel hopping so that the new node can follow the beacon once it receives it.

#### 5.2.2. Bootstrapping the network

To bootstrap the network, one of the boards is configured as the PAN coordinator, *i.e.*, *Node* 0, which starts the network by advertising beacons. Another node, *i.e.*, *Node* 1, when it wishes to join the network, wakes up and randomly chooses a channel among all the available channels to wait for the reception of the beacon. After the beacon is received, the node extracts the synchronization information from the beacon and follows the beacon for some time to ensure that synchronization has been established. Once this is ensured, another node (*Node* 2) is brought into picture which can now get the synchronization information from the beacon of either *Node* 0 or *Node* 1. When the beacon is received, the node gets synchronized and starts transmitting its beacon. In this manner, a node is added to the network, one at a time, and depending upon which node's beacon is used as the source of synchronization information, its initial hop distance from the PAN coordinator is established. For instance, let us consider the situation where both *Node* 0 and *Node* 1 are transmitting beacons and *Node* 2 is waiting to receive a beacon. In case, *Node* 2 receives the beacon from *Node* 0, both *Node* 1 and *Node* 2 are one hop away from *Node* 0. However, if *Node* 2 receives the beacon from *Node* 1, *Node* 2 becomes two hops away from *Node* 0. The initial topology of the network is inconsequential when considering the time it takes for a new node to join the network. Hence, we have not paid attention to identify the node whose beacon is received by the new while joining the network. We have focused on the time it takes for a new node to receive the beacon and get synchronized with the other nodes in the network. Moreover, the topology of the network is defined by the routing algorithm used. Typically, in TSCH networks, RPL (Routing Protocol for Low Power and Lossy Networks) is used, a dynamic protocol that eventually leads to an efficient topology.

### 5.3. Sparse Beacon Advertisement – Experimental study

Fig. 9 shows the experimental testbed. In our experiments, we set up a network consisting 4 nodes, all of which were synchronized at the end of the bootstrap. This initial bootstrap synchronization is based on beacons, and we have implemented the Minimal configuration and the proposed Sparse Beacon Advertisement for beacon scheduling. For the experiments, we have assumed that all 16 channels are available for communication. Irrespective of the beacon scheduling algorithm, the network is bootstrapped, as described in Section 5.2.2. We start with the PAN coordinator (*Node* 0), and it starts advertising beacons with a period of 7.86 s (Section 5.2.1). A new node joins the network when it receives a beacon. Once a node has gathered the synchronization information from the beacon, we make the node follow the beacon for at least 5 multi-slotframes. This ensures that the new node is fully synchronized and can now transmit its beacon. This way, we start with a single node and keep adding one node at a time to the network.

To ensure that beacons are adequate to maintain synchronization in the network, we have conducted a simple experiment. We program a node to broadcast beacons at regular intervals over all 16 channels. Another node is placed in the network, which at first listens for the

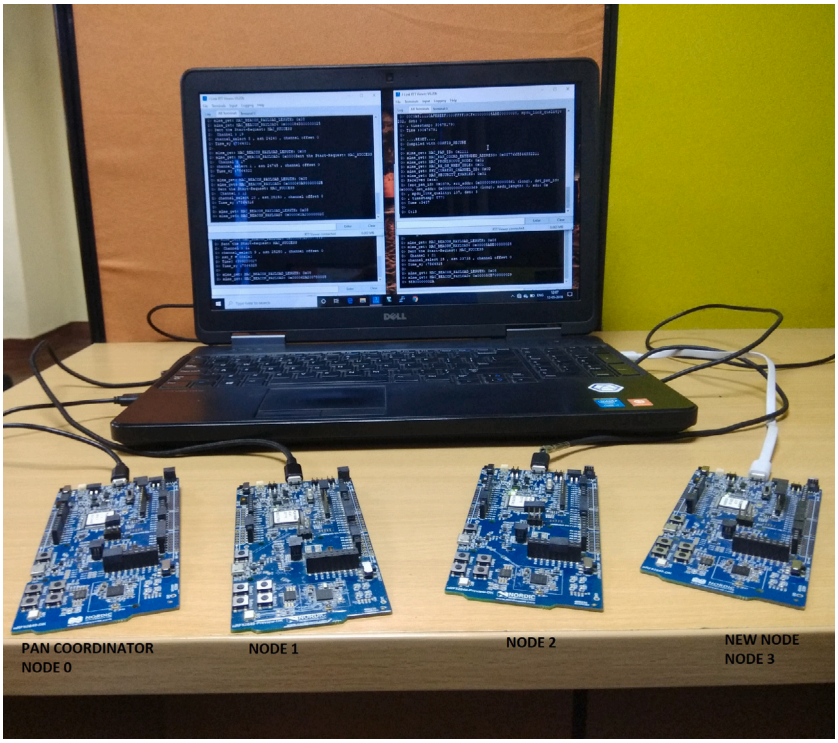


**Fig. 9.** Testbed for implementing the IEEE 802.15.4e-TSCH network. The figure shows that the network consists for 4 nodes, one of them being a PAN Coordinator (*Node* 0) and one of them being a new, unsynchronized node (*Node* 3).

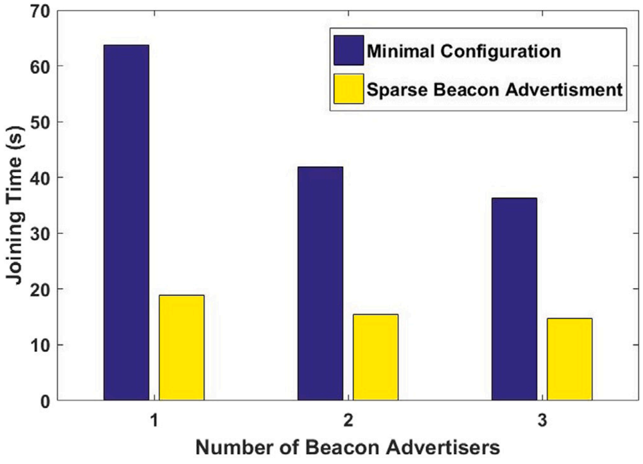


**Fig. 10.** Comparing joining times as obtained with the different beacon scheduling algorithms as the number of beacon advertisers are varied in the testbed implemented for TSCH.

**Table 2**
Comparing the energy consumed by a node while waiting for a beacon. The beacons are scheduled using the Minimal Configuration and the proposed Sparse Beacon Advertisement. The number of beacon advertisers are varied from 1 to 3.

| | Energy expenditure (J) | | |
|---|---|---|---|
| Number of advertisers | Minimal configuration | Sparse Beacon Advertisement | Percentage improvement |
| 1 | 3.66 | 1.08 | 70.5 |
| 2 | 2.40 | 0.88 | 63.3 |
| 3 | 2.08 | 0.84 | 59.6 |

beacon and once it receives the beacon, it attempts to follow the beacon for as long as it can. We ran multiple instances of these experiments for extended periods (more than 8 hours). We observed that in every instance, the node was able to follow the beacon across all the different frequencies and did not miss the beacon even once after receiving the beacon for the first time.

We have then evaluated the performance of both the beacon scheduling algorithms in terms of the observed average joining times and energy expenditure of a new node. More than 1000 iterations of experiments are conducted to obtain the average value for the joining times and the corresponding energy expenditure. Fig. 10 compares the average joining times obtained with the beacon being scheduled using Minimal Configuration and the proposed algorithm. The number of beacon advertisers is varied from 1 to 3. We can see that the experimental results closely follow the results obtained using simulations.

Next, we discuss the energy expenditure of a new, unsynchronized node while it waits to receive a beacon. Since we have conducted experiments for Minimal Configuration and our proposed Sparse Beacon Advertisement algorithm on the $nRF52840$ boards, we consider average energy spent by the nodes before they get to join the network for only these two algorithms. The energy consumed by a node while waiting for a beacon is a function of the time it takes for the node to receive the beacon. Till the time a node receives a beacon, it needs to keep its radio continuously ON. The total energy consumed before a node can join the network is a function of the joining time, the current drawn during this time and the voltage that the node was operating at. Therefore, if a node has to wait for a longer duration before it can join the network, it will consume more energy. Table 2 gives the details. We can observe that the energy consumed is reduced by at least 60% when the proposed Sparse Beacon Advertisement is implemented.

### *5.4. Sparse Beacon Advertisement – Discussion and insights*

Once we have implemented and evaluated the proposed Sparse Beacon Advertisement for beacon scheduling in IEEE 802.15.4e -TSCH network, several inferences can be drawn.

The first thing that we notice is that reducing the number of channels used for beacon transmission reduces the joining time. With fewer channels, the time it takes for a node to hop across the channels and return on a particular channel is less than the scenario when the node has to hop across a larger set of channels. Therefore, a "new" node waiting at a particular channel will encounter a beacon sooner rather than later if the set of channels on which beacons are being transmitted is restricted. This can be seen from the simulations results, as depicted in Fig. 6, Fig. 7, and Fig. 8, and the experimental results, as illustrated in Fig. 10. From the simulation study, it can be seen that there are significant improvements in joining time when there are very few beacons being advertised in the network. The joining time shows at least 73% reduction when the proposed algorithm is employed instead of transmitting beacons on each one of the 16 channels. Similarly, a drop of at least 70% is observed when only 4 out of a total of 12 channels are configured for beacon advertisements. When a total of 8 channels are configured to be used for data transmission, a reduction of at least 62% can be observed when only 4 of them are being used for beacon advertisements. Also, as the nodes need to wait for a shorter duration before they encounter a beacon, they do not need to keep their radios continuously ON for long duration looking for beacons. This leads to a reduction in the energy consumed by a new node, as tabulated in Table 2.

Even though we have restricted the number of channels for beacon transmission, data can continue to be transmitted on all the available channels. Despite this, we need not implement a separate channel hopping sequence for beacon and data transmissions, as explained in Section 5. This way, we do not have any additional memory requirement for storing a separate channel hopping sequence with the node.

Another significant inference that can be drawn from the obtained results is that we can now have lower joining times even with few beacons in the network. Therefore, moving forward, we can establish that even for an extensive network, all nodes need not be advertising beacons for quick joining times. It will suffice that a small subset of nodes advertises beacons. Therefore, we can also avoid delays in joining experienced by nodes due to beacon collisions that are mostly encountered when a large number of beacons are advertised in a network. Additionally, for a network with energy-constrained nodes, this small subset needs not be fixed and can be chosen based on the energy available with the different nodes. Another method of selecting the nodes that advertise beacons can be based on the number of neighbors. Also, the number of nodes responsible for advertising the beacons can be decided based on the expanse of the network.

We observed that when our proposed algorithm is implemented, there are minor mismatches between the experimental results and the corresponding simulation results. One of the reasons might be that the total number of experimental iterations run to calculate the average joining times is considerably less than the iterations run during the simulation study. Another reason is that the parameters used for setting up the TSCH structure are slightly different in simulations and experiments. During simulations, consecutive beacons transmitted by a node are 505 slots apart. Each of these slots is 15 ms long, and hence the duration between consecutive beacons is 7.57 s. However, as explained in Section 5.2.1, a single TSCH slot is now 15.36 ms, and the duration between two consecutive beacons is 7.86 s. This amounts to 512 TSCH slots between consecutive beacons. Therefore, we can see that while performing the experiments, the beacons are slightly farther apart than they are in simulations. This amounts to the joining times obtained using experiments being larger than the ones obtained using simulations.

An important point to notice is that it is essential that the new node is aware of the channels used for beacon transmissions, and it chooses to listen for the beacon on one of these channels itself. Otherwise, there is a non-zero probability that the node can select a channel over which there are no beacon transmission. Considering that there are a total of $N_c$ channels, out of which beacon transmissions are allowed on $N_b$ channels, this probability is $(1 - N_b/N_c)$. In such a situation, the node can stay on the channel without ever receiving any beacon, which is highly undesirable. However, as mentioned in Section 5, we can update the list of channels used for beacon advertisements in the network while configuring a node before deploying it in the network. This will avoid the situation where a node can potentially pick a channel that is not being used by the network for any purposes.

It should be noted that the network topology is not decided by the manner nodes "join" the network. A node is considered to have "joined" the network once it receives the synchronization information about the network. This can be received from any node in the network. Routing protocols are responsible for the formation of network topology. In TSCH networks, routing is accomplished using RPL. The topology constructed by RPL for data transmissions is based on a Destination Oriented Directed Acyclic Graph (DODAG). The topology is built around a "root" node, which usually is the PAN coordinator. The root node transmits DODAG Information Objects (DIO) messages that trigger the topology formation process. These DIO messages are transmitted from node to node in the network, which assists the nodes discover their neighbors and potential parents [36]. The nodes can then choose their respective parents to whom they will transmit their data. Moreover, RPL is a dynamic routing protocol. Even if a node chooses a parent initially, it can switch to a different parent, if it discovers that routing through this latter node will result in better reliability in data delivery as compared to the earlier route.

With the help of the proposed algorithm, we can not only reduce the time a node takes to synchronize with the network but also lead to a faster topology formation. In the usual case, the first timeslot of every slotframe is reserved for transmission of control packets — beacons and DIO messages. However, the priority of these DIO messages is lower than the beacons and hence, at times, transmission of these packets is delayed due to the ongoing beacon transmissions. This can be entirely avoided with our proposal as with the beacons restricted to few channels, links will be available over which routing packets can be transmitted. This, in turn, will lead to faster routing handshakes and subsequent data transmission.

## 6. Conclusion

In this work, we have studied networks where data transmissions have associated deadlines. We have proposed methods that facilitates quick access to the network resources so that data transmission deadlines are not compromised. We have worked with IEEE 802.15.4 and IEEE 802.15.4e-TSCH networks. We have proposed a "Device Registration" algorithm using which the existing deployments of IEEE 802.15.4 networks can be made suitable for handling time-sensitive sporadic data. The algorithm implements minor parameter modifications within the existing MAC framework of IEEE 802.15.4. The results from the implementation of the proposed algorithm achieve significantly higher success probabilities with which a node can transmit its registration request for accessing GTS resources to the base station. To support a network with a large number of nodes that require timely data delivery, employing IEEE 802.15.4e-TSCH proves to be beneficial. For these networks, we have proposed "Sparse Beacon Advertisement", a beacon scheduling algorithm that aims at reducing the joining times even when there are few beacon advertisements so that the node can participate in the network at the earliest. This was achieved by restricting the number of channels over which beacons can be transmitted. The evaluation results show significant reductions in the joining times. Both the proposed algorithms are evaluated rigorously with experiments and MATLAB simulations. Energy expenditures are also studied for both these algorithms and they reduce the energy consumed by a node while accessing the network resources for communication.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] Renesas Electronic Corporation, SOTB™ process technology, 2019, Online; Accessed20August2019.

[2] Atmosic Technologies, Atmosic - connected forever, anywhere, 2019, Online; accessed20-August-2019.

[3] S. Bluetooth, Bluetooth architecture & terminology overview, in: Bluetooth Specification Version 4.2, Vol 0, 2014.

[4] Silicon Laboratories, Safer, smarter homes start with Z-wave, 2019, Online; Accessed20August2019.

[5] IEEE Standards Association, et al., IEEE Standard for local and metropolitan area networks part 15.4: Low-rate wireless personal area networks (LR-WPANs), amendment 1: Mac sublayer, Technical Report IEEE Std 802.15. 4e, 2012.

[6] H. Kurunathan, R. Severino, A. Koubaa, E. Tovar, IEEE 802.15.4e in a nutshell: Survey and performance evaluation, IEEE Commun. Surv. Tutor. 20 (3) (2018) 1989–2010, http://dx.doi.org/10.1109/COMST.2018.2800898.

[7] B. Carroll, Zigbee leads the wireless mesh sensor network market, 2018, Online; Accessed20August2019.

[8] Nordic Semiconductor, nRF52840 High-end Multiprotocol Bluetooth 5 SoC Supporting: Bluetooth 5/Bluetooth mesh/Thread/Zigbee/802.15.4/ANT/2.4 GHz.

[9] T. Watteyne, M. Palatella, L. Grieco, RFC 7554: Using IEEE 802.15. 4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement, in: Internet Engineering Task Force (IETF) Informational Document; Internet Engineering Task Force: Fremont, CA, USA, 2015.

[10] M. Khanafer, M. Guennoun, H.T. Mouftah, A survey of beacon-enabled IEEE 802.15. 4 MAC protocols in wireless sensor networks, IEEE Commun. Surveys Tutor. 16 (2) (2014) 856–876, http://dx.doi.org/10.1109/SURV.2013.112613.00094.

[11] M. Takaffoli, E. Elmallah, W. Moussa, Scheduled access using the IEEE 802.15.4 guaranteed time slots, in: Communications (ICC), 2010 IEEE International Conference on, IEEE, 2010, pp. 1–5, http://dx.doi.org/10.1109/ICC.2010.5502148.

[12] M. Kim, C.-H. Kang, Priority-based service-differentiation scheme for IEEE 802.15.4 sensor networks in nonsaturation environments, IEEE Trans. Veh. Technol. 59 (7) (2010) 3524–3535, http://dx.doi.org/10.1109/TVT.2010.2046757.

[13] P. Park, C. Fischione, K.H. Johansson, Performance analysis of GTS allocation in beacon-enabled ieee 802.15.4, in: 2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, IEEE, 2009, pp. 1–9, http://dx.doi.org/10.1109/SAHCN.2009.5168958.

[14] A. Koubaa, M. Alves, E. Tovar, I-GAME: An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks, in: 18th Euromicro Conference on Real-Time Systems, ECRTS'06, IEEE, 2006, p. 10, http://dx.doi.org/10.1109/ECRTS.2006.13.

[15] L. Cheng, X. Zhang, A. Bourgeois, GTS Allocation scheme revisited, Electron. Lett. 43 (18) (2007) 1005–1006, http://dx.doi.org/10.1049/el:20071682.

[16] L. Cheng, A.G. Bourgeois, X. Zhang, A new GTS allocation scheme for IEEE 802.15. 4 networks with improved bandwidth utilization, in: 2007 International Symposium on Communications and Information Technologies, IEEE, 2007, pp. 1143–1148, http://dx.doi.org/10.1109/ISCIT.2007.4392189.

[17] Y. Zhan, Y. Xia, M. Anwar, GTS Size adaptation algorithm for IEEE 802.15. 4 wireless networks, Ad Hoc Netw. 37 (2016) 486–498, http://dx.doi.org/10.1016/j.adhoc.2015.09.012.

[18] A.N. Alvi, S. Khan, M.A. Javed, K. Konstantin, A.O. Almagrabi, A.K. Bashir, R. Nawaz, OGMAD: Optimal GTS-allocation mechanism for adaptive data requirements in IEEE 802.15. 4 based internet of things, IEEE Access 7 (2019) 170629–170639, http://dx.doi.org/10.1109/ACCESS.2019.2955544.

[19] T.N. Dinh, P.H. Ha, Resource management for improving performance of IEEE 802.15. 4-based home automated systems, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/GLOCOM.2018.8647804.

[20] T.N. Dinh, P.H. Ha, Advanced GTS scheduling in IEEE 802.15. 4 networks for industrial application, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2019, pp. 1–4, http://dx.doi.org/10.1109/CCNC.2019.8651809.

[21] X. Vilajosana, K. Pister, Minimal 6TiSCH Configuration-draft-ietf-6tisch-minimal-00, IETF, Fremont, CA, USA, 2013.

[22] R.C. Alves, C.B. Margi, IEEE 802.15. 4e TSCH mode performance analysis, in: 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems, MASS, IEEE, 2016, pp. 361–362, http://dx.doi.org/10.1109/MASS.2016.054.

[23] L. Wang, A. Reinhardt, A simulative study of network association delays in IEEE 802.15. 4e TSCH networks, in: 2017 IEEE 18th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2017, pp. 1–3, http://dx.doi.org/10.1109/WoWMoM.2017.7974347.

[24] D. De Guglielmo, A. Seghetti, G. Anastasi, M. Conti, A performance analysis of the network formation process in ieee 802.15.4e TSCH wireless sensor/actuator networks, in: 2014 IEEE Symposium on Computers and Communications, ISCC, 2014, pp. 1–6, http://dx.doi.org/10.1109/ISCC.2014.6912607.

[25] D. De Guglielmo, S. Brienza, G. Anastasi, A model-based beacon scheduling algorithm for IEEE 802.15. 4e TSCH networks, in: 2016 IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2016, pp. 1–9, http://dx.doi.org/10.1109/WoWMoM.2016.7523517.

[26] E. Vogli, G. Ribezzo, L.A. Grieco, G. Boggia, Fast join and synchronization schema in the IEEE 802.15. 4e MAC, in: 2015 IEEE Wireless Communications and Networking Conference Workshops, WCNCW, IEEE, 2015, pp. 85–90, http://dx.doi.org/10.1109/WCNCW.2015.7122534.

[27] E. Vogli, G. Ribezzo, L.A. Grieco, G. Boggia, Fast network joining algorithms in industrial IEEE 802.15. 4 deployments, Ad Hoc Netw. 69 (2018) 65–75, http://dx.doi.org/10.1016/j.adhoc.2017.10.013.

[28] I. Khoufi, P. Minet, E. Livolant, B. Rmili, Building an IEEE 802.15. 4e TSCH network, in: 2016 IEEE 35th International Performance Computing and Communications Conference, IPCCC, IEEE, 2016, pp. 1–2, http://dx.doi.org/10.1109/PCCC.2016.7820597.

[29] I. Khoufi, P. Minet, An enhanced deterministic beacon advertising algorithm for building TSCH networks, Ann. Telecommun. 73 (11–12) (2018) 745–757, http://dx.doi.org/10.1007/s12243-018-0638-3.

[30] A. Karalis, ATP: A fast joining technique for IEEE802. 15. 4-TSCH networks, in: 2018 IEEE 19th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2018, pp. 588–599, http://dx.doi.org/10.1109/WoWMoM.2018.8449759.

[31] A. Karalis, D. Zorbas, C. Douligeris, Collision-free advertisement scheduling for IEEE 802.15. 4-TSCH networks, Sensors 19 (8) (2019) 1789, http://dx.doi.org/10.3390/s19081789.

[32] K. Singh, T.V. Prabhakar, J. Kuri, Improving access to network resources in IoT networks, in: 2020 International Conference on COMmunication Systems NETworkS, COMSNETS, 2020, pp. 259–266, http://dx.doi.org/10.1109/COMSNETS48256.2020.9027349.

[33] Texas Instruments, CC2530 data sheet, 2009, [2013-01-05]. http://www.ti.com.cn/product/cn/cc2530.

[34] P.M. Checker, IEEE 802.15.4 CSMA-CA protocol (ZigBee), 2015, Online; accessed20-August-2019.

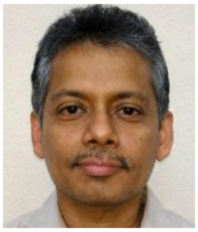
[35] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pp. 95–107, http://dx.doi.org/10.1145/1031495.1031508.

[36] C. Vallati, S. Brienza, G. Anastasi, S.K. Das, Improving network formation in 6TiSCH networks, IEEE Trans. Mob. Comput. 18 (1) (2018) 98–110, http://dx.doi.org/10.1109/TMC.2018.2828835.

**Kaumudi Singh** has a B.Tech. in Electronics and Communication Engineering from NERIST, Itanagar. Then she joined the M.Tech program in the Department of Electrical and Electronics Engineering at Amrita School of Engineering, Coimbatore. Currently she is pursuing her Ph.D. in the Department of Electronic System Engineering at Indian Institute of Science, Bengaluru. She is interested in the research areas related to MAC protocols for low power, low data rate networks, indoor localization and tracking using RFID communication.

**Vasanth Kumar T** received the Bachelor's Degree in Electronics and Communication Engineering from RNS Institute of Technology, Bangalore, India in 2017. He is currently working as Project Assistant with Department of Electronic Systems Engineering at Indian Institute of Science (IISc), Bangalore since 2017. His current research interests are in applications of Internet of Things, Bluetooth Low Energy (BLE), Communication Protocols and Sensor Networks.

**Prabhakar T V** is a Principal Research Scientist at IISc. His main area of research is Networked Embedded Systems and IoT. The focus is on hardware system design, power management algorithms, IoT data management and technologies to build smart IoT applications. Some recent work is related to RFID localization, PUF based device ID, Fast MAC access and Tactile networking, Fast and Slow loop algorithms for IoT data management, edge intelligence systems, application of powerline modems for smart sensing and machine learning for speech to text systems. His laboratory has published over 70 papers in IEEE and ACM conferences. Prabhakar holds M.Sc. (Physics), M.Sc. (Engg) and Ph.D. degrees.



**Joy Kuri** has a B.E. in Electronics and Telecommunication Engineering from Jadavpur University, Kolkata. After a year in the industry, he joined the M.E. program in the Department of Electrical Communication Engineering at the Indian Institute of Science. He went on to receive a Ph.D. from the same department at IISc in 1995. Subsequently, he spent two years at Ecole Polytechnique, University of Montreal, Canada and one and a half years in INRS-Telecommunications, University of Quebec, Canada as a Research Associate. Since 1999 he has been with the Centre for Electronics Design and Technology, Indian Institute of Science, where he is currently a Professor. He is a co-author of the books ‘‘Communication Networking: An Analytical Approach’’ and ‘‘Wireless Networking,’’ both published by Morgan Kaufmann, an imprint of Elsevier. He has over 75 publications in international journals and conferences. His research and teaching interests are in the areas of modeling, analysis and control of communication networks and stochastic systems.