



HAL
open science

FAN: Fast and Active Network Formation in IEEE 802.15.4 TSCH Networks

Mohamed Mohamadi, Badis Djamaa, Mustapha Reda Senouci, Abdelhamid
Mellouk

► **To cite this version:**

Mohamed Mohamadi, Badis Djamaa, Mustapha Reda Senouci, Abdelhamid Mellouk. FAN: Fast and Active Network Formation in IEEE 802.15.4 TSCH Networks. Journal of Network and Computer Applications (JNCA), 2021, 183-184, pp.103026. 10.1016/j.jnca.2021.103026 . hal-04344343

HAL Id: hal-04344343

<https://hal.science/hal-04344343v1>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

FAN: Fast and Active Network Formation in IEEE 802.15.4 TSCH Networks

Mohamed Mohamadi, Badis Djamaa, Mustapha Reda Senouci, Abdelhamid Mellouk*

Distributed & Complex Systems Lab., Ecole Militaire Polytechnique, Algiers, Algeria

LiSSi Laboratory, University of Paris-Est, France

Abstract

The IEEE 802.15.4 TSCH standard is designed to improve the reliability, timeliness, and energy efficiency of short-range wireless communications in industrial applications. TSCH relies upon a network formation process to efficiently create and maintain a synchronized reliable mesh network. The standard adopts a purely passive scan mechanism enabling joining nodes to listen to periodic Enhanced Beacons (EBs) in order to locate a TSCH network and associate to it. Nevertheless, the standard defines neither the advertisement strategy nor the rate at which the EBs should be sent. Consequently, long (re)association times might be observed, which postpones the proper functioning of the network and might have devastating consequences on the industrial application. In this paper, we propose a Fast and Active Network formation scheme (FAN) that leans on active scan procedures and Trickle beacon advertising strategies to accelerate the (re)association process. To do so, FAN equips joiners with a collision avoidance mechanism and allows them to initiate EB requests, on free channels, to trigger EBs. FAN is implemented in Contiki and evaluated through extensive realistic simulations and public testbed experiments. Obtained results assert the robustness of FAN compared to state-of-the-art solutions in terms of association times and overhead.

Keywords: Industrial Internet of Things, TSCH, 6TiSCH, Network Formation, Active Scan, Trickle Algorithm.

1. Introduction

With the introduction of the Time Slotted Channel Hopping (TSCH) mechanism as part of the IEEE 802.15.4e amendment in 2012 [1], a new generation

*Corresponding author

Email addresses: med_mohamadi@yahoo.com (Mohamed Mohamadi),
badis.djamaa@gmail.com (Badis Djamaa), mrsenouci@gmail.com (Mustapha Reda Senouci),
mellouk@u-pec.fr (Abdelhamid Mellouk)

Preprint submitted to Elsevier

January 27, 2021

of ultra low-power, real-time, and reliable networking solutions became feasible. This, in turn, drove a movement towards an economically practical wireless solution for critical industrial monitoring applications. When used with an embedded IPv6 networking stack for low-power wireless sensor networks, TSCH applications create unprecedented business opportunities giving birth to the so-called Industrial Internet of Things (IIoT).

To achieve reliability and timeliness, TSCH uses deterministic time-slotted channel access solutions while hopping between 16 communication channels to respond to industrial environments' harsh conditions. To ensure low-power consumption, TSCH enables nodes to sleep most of the time while allowing communication when needed. To do so, nodes' clocks are synchronized, and the time is slotted into periods of sleep and wake-ups.

Enabling communication in a TSCH network requires efficient network formation along with optimized scheduling mechanisms to decide when two nodes can communicate and over which channel. The former is the key element in constructing the network and is the subject of this paper. It is mainly realized by nodes making passive scans for advertised EBs [2]. This approach presents many drawbacks, including long joining time, long scanning periods, delaying network operations, and unbalanced energy consumption [3].

To address the aforementioned issues, we introduce in this paper a Fast and Active Network formation scheme (FAN) that combines active scan procedures with Trickle beacon advertising strategies. The aim is to accelerate the network formation process and support a rapid association of new nodes, while reducing network overhead. The main contributions of this paper can be summed up as follows:

- [Adaptive Trickle based advertisement strategy of EBs;](#)
- [Enhanced active scan procedure for TSCH networks with Trickle Enhanced Beacon Requests \(EBRs\);](#)
- [A collision avoidance mechanism to detect and avoid collisions between EBRs and a running TSCH schedule;](#)
- [Extensive evaluations and comparisons with the state-of-the art solutions under realistic simulations and public testbed experiments in dedicated IIoT platforms.](#)

The remainder of this paper is organized as follows. Section 2 gives a brief description of the TSCH MAC mode along with the network formation process and discusses related research. Section 3 provides the driving mechanisms of our proposal as well as an overview of its global operations, while Section 4 details and explains the main components of FAN. This is followed by extensive performance evaluations in both extensive realistic simulations in Section 5, and testbed experiments in Section 6. The paper ends in Section 7 with some concluding remarks and ideas for future work.

2. Background & Related Work

2.1. Background on TSCH Network Formation

As stated above, TSCH is designed to support the requirements of industrial IoT applications. In TSCH, time is sliced up into timeslots grouped into one or more slotframes, which repeat over time. Communication between two nodes is always performed on a link that is identified by $[TimeOffset, ChannelOffset]$. Fig. 1 presents a slotframe (SF) composed of 10 timeslots and 5 available channel offsets (N_{ch}) that repeats over time (2 times in this figure). Colored cells refer to scheduled links. The same link has different sending channels at different cycles of the slotframe since the sending channel Ch is calculated as follows:

$$Ch = T[(ASN + ChannelOffset) \% N_{ch}] \quad (1)$$

Where the Absolute Slot Number (ASN) is a timeslot counter, it serves as a reference time for new devices. As shown in Fig. 1, the ASN is initialized to 0; then it increments continuously by one at each timeslot. It is encoded as a 5-byte number so that it can increment for hundreds of years. The function T is a look-up table to select a sending channel.

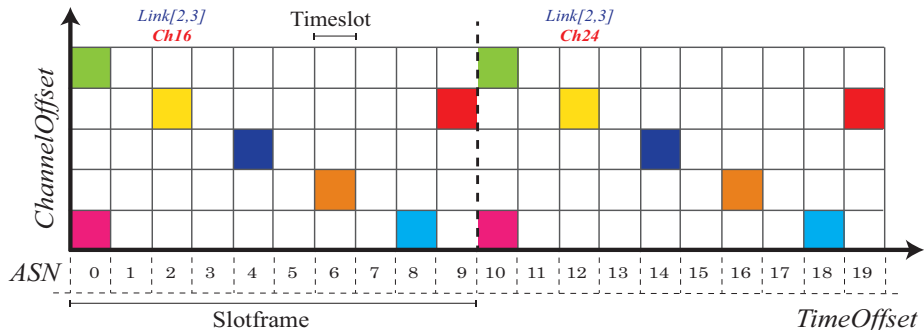


Figure 1: Example of TSCH schedule

To construct a TSCH network, a coordinator announces the network's presence by sending periodically, in broadcast, Enhanced Beacons (EBs) containing the network configuration parameters such as ASN, number of available channels, timeslot length, and the current schedule. The EB advertising period (t_{EB}) is a tunable parameter that trades-off the network formation time, network capacity, and energy consumption [4].

When a new node wants to join the network, it keeps its radio transceiver always-on and begins scanning passively and randomly for incoming EBs on all available channels. Once a joining node falls on the same channel as an EB, it receives the EB, reads the contained network parameters, and associates to the TSCH network. [At this point, the joiner switches into TSCH mode, becomes an advertiser, and periodically broadcasts EBs to announce the network's presence.](#)

This passive scan-based joining process continues until all nodes join the network. Hence, the advertisers keep sending periodically EBs along the network lifetime to announce the presence of the network and allow (re)synchronization.

The aforementioned standard network formation process depends on a multitude of parameters, including the slotframe size, the EB advertising period, network density, and the number of available channels, and may take a long time and/or generate high overhead [5]. This might become a serious problem knowing that the process does not only show up during initial deployment but can also occur during network operations due to node churn, connectivity loss, mobility, battery replacement, and unexpected disassociation from the network [6]. In addition, within this process, nodes far from the coordinator may lose a considerable amount of energy before joining the TSCH network, which might impact their lifetime [7].

The above-mentioned network formation issues attracted the research community and have been studied widely in [8, 9, 10]. Accordingly, enhancing TSCH network formation became an active research topic. It was also placed at the head-list of TSCH issues identified by the IPv6 over TSCH (6TiSCH) working group [11] at the Internet Engineering Task Force. The following section reviews the most relevant solutions that have been proposed in this context.

2.2. Related Work

Recently, a plethora of solutions have been proposed to optimize the network formation process in TSCH networks. As the standard does not define any advertisement strategy of EBs, the authors of [5] proposed and evaluated a random-based advertisement algorithm with a simple mechanism to reduce the collision probability. This study reveals that the joining time depends on several factors. Authors in [3] and [12] proposed a set of mechanisms to speed up the joining phase: once exploiting all the available timeslots randomly and once exploiting all the available channel frequencies. Each mechanism has been analytically modeled and evaluated through experiments.

In [13], the authors proposed a reliable lightweight joining scheme that divides the SF into two parts, including advertisement slots, which are reserved for broadcasting EBs, and the communication slots that are only used to exchange data packets. An extension of this work, based on fuzzy logic, was proposed in [14] to adjust the number of sent EBs adaptively. In [15], the authors modeled the problem as a Discrete-Time Markov Chain and derived an analytical expression of the average joining time. Relying on this, they have formulated an optimization problem to minimize the average joining time by calculating an optimal EB schedule. Then, they have defined a model-based beacon scheduling algorithm in order to approximate the optimal EB schedule in real world scenarios. Later, the authors of the works described in [16, 17, 18] have proposed a deterministic beacon advertising algorithm to ensure collision-free EB advertisements that minimize the average joining time.

In the same prospective, the authors of [19] proposed a joining scheme based on channel quality estimation to improve the network formation process. It is particularly designed for congested environments with high radio interference.

The joiners select the best quality channel locally and passively scan for EBs messages to provide a short joining time even in severe congestion situations. Whereas, the authors of [20] and [21] proposed a timeslot partitioning technique and a collision-free broadcasting method. Their combinations resulted in a scheme that increases the EB rate without allocating additional slots and ensures a collision-free EB transmission.

Authors of [7] proposed a new beacon advertising algorithm based on a custom Trickle timer that allows nodes to increase or decrease the EB period periodically. However, this customized Trickle timer is not adaptive. According to the authors, it leans on a pre-configured period, which depends strongly on the considered network topology. Indeed, there are no real inconsistencies to trigger higher/lower EB sending frequencies. In response, Bell-X was proposed later in [6] to provide a dynamic configurable EB period. Besides, a parallel rendezvous-based algorithm was proposed in [22] to reduce the joining time. The proposal allows joining nodes, while passively scanning, to rendezvous with each other to create a cluster in which each node can calculate the other nodes' scan channel. Hence, upon the association of the first node, it can predict the scan channels of its cluster and then send EBs to those channels. So the other nodes belonging to the same cluster can associate rapidly.

Building upon previous research, the authors of [23] introduced a dynamic resource management algorithm during the network bootstrap. They proposed a dynamic allocation strategy of shared slots based on the estimated rate of control messages exchanged in the neighborhood to improve the network performance during the formation process. While in [24], the authors proposed a channel condition-based dynamic beacon interval scheme in which the beacon transmission period varies according to the channel congestion status. In [25], the authors have shown the demerit of the high priority given to EBs on the network formation and propose a solution in which the control packets have been taken into consideration during the formation process.

Recently, the authors of [26] proposed a distributed radio listening that uses a two-way transmission strategy based on the parallel rendezvous technique presented in [22]. The purpose of this solution is to divide the scan channel hopping sequence between the passively scanning joining rendezvous chain members in order to increase the probability of EB reception within the same cluster and achieve high synchronization speed. Besides, in [27], the authors have identified some challenges that may delay or hinder the adoption of the 6TiSCH stack for Industry 4.0. They mainly focused on the initial 6TiSCH network formation and introduced their algorithm, called Diagonal, for a rapid and reliable formation of dense networks. The work still in progress, and no experimental results were given. Finally, the authors of [28] analyzed and identified the negative effect of the TSCH-minimal configuration[4] on the formation process then proposed two schemes, namely: opportunistic priority alternation and rate control; and opportunistic channel access. The former dynamically adjusts the priority of control packets to provide sufficient routing information during network bootstrapping, whereas the latter allows the nodes having urgent packet to transmit it quickly.

Table 1 presents a comparison of the solutions discussed above under different metrics. In summary, we can observe that the proposed solutions concern only advertising nodes in most cases. Generally, the designers impose to use a specific format of the slotframe or the timeslot, and this can prevent applying the solution with any other scheduling mechanisms [13, 14, 20]. Moreover, in [3, 12, 16, 17, 23], the proposed schemes aim mainly to add several advertisement slots, which may increase collisions, spend more energy, and reduce the network capacity. Another issue concerns the advertisement rate, which is generally taken as a function of a single parameter such as the number of neighbors or number of advertisers in the network. At the same time, it should be tuned carefully in an adaptive manner according to the need.

Table 1: Comparison of TSCH network formation solutions

Algo.	Solution ported on	Add slots	Modify SF	Modify Slot	EB rate	Scan type
[5]	Advertisers	No	No	No	Periodic	Passive
[3, 12]	Advertisers	Yes	Yes	No	N/A	Passive
[13]	Advertisers	Yes	Yes	No	Periodic	Passive
[14]	Advertisers	Yes	Yes	No	Adaptive	Passive
[15]	Advertisers	Yes	No	No	Periodic	Passive
[16, 17]	Advertisers	Yes	No	No	Periodic	Passive
[18]	Advertisers	Yes	No	No	Periodic	Passive
[19]	Advertisers + Joiners	Yes	Yes	No	Periodic	Passive
[20]	Advertisers	No	No	Yes	N/A	Passive
[21]	Advertisers	Yes	Yes	No	N/A	Passive
[6, 7]	Advertisers	No	No	No	Adaptive	Passive
[22]	Joiners	No	No	No	N/A	Passive
[23]	Advertisers	Yes	No	No	Adaptive	Passive
[24]	Advertisers	No	No	No	Adaptive	Passive
[25]	Advertisers	No	No	No	Periodic	Passive
[26]	Joiners	No	No	No	N/A	Passive
[27]	Advertisers	Yes	No	No	N/A	Passive
[28]	Advertisers	No	No	No	Adaptive	Passive
FAN	Advertisers + Joiners	Yes	No	No	Adaptive	Active

Additionally, we can notice in this table that all the solutions adopt a purely passive scan procedure. The negative implications of such a scan process can be summarized as follows:

- Indeed, the multi-channel used by TSCH mitigates the effects of interference and multi-path fading; however, it became difficult to ensure that the joining node is aligned on the same channel as the advertising node since the joining nodes scan randomly over all available channels;
- The aforementioned problem can lead to very long listening periods for

joiners that consume valuable energy. Besides, a very long network formation process hinders the network from being operational just after deployment and prevents quick data acquisition and delivery;

- In a large-scale network, the larger the diameter of the network the bigger is the joining time. Furthermore, distant nodes from the coordinator would lose a large amount of their energy on listening for EBs resulting in poor energy balancing;
- Nodes mobility may cause significant network downtime—especially where nodes are unable to associate to the network for a long period because of synchronization loss;
- The passive scan is less sensitive; there is no priority when it comes to sending EBs, which are sent randomly over all the channels;
- To accelerate the network formation using the passive scan, generally, we need to add more advertisement slots, which may increase collisions and reduce the network capacity.

The above-mentioned issues of existing state-of-the-art solutions motivated us to propose an active scan based joining scheme, dubbed FAN, which accelerates the network formation by investing on both sides (i.e., joiners and advertisers) along with an adaptive mechanism to disseminate beacons. Accordingly, and more precisely, FAN combines an enhanced active scan procedure with a Trickle algorithm to send EBs on demand using the same communication channel. This would reduce the scan periods, handle mobility and shorten the association time, making the network functional rapidly without losing a big amount of energy. The next section is devoted to introducing the building blocks of our mechanism, along with its general operations.

3. FAN: Building Blocks and Global Design

This section states the considered FAN assumptions, before introducing its building blocks, namely: the active scan procedure and the Trickle algorithm. Next, it gives an overview of FAN's operations.

3.1. Assumptions

Throughout this paper, the following assumptions are considered:

1. We consider a Low-power and lossy network running the IEEE 802.15.4 TSCH MAC mode;
2. At the network bootstrap, all the nodes are joiners, and there is only one coordinator that announces the network presence via periodic transmissions of EBs;
3. The joiners perform active scan on clear channels using EBRs to extract EBs, from a coordinator or any other advertiser, and become advertisers;

4. Just like the coordinator, advertisers run any schedule or at least the TSCH-Minimal configuration;
5. To enable efficient and collision free active scan operations, the slotframe size is assumed pre-configured on all joiners.

While most of these assumptions are common with the state-of-the-art solutions, assumptions 3 and 5 are special to our scheme. Assumption 3 deploys the active scan mechanism defined in the TSCH standard [2]. Concerning point 5, assuming that the SF size is initially configured on joiners allows efficient operations of FAN. Nevertheless, if such information can not be assumed, the joiner can select a recommended SF size with two main repercussions on FAN performance as discussed in Sec. 3.2.1 below.

3.2. FAN Building Blocks

3.2.1. Active scan in TSCH networks

The authors of [6] have carried out an experiment to evaluate the viability of the TSCH active scan method. The results showed that the active scan procedure, as proposed by the standard, does not improve the network formation process since the probability that synchronized nodes receive a beacon request is too low. Because, in the standard’s active scan method, a joiner sends only one single EBR each specific period and keeps its radio on passively scanning for incoming EBs until the next period. This has prompted us to design a smart EBR sending strategy in order to increase the EBR reception probability.

In our design, EBRs are sent repeatedly and periodically on randomly chosen channels until association, as shown in Fig. 2. Indeed, at bootstrap, each joiner chooses a random channel that will be used for repetitive and consecutive EBR transmissions. Sending several consecutive EBRs aims to increase their reception probability by advertisers. To ensure receiving a submitted EBR from a non-associated node by one of the active slots of the running schedule in the same channel, the sending time T_{req} should be at least the same length as the SF duration defined in Eq. 2 (see Fig. 2); hence T_{req} can be defined by Eq. 3. It should be noted that using a T_{req} smaller than the SF duration can enable EBR reception by advertisers on the same channel but does not guarantee it. On the other hand, a larger T_{req} might increase collisions with a running schedule. In our proposal, we set $T_{req} = SF_duration$.

Once done, the advertiser will schedule a solicited EB TX slot in the next SF on the same physical channel on which it received the EBR, as shown in Fig. 2. In order to make our active scan more efficient, the solicited EB should be scheduled as to coincide within the joiner’s passive scan period T_{pscan} (Fig. 2), which should have a size of at least the SF duration as imposed in Eq. 4. Otherwise, the joiner could miss the TX slot on which the requested EB is sent. At the reception of the solicited EB, the joiner associate to the TSCH network. If not, the above process repeats until association.

$$SF_duration = Nb_timeslots \times timeslot_duration \quad (2)$$

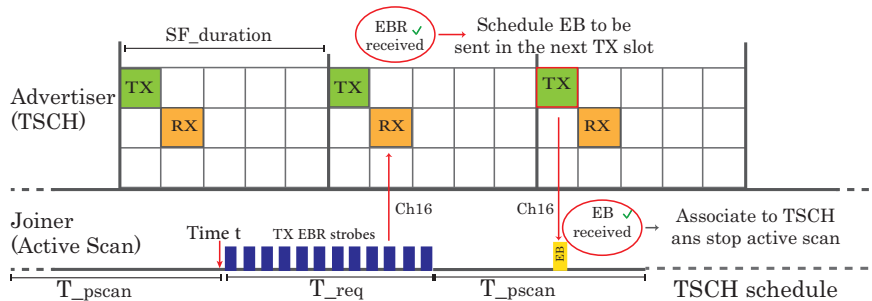


Figure 2: The proposed active scan procedure

$$T_{req} \geq SF_{duration} \quad (3)$$

$$T_{pscan} \geq SF_{duration} \quad (4)$$

Now that the active scan can yield acceptable reception probability and be effective in improving the formation process, adaptive and efficient solutions to avoid potential collisions need to be designed. To do so, we appeal for the Trickle algorithm presented in the sequel among other mechanisms presented later in Section 4.

3.2.2. The Trickle Algorithm

Trickle is an algorithm used for information dissemination in Low power and Lossy Networks (LLN). It dynamically adjusts its transmission rates to spread new (inconsistent) information very rapidly within milliseconds while sending only a few messages per hour when the information does not change (remain consistent) [29]. Besides, it allows nodes to hear each others' transmissions and delete their owns' if they have recently heard enough of the same. These two actions constitute the main Trickle mechanisms, namely the adaptive period and the suppression mechanism.

To realize the aforementioned behavior, Trickle divides time into intervals and defines a set of parameters and variables. The global parameters that are the same for all nodes involve the minimum and maximum sizes of the time intervals I_{min} and I_{max} , and the redundancy constant k . The local variables for each node are: the current size of the time interval I , the moment of a scheduled transmission t within I , and the counter c of received consistent messages within I . Each node will follow the following six rules to decide the action to take:

1. First, choose I randomly between I_{min} and I_{max} .
2. Initialize c to zero, select t randomly between $[I/2, I]$, and start the interval.
3. For each consistent message received, increment c .
4. At time t , if c is less than k , transmit the data.

5. At the end of an interval, double the size up to I_{max} and go to step 2.
6. If an inconsistency is detected, shrink I to I_{min} , if it does not already equal to I_{min} , and go to step 2.

The parameters and variables I_{min} , I , and I_{max} along with the related rules (1, 2, 5, and 6) enable to realize the adaptive period mechanism of Trickle; while parameters and variable k , c , and t along with rules 2, 3, and 4 enable the suppression mechanism. Thus, choosing $I_{min} = I_{max}$ will disable the adaptive period and choosing k to infinity (coded as $k = 0$) will disable the suppression mechanism. Finally, it should be noted that choosing t randomly between $[I/2, I]$ in rule 2 allows for a half-interval *listen-only* period in which a node only listens for incoming messages. Such a *listen-only* period enables effective suppression and is of great interest to our use of Trickle in FAN.

Indeed, the use of Trickle in FAN is two-folds. On the one hand, its *listen-only* period and the suppression mechanism are deployed by active scan joiners to ensure T_{pscan}/T_{req} succession and suppress redundant EBRs, respectively (see Fig. 3), and hence minimize collisions with a TSCH schedule. On the other hand, Trickle's adaptive period and its *listen-only* period are used by advertisers to send in a smart manner EBs as to coincide within the T_{pscan} period and respond rapidly to EBRs, which allow nodes to associate faster, and will further minimize potential collisions.

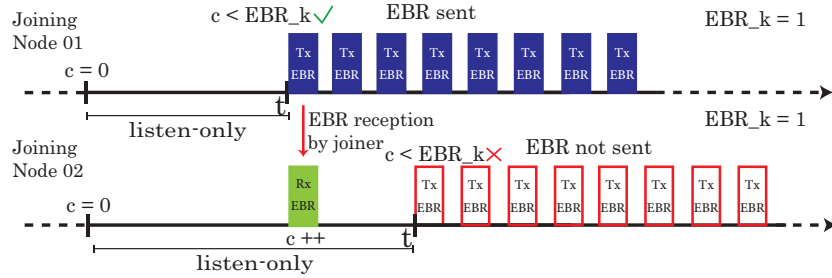


Figure 3: EBR suppression mechanism

3.3. FAN's Overview

In FAN, the TSCH network formation is established, on the one hand, through advertising the presence of the network by sending EBs. And, on the other hand, through active scanning for EBs via EBR transmissions. Basically, our proposal leans on an adaptive Trickle-based advertisement strategy of EBs (Trickling EBs) and an enhanced Trickle-based active scan procedure for joiners to look for a TSCH network by sending EBR packets (Trickling EBRs) while avoiding collisions with a running schedule.

Initially, the network coordinator is the only advertiser node. It starts by sending EBs periodically, while the rest of the nodes scan actively for EBs. In FAN and different from the standard procedure, the period between two consecutive EBs (t_{EB}) is adaptive and handled by a Trickle algorithm. Thus,

an advertiser increases the rate of sending EBs at its startup or upon receiving an EBR to quickly advertise the presence of the network. Otherwise, it decreases the sending frequency gradually. By doing so, we can achieve a high sending frequency of EBs only when needed. The details concerning the EBs Trickle algorithm parameters and operations are reported in Section 4.1.

At the same time, a joining node performs an active scan and keeps sending repeatedly and periodically EBR packets until receiving an EB and associate to the network. To minimize active scan traffic, EBR transmissions are handled through another Trickle timer that avoids interference between joining nodes and minimizes collisions with the running TSCH schedule. For instance, if a joining node receives an EBR, it will delete its own EBR transmission to avoid EBR broadcast storms that might prevent the proper functioning of the active scan. In addition, such EBR suppression minimizes the probability of collisions with the running TSCH schedule. Related configuration parameters and sub-modules are reported in Section 4.2. Finally, while the above helps to minimize collisions with a running TSCH schedule, the aim of FAN is to eliminate any collisions. To do so, before sending every single EBR strobe, the joiner performs a Clear Channel Assessment (CCA) to check the availability of the channel. In the case where the channel is not clear, the transmission of the current request (EBR strobe) is canceled. Otherwise, the EBR strobe can be sent. Fine-grained tuning and corner case considerations that allow proper operations of our collision avoidance mechanism are further discussed in Section 4.3. In summary, FAN provides an adaptive advertising scheme to optimize TSCH network formation via the three aforementioned mechanisms. The diagram in Fig. 4 summarizes the overall function of FAN through the interactions between its three main components. Detailed discussions of each component are presented in the following section.

4. FAN: Mechanisms and Detailed Operations

This section details the three main components of FAN, namely Trickle-based EB advertisements, Trickle-based EBR transmissions, and EBR-TSCH collision avoidance.

4.1. *Trickling EB Advertisements*

The operations ensured by EBs require an adaptive strategy in order to optimize their transmissions. Indeed, EBs should be sent so often in some cases (i.e., inconsistency) and inversely in other cases (i.e., consistency). Thus, we adopt Trickle to adapt EB advertisements to the network needs. In our proposal, node startup in TSCH mode and EBR reception by any advertiser constitutes *inconsistent* situations that require fast EB advertisements because the advertiser has learned that some nodes in the neighborhood need to join the network. Thus, it increases the rate of sending EBs on the same physical channel on which it has received the request by resetting the EB Trickle timer

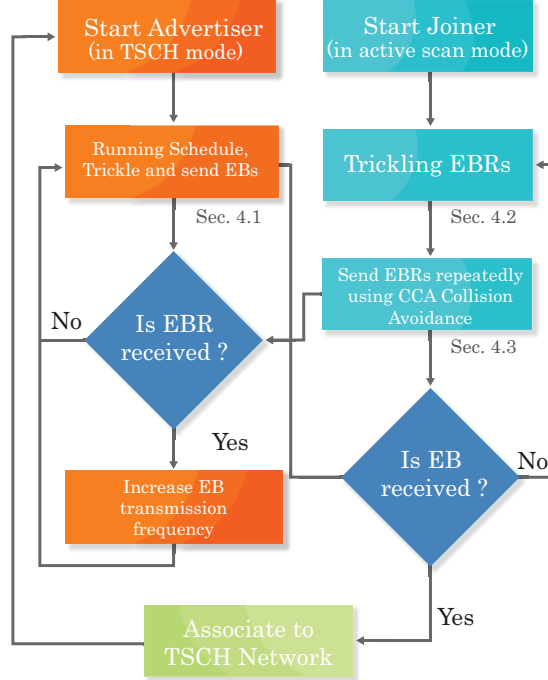


Figure 4: FAN’s overview diagram

to I_{min_EB} . On the other hand, receiving the same EB from advertisers constitutes a *consistency*, which indicates that the network is already established, and hence the t_{EB} could be increased up to I_{max_EB} .

EB Trickle configuration parameters should be carefully defined to provide an efficient announcement strategy of the network. So, the minimum interval, I_{min_EB} , should be as low as possible to disseminate EBs rapidly and allow the neighboring nodes to associate to the network. It can be defined as the minimum distance between two consecutive TX slots. For instance, in TSCH-minimal configuration [4], the distance between these two slots is equal to the slotframe length (in *ms*) as defined in Eq. 2. Besides, I_{min_EB} should be set as to ensure coincidence between the solicited EB to be scheduled and the T_{pscan} of the joiner. Taking a worst-case scenario where the advertiser has received the first EBR strobe, the *listen-only* period of the I_{min_EB} would equal to T_{req} (Eq. 5). From the above and taking into account Eq. 3, I_{min_EB} can be given by Eq. 6.

$$\text{listen-only period of } I_{min_EB} = T_{req} \quad (5)$$

$$I_{min_EB} = 2 \times T_{req} \quad (6)$$

Now, based on Eq. 6, if the last EBR strobe is received by an advertiser, the first part of the I_{min_EB} will be a *listen-only* period, leaving the second one for scheduling the EB. Thus, and for the sake of avoiding EB collision with potential EBRs, the T_{pscan} period should equal two times the T_{req} period as defined in Eq. 7. Hence, respecting these equations would ensure a proper interplay between the joiners and advertisers when exchanging beacons.

$$T_{pscan} = 2 \times T_{req} \quad (7)$$

The maximum interval, I_{max_EB} , representing the maximum period between two consecutive EBs, should be as high as possible to reduce the number of EBs and allow only those necessary for (re)synchronizations and (re)associations. In the implementation of [30], authors propose to use 50s as the maximum EB period but let it configurable and tunable according to the network density. Finally, the redundancy constant k should be as large as possible to increase the chance of receiving EBs by joiners to accelerate association. In the current version of FAN, the suppression module is disabled. So, $EB.k = 0$.

Following the above, we can guarantee that EBs are sent Trickled all along the network lifetime while giving more chances to the new nodes or disassociated nodes to rejoin the network quickly. The diagram in Fig. 5 summarizes the functioning of the EB Trickle algorithm.

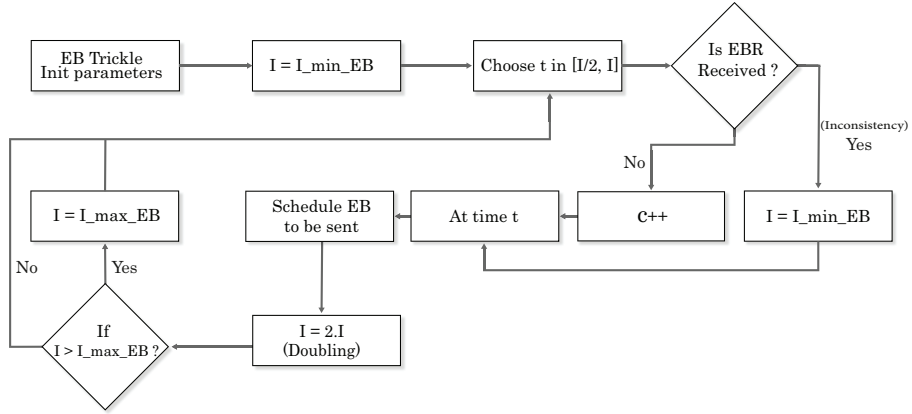


Figure 5: Trickleing EB advertisements

4.2. Trickleing EBR Transmissions

The following sub-sections detail the sub-modules of the second component of FAN, which include Trickle-based EBR transmissions, EBR-TSCH communications, and the random dynamic schedule.

4.2.1. EBR Trickle timer configuration parameters

As stated in Section 3, joiners deploy Trickle timers to minimize EBR traffic. Indeed, at start-up, each joiner chooses a random channel and starts sending periodic EBRs using Trickle. The EBR Trickle’s parameters should be fine-grained in order to send EBRs seamlessly without interference with the neighboring joiners on the same channel as well as with the running TSCH schedule.

Following Eq. 7, the *listen-only* period of the EBR Trickle timer, which represents the T_{pscan} period (see Fig 2) should equals twice T_{req} . This imposes a minimal size of the I_{min_EBR} of four times T_{req} . Taking into account Eq. 3, the interval I_{min_EBR} should be at least equal the four times the SF length. In addition, and in order to enable proper functioning of the EBR’s Trickle suppression mechanism, I_{min_EBR} should be set at least two to three times as long as it takes to transmit k EBR packets [29] (i.e., $I_{min_EBR} \geq 2 \text{ to } 3 \times k \times T_{req}$). Otherwise, unintended behavior and deteriorated performance might be witnessed. Furthermore, considering the interplay between EB and EBR Trickle timers defined in Eq. 5, and in order to enable fast associations (i.e., setting the I_{min_EB} as small as possible), it is necessary to set the constant k to one ($EBR.k = 1$). Accordingly, the interval I_{min_EBR} is defined as by Eq. 8. It should be noted that a joiner only increments $EBR.c$ at the reception of the first strobe, which occurs at time t of the current interval. This way, Trickle operations are not affected by an EBR packet that crosses to the next interval.

$$I_{min_EBR} = 4 \times SF_duration \quad (8)$$

The interval I_{max_EBR} should equal to I_{min_EBR} since we disable, in this version of FAN, the adaptive period of the EBR Trickle timer (Eq. 9). This is because, in our use-case, all joiners exchange the same EBR, which constitutes a *consistency*. No *inconsistency* is defined to reset the timer. The diagram in Fig. 6 sums-up the overall operations of the EBR Trickle algorithm.

$$I_{max_EBR} = I_{min_EBR} \quad (9)$$

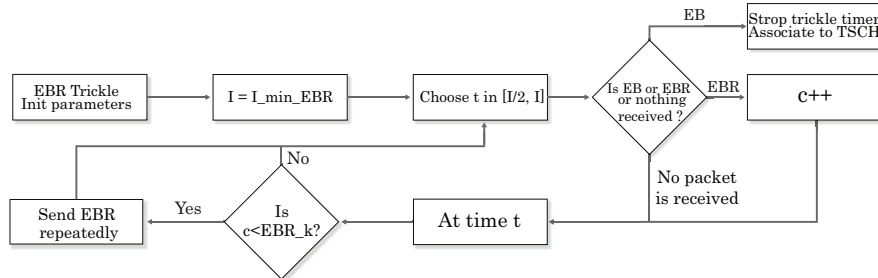


Figure 6: Trickling EBR transmissions

4.2.2. EBR-TSCH communications

To ensure that one EBR strobe gets received by an advertiser, several timing values should be tuned. For instance, in [31], the authors reported the exact structure of a TSCH TX/RX timeslot as defined by the standard (Fig. 7). From this figure, a reception only happens within a period of T_{rx} length (RX Packet in Fig. 7).

Basically, to be able to receive an EBR strobe by an advertiser, the T_{rx} period of the RX timeslot should be at least sufficient to hold two consecutive EBR strobes along with the inter-strobe interval T_i as by Eq. 10. This ensures that even if the T_{rx} starts just after the transmission of the first EBR strobe, its length is enough to get the second one. Otherwise, we can not ensure that the EBR will be aligned with the T_{rx} part of the RX slot. This condition can be met, thanks to the small size of EBR strobes.

$$(2 \times T_s) + T_i \leq T_{rx} \quad (10)$$

For instance, according to [7] and the implementation proposed by [30], a TSCH node turns its radio only for $T_{rx} = 5.452ms$ with a $10ms$ RX timeslot to receive a packet. Accordingly, we just need to ensure that we can get at least two EBR strobes within this T_{rx} time. This can be met since the size of an EBR strobe is about 34 bytes, which needs only $T_s = 1.05ms$ to be transmitted. Consequently, an RX TSCH slot can receive at least two EBR strobes and remains $3.352ms$ as inter-strobe interval T_i (see Fig. 7). It should be noted that no acknowledgment is generated since EBRs are sent in broadcast.

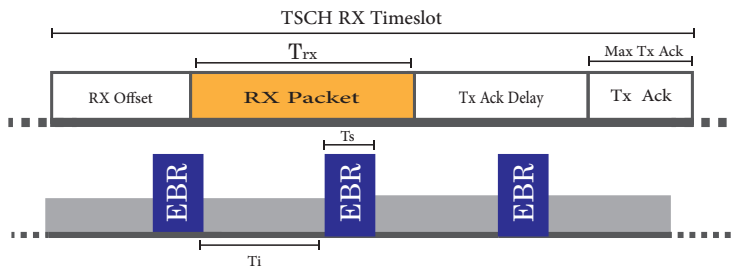


Figure 7: Enabling EBR communication with TSCH

4.2.3. Random dynamic RX schedule

The advertised TSCH schedule might not contain enough RX slots for receiving EBRs. For instance, the minimal schedule [4] includes only one common shared (CS) slot in which all nodes compete to send/receive packets. For such minimal schedules, it is necessary to randomly add some RX slots all along the SF's unscheduled slots to achieve a high reception rate of EBRs. The number of added RX slots is tunable and trades-off between joining time and energy consumption.

Fig. 8 shows an example where three RX slots are added randomly in an SF of 15 timeslots and 4 channel offsets. It is clear from this figure that adding multiple slots randomly on different channel offsets allows advertisers to listen to numerous joiners' requests within the same SF and achieve faster joining times. Finally, the number of added slots can be tuned depending on EB requests. Thus, upon joining a network, a node might add the maximum possible RX slots during some SFs, then switches to a lower number until receiving an EBR where the number of extra slots might be increased.

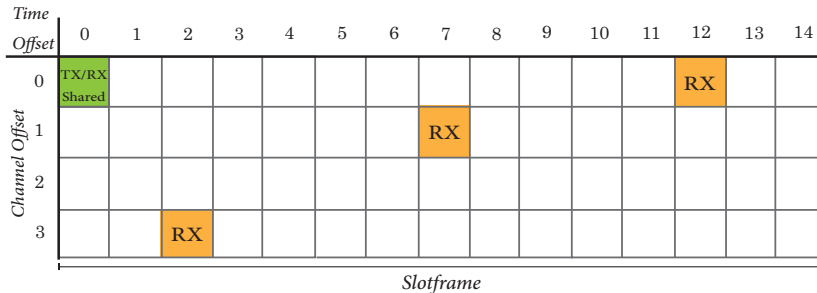


Figure 8: Random Dynamic Schedule

4.3. Collision avoidance with a running TSCH schedule

One of the disadvantages of an active scan is potential collisions of EBRs with the TX packets of an already established TSCH network, which may lead to deteriorating the performance of the network. To overcome this issue, we exploit the time between two consecutive EBR strobes T_i to perform several consecutive CCAs as shown in Fig. 9, and abort EBR transmissions if the channel is found to be busy.

In the standard TSCH TX timeslot, a node can turn on its radio to transmit for a maximum period of $T_{tx} = 4.256ms$ [2, 4]. Hence, to avoid any EBR transmissions within T_{tx} , the EBR CCA and T_i interval should be tuned carefully, while respecting Eq. 10. To do so, taking the case where the joiner performs an EBR CCA before sending a given EBR strobe and detects a TSCH packet transmission. In order to avoid any collisions, the current strobe must be discarded, and T_i should be set as high as possible in order not to interfere with the longest TSCH packet of size T_{tx} and its potential acknowledgment as specified in Eq. 11.

$$T_s + T_i \geq T_{tx} + Rest\ of\ TX \quad (11)$$

Nevertheless, while Eq. 11 can prevent collisions, it violates Eq. 10 for the standard timeslot size of $10ms$. Indeed, from Eq. 10, the maximum value of T_i is $3.352ms$. However, satisfying Eq. 11 requires at least a T_i of $6.806ms$ since, in a $10ms$ timeslot, the value of *Rest of TX* derived from the standard [2] equals $3.6ms$. To resolve this issue, we propose to suppress the transmission

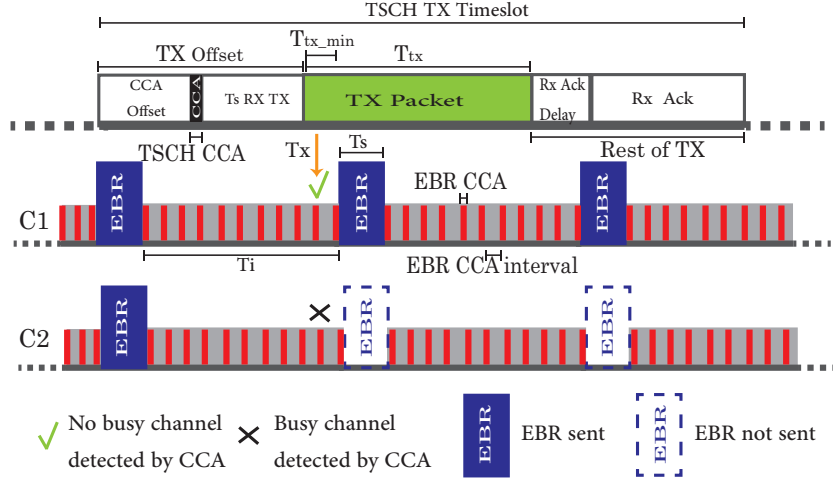


Figure 9: Avoiding EBR collision with TSCH packets

of x consecutive EBRs when detecting a busy channel instead of one. In order not to affect negatively the EBR reception probability, x should be set as the smallest integer allowing to give the T_i value that makes Eq. 12 feasible without conflicting with Eq. 10. This proposal relaxes Eq. 11 and replaces it with Eq. 12. For instance, applying Eq. 12 in a $10ms$ timeslot with an $x = 2$ gives a T_i of $2.878ms$ that satisfies both equations, as shown in Fig 9. We can see from this figure that we have two main cases. In the first case (C1), during the transmission of a TSCH packet, the joiner performs several EBR CCAs during its T_i interval, then it sends its EBR packets (not canceled), and this is due to different operation channels. In other words, if the EBR CCA does not detect that the channel is busy, then the next x EBRs will be sent normally. On the contrary, in case (C2), if the joiner's EBR CCA detects that the channel is busy (used for TSCH packet transmission), the next x EBRs will be canceled to not collide with the transmitted TSCH packet as well its eventual acknowledgment.

$$x \times (T_s + T_i) \geq T_{tx} + Rest\ of\ TX \quad (12)$$

Now that our proposal can avoid interference detected by the EBR CCA, it remains to ensure that EBR CCA can detect all potential collisions. To do so, the followings must be met:

- The time between two consecutive EBR CCA should be as short as the time it takes to transmit the smallest TSCH packet (T_{tx_min}). This ensures that if an EBR CCA misses the packet, the following one can detect it.
- The last EBR CCA in the T_i period should be just before the EBR strobe to be transmitted.

- The first T_i period of each EBR packet should equal to $(T_{tx} - T_{tx.min} + \text{Rest of TX})$ to avoid collision with a potential acknowledgment of a missed CCA detection. This ensures that if the first T_i starts exactly after the transmission of a minimum size packet (after $T_{tx.min}$), the EBR strobe will not collide with the eventual upcoming acknowledgment.

With the above rules, the only cases where our mechanism fails to detect a TSCH transmission is when the last EBR CCA of 128us happens within a T_s period before the start of T_{tx} , which has a small probability of occurring. Besides, the TSCH CCA might detect such a case and postpones the transmission. Finally, it should be noted that when an EBR CCA detects an activity, the silent period defined by Eq. 12 does not require CCA, and hence the joiner should not perform EBR CCAs on such a period. This avoids detecting a potential TSCH acknowledgment as a TX packet.

4.4. The big picture

According to the above, the CCA-based active scan (Sec. 4.3) can allow submitted EBR packets to be received by advertisers without conflicting with the schedule. The Trickle-based EBR (Sec. 4.2) also minimizes collisions within the schedule while avoids interference with neighbors, and the EBR-TSCH fine-grained timings along with the dynamic random RX schedule allow high reception rates of EBR/EB for fast joining times. While, the Trickle-based advertisement of EBs (Sec. 4.1) optimizes their transmission. The interactions between these three sub-modules of the FAN's active-scan mechanism are illustrated in Fig. 10.

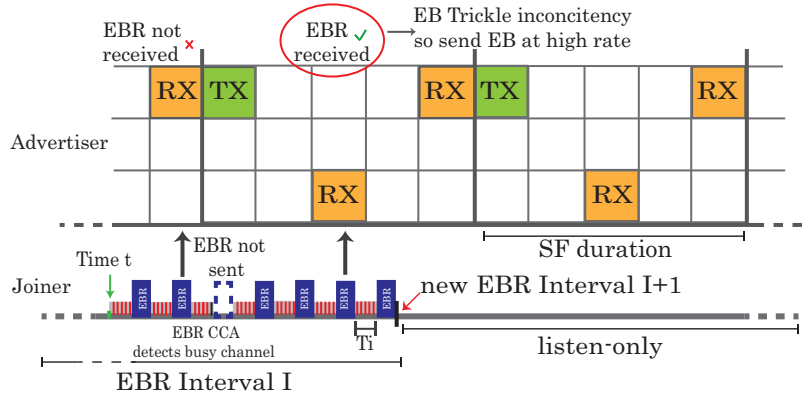


Figure 10: CCA-based active scan association

As can be seen from Fig. 10, the joining node runs the EBR Trickle, so it selects a time t within the EBR interval $[I/2, I]$, and turns on its radio transceiver for T_{pscan} period until time t arrives. At this time, it starts sending repeatedly several EBRs during T_{req} . Firstly, we can see that the second EBR strobe has

not been received by the dynamic TSCH RX slot of the advertiser even if their timings are aligned, and this is due to the channels mismatching. Secondly, we have the case where the TSCH TX has a packet to transmit. In this case, the collision avoidance mechanism leads to canceling the x outgoing EBR strobes in order not to conflict with the running schedule. Finally, in the case where the EBR is aligned with a dynamic TSCH RX slot timings and communication channel, the EB request can be correctly received by the advertiser, which triggers an EB Trickle timer inconsistency that increases the sending frequency of EBs. Doing so, the next TSCH TX slots can be used to send the requested EB. Once the T_{req} period is over, a new T_{pscan} is launched. During this latter, the joining node can: (1) receive an EB and associate to the TSCH network, (2) receive an EBR from a neighboring joiner; which leads to canceling the outgoing EBRs to avoid collisions, or (3) do not receive any packet if no EBR has been received. It should be noted that the whole process is repeated until node association.

5. Performance evaluation

In this section, we present the methodology used to evaluate the performance of our proposal along with the principal findings. In the aim to appraise the performances of our FAN scheme, we evaluate and analyze its performance accurately and compare it to state-of-the-art solutions, under varied network parameters and conditions, namely: RV/RH [3], Bell-X [6], and the standard’s passive scan scheme based on the TSCH-minimal schedule (PS-TSCH) [4]. For this, we run extensive simulations with varied metrics using the Cooja simulator under the Contiki OS, a well-known operating system for IoT devices. We implemented our proposal on emulated Zolertia Z1 motes, which operate at 2.4 GHz with an effective data rate of 250 Kbps and holds only 8 KB RAM and 92 KB Flash memory, to add on the realistic-aspect of the evaluated algorithms and prove their lightweight aspects. Every single simulation was repeated 10 times with different random seeds. We report the mean along with the standard error in the graphs of Section 5.2.

5.1. Evaluation methodology and measured metrics

The analysis of the network formation joining mechanism relies on several compromises. Such a mechanism involves many parameters that can affect its performance, so it is essential to select the configuration that improves the synchronization and connection process while keeping low energy consumption and high throughput. After a battery of experimental parameters tuning, we have chosen a slotframe composed of five timeslots as SF size with $15ms$ as timeslot length to achieve low latency while keeping a low duty cycle [32]. The timings configuration and the Trickle’s parameters are defined and presented in Table 2.

We have identified five key parameters that influence the network formation process, namely: network size, number of channels, reception success rate, node

Table 2: Configuration parameters

Parameter	Value	Parameter	Value
<i>SF_duration</i>	<i>75ms</i>	<i>I_min_EB</i>	<i>150ms</i>
<i>T_req</i>	<i>75ms</i>	<i>I_max_EB</i>	<i>54s</i>
<i>T_s</i>	<i>1.05ms</i>	<i>EB_k</i>	<i>0</i>
<i>T_i</i>	<i>3.352ms</i>	<i>I_min_EBR</i>	<i>300ms</i>
<i>EBR_CCA</i>	<i>128μs</i>	<i>I_max_EBR</i>	<i>300ms</i>
<i>EBR_CCA interval</i>	<i>0.8ms</i>	<i>EBR_k</i>	<i>1</i>
<i>Dynamic RX cells</i>	<i>1</i>	<i>x canceled EBRs</i>	<i>3</i>

mobility, and slotframe length. To study the effect of these parameters on the performance of our proposed solution and compare it to the state-of-the-art solutions, two main metrics are measured: (i) Association time: time elapsed from the network start-up until all nodes have joined the network, and (ii) Number of sent EBs: refers to the total number of sent EBs by all nodes during experiments.

We have randomly generated multiple distributed network topologies with an average density of 8 neighbors per node. In each topology, one coordinator is deployed, and the remaining nodes are initially joiners. As mentioned before, our solution FAN is compared to the standard’s passive scan scheme. As this later does not define any advertisement policy and its performance heavily depends on t_{EB} , we evaluated two variants of the passive scan. In the first, $t_{EB} = 2s$ (PS-TSCH-2S), and the second, $t_{EB} = 5s$ (PS-TSCH-5S). For the sake of collision avoidance, nodes randomize their EB transmissions in the range $[0.75, 1]$ of the above periods. In addition to PS-TSCH, RV and RH solutions have been implemented as they are described in [3], and to have a fair comparison with our scheme, we have used an EB period $t_{EB} = 5s$. Finally, for the Bell-X implementation, we have reproduced exactly the same Bell-32 configuration as it was described in [6] for the sake of having a rapid and efficient association process.

Mainly, our evaluations lean on five simulation sets. In every single set, we vary the above-mentioned parameters to study a particular aspect. In the first set, we vary the network size. Hence, we vary the number of nodes per network while keeping the same average density. This is so to encompass a wide range of typical industrial IoT applications, going from small to large networks. In the second set, we vary the number of available communication channels in the 100-nodes network. Although TSCH can use 16 channels, some channels could be blacklisted because of interference. In the third set, we vary the reception success rate in the 100-nodes network in order to study the effect of data loss in LLNs on the evaluated solutions. In the fourth set, we wait for the network formation and stabilization, and then we inject new nodes in the network to study the effect of network churn, nodes’ re-associations and/or mobility, as well as the performance of the collision avoidance mechanism by varying the data rate and the number of added nodes. [In the last set, we vary the SF size](#)

in order to assess the performance of our solutions under different SF lengths. To do so, three slotframe lengths of 5, 50, and 101 timeslots per SF have been used. Table 3 summarizes the varied parameters for each set. Finally, in order to study the complexity of our proposal, we calculate and compare its code size and memory usage to those of the state-of-the-art solutions.

Table 3: Varied parameters

Parameters	Set 1	Set 2	Set 3	Set 4	Set 5
Network size	49,100,169,225	100	100	49	100
Nb. channels	8	4,8,12,16	8	8	8
Success rate	1	1	1,0.7,0.5,0.3	1	1
Added nodes	0	0	0	1,2,3,4	0
SF length	05	05	05	05	5,50,101

5.2. Results and discussions

5.2.1. Effect of Network size

The first experiment aims to study the effect of the network size on the network formation process.

Fig. 11a depicts the time needed to form the network when using FAN, PS-TSCH, RV, RH, and the Bell-X solutions on different network topologies. We can observe that the association time increases in proportion with the network size and diameter in all approaches. So small values lead to form the network quickly, while large values can delay the association time. However, the FAN solution recorded the lowest association time values. This is due to the fact that FAN uses the EBR packets to speed up the association process by extracting EBs from the already associated neighbors. In addition, the advertisers use a smart advertisement policy, which allows increasing the rate of the EB announcements at need.

FAN is followed by Bell-32, which remains better than the rest of the solutions because of its adaptive manner of sending EBs. However, such adaptivity is still limited because it is mainly based on the network topology and exhaustive configuration timings, which results in a high number of EBs (see Fig. 11b). In the standard's passive scan, the association time values increased by increasing the interval between two consecutive EBs. And this can be explained by the fact that the passive scan advertisement policy relies basically on a fixed period to send EB in which the joining nodes scan passively and randomly on all available channels. Meanwhile, the RV and RH solutions can improve the association time of the PS-TSCH when using the same interval $t_{EB} = 5s$, and this is due to the fact that each advertiser selects randomly its own *TimeOffset/ChannelOffset* to advertise EBs.

Concerning the number of exchanged EBs, Fig. 11b draws the total number of EBs that have been sent all along the simulation time by all nodes in each network topology. It is noticeable from this graph that the number of exchanged

EBs increased with network size. This is owing to the additional number of nodes that send EBs to announce the presence of the network. Unlike all the passive scan-based solutions, which reveal a high number of exchanged EBs, the number of exchanged EBs in FAN is very low. Indeed, using Trickle as an EB advertisement policy along with active scan reduces the number of exchanged EBs drastically while keeping a very rapid and efficient association process, notably in large networks.

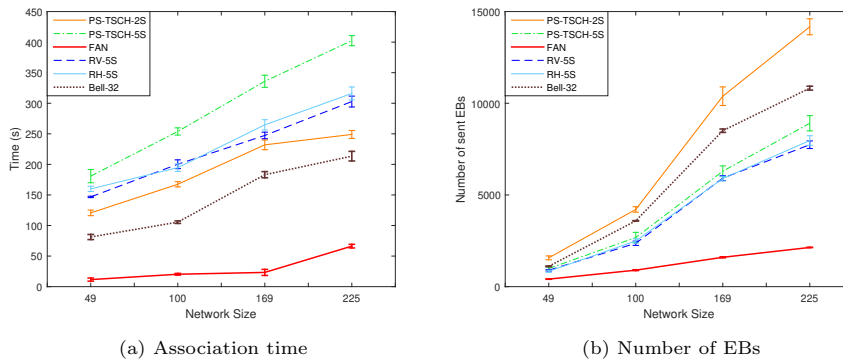


Figure 11: Network size effect

5.2.2. Effect of available communication channels

In this second set of experiments, we aim to assess the impact of the number of available channels used for communication on the association process.

Fig. 12a presents the time elapsed from the network startup until the association of all nodes within the 100-nodes network when varying the number of communication channels. From this chart, we can notice that the association time increased as long as the number of available communication channels increased. This returns to the evidence that joining nodes can not be aligned quickly with the same advertiser's channel when using a large number of communication channels. Nevertheless, FAN presented a very low association time, which is due to the use of the active scan procedure along with the dynamic random RX slots that help to discover the presence of a TSCH network and the Trickled advertisement strategy that spreads EBs over all the channels. Concerning the other solutions, we can see that Bell-32 is better than the other passive scan-based solutions because it generated a large number of EBs to speedup the association process. For the rest of the solutions, we can clearly see that the association time is very important, particularly when using PS-TSCH with $t_{EB} = 5s$; this is due to the low rate of sending EBs.

Concerning the number of exchanged EBs, Fig. 12b shows the number of exchanged EBs between all the nodes in the network when using several channel sequences. It is clear from the charts that using only two channels for communication generates a reduced number of EBs compared to sixteen channels. This

is owing to the rapid alignment of communication channels. Yet, our proposal, which leans on the Trickle algorithm to advertise beacons, revealed a very low number of exchanged EBs compared with passive schemes. Indeed, Bell-32 and PS-TSCH with a high advertising rate $t_{EB} = 2s$ recorded a high number of EBs comparing to those with a reduced sending rate $t_{EB} = 5s$, but this last comes to delay the association time.

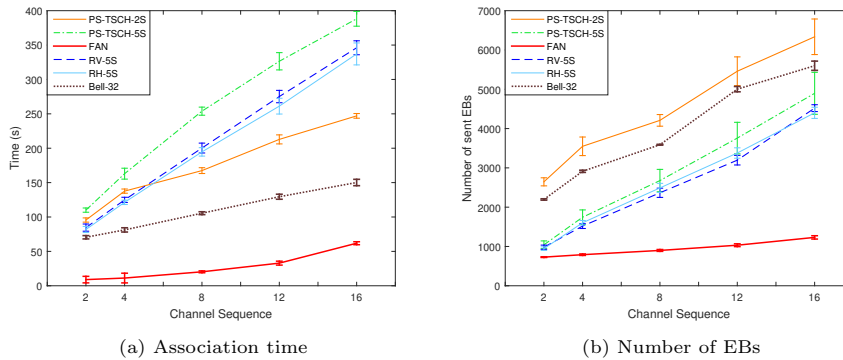


Figure 12: Channel hopping sequence effect

5.2.3. Effect of the network loss rate

The third set of experiments targets to appraise the effect of network loss on the formation process.

Fig. 13a depicts the association time in the 100-nodes network when varying the reception success rate. As shown in the figure, using a low reception success rate (e.g., 30%) leads to delay the association time in both FAN and passive scan solutions. This is due to the recurrent re-transmissions of lost packets. Nevertheless, FAN is not as strongly affected as the passive scan-based solutions because, in our proposal, the association and advertising plans are adaptive, so if a given node does not receive an EB, it asks for it until its reception. On the other hand, advertiser nodes do not lean on a fixed sending period of EBs; such period is handled with more flexibility according to the need.

Concerning the number of exchanged EBs, Fig. 13b presents the resulting number of transmitted EBs during the network formation process when varying the reception success rate. Comparing the results of passive scan-based solutions with those of FAN, it is clear that in our scheme, the number of EB sent increases slightly when reducing the success rate. This can be explained by the fact that our design relies on a guided association process, unlike the other schemes, which are based on a semi-adaptive / fixed advertising period and a totally random scanning process. These latter led to a high number of exchanged EB in the network, particularly when using a small advertising period, which reduces the network capacity.

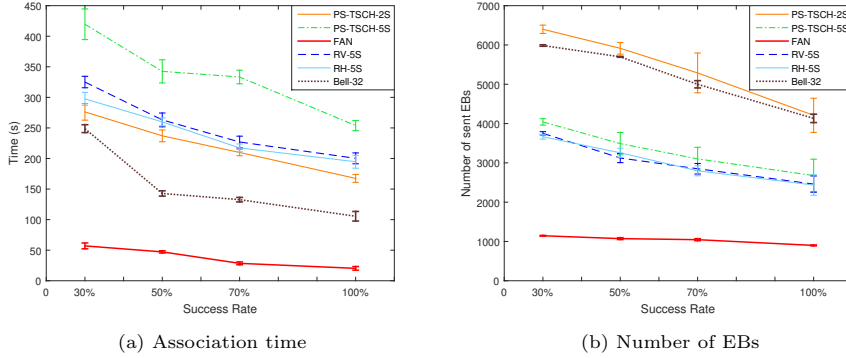


Figure 13: Success rate effect

5.2.4. Impact of churn, mobility, and re-associations

The fourth set of experiments aims to study the effect of network churn, node mobility, and re-association on an already established TSCH network. Besides, it considers a study of the collision avoidance mechanism.

Fig. 14a depicts the time needed for one new node to join a stable network (i.e., already formed network). It is clear from the chart that our proposal recorded the lowest association time values compared to PS-TSCH, even when using a high advertising rate of EBs ($t_{EB} = 2s$). This is due to the adaptive manner of generating EBs, which is basically triggered by the active scan. Obviously, in the passive scan-based solution, the association time increased as long the inter-beacon interval increased.

Fig. 14b illustrates the frequency of generated EBs by a given advertiser. In PS-TSCH, the frequency and the number of generated EBs are mainly based on the inter-beacon interval, which is maintained all the network lifetime. Hence, a high advertising rate leads to accelerate the network formation. Still, it leads to reduce the network's capacity and increase the overhead and vice-versa. However, in our FAN scheme, it is clear from the figure that the number of EBs is high at the beginning of the network formation, then it is reduced as long the network is more stable. At the moment where a new node tries to join the network (50s), we can see that the number of EBs increases for a while then goes down as the network is stable again.

To get an in-depth study of the effectiveness of the collision avoidance mechanism, we waited until the network were formed and started running a schedule; in which several RPL control and data packets are exchanged, then we varied two parameters: (i) the data sending rate and (ii) the number of new joiners. In the former, we injected one new joiner, and varied the data sending rate, while in the latter, we fixed the data sending rate to one packet every 500ms and varied the number of injected nodes. We report the percentage of EBR packets that were suppressed by the collision avoidance mechanism, among all sent EBRs during the injected node(s)' association time(s).

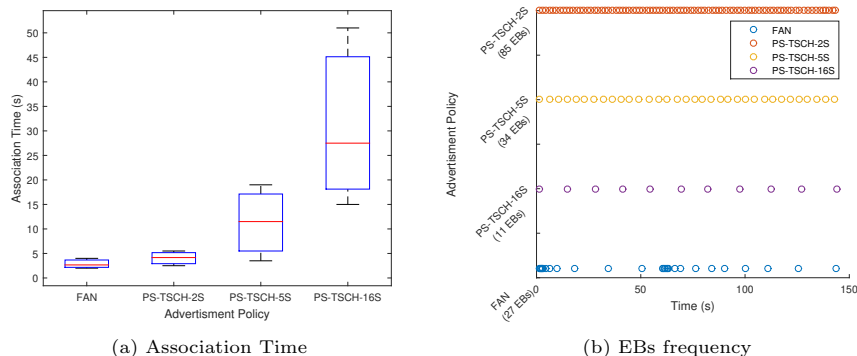


Figure 14: Effect of node de-association/re-association

Fig. 15a presents the percentage of EBR transmissions that were suppressed to avoid collision with the schedule when injecting only one node. This chart shows that our collision mechanism is triggered more often to avoid interference when the data rate increased and vice-versa. For instance, 15% of EBRs were suppressed when increasing the data rate to one packet each $75ms$. This shows the ability of our mechanism to avoid such collisions by canceling interfering EBR transmission even if it might slightly delay the joining process. Nonetheless, if with such delays, FAN remained faster and efficient than PS-TSCH, as can be witnessed from Fig. 14a.

Fig. 15b shows the percentage of the EBR transmissions that were suppressed when varying the number of injected nodes. We can see from this figure that adding 4 new joiners simultaneously led to an increase in the probability of collision with the running schedule (14% of the total number of planned EBR transmissions were suppressed). This can be due to the fact that several joiners were scanning actively on different channels at the same time. Nevertheless, the percentage of suppressed EBRs were not very important thanks to the inherent EBR suppression mechanism and the use of extra dynamic RX slots that enabled a high reception of sent EBRs. This has accelerated the joining process and avoided recurrent collisions.

5.2.5. Effect of the Slotframe Size

The last set of experiments intends to assess the effect of the slotframe size on the network formation process.

Fig. 16a depicts the effect of the SF size on the association time. On this chart, we can observe that the association time increased with increasing SF lengths in all solutions. This can be explained by the fact that the length of a slotframe introduced a trade-off in traffic capacity, network latency, and energy consumption. For instance, in a slotframe of five slots, the common shared slot repeated ten times compared to one in an SF of fifty slots. Thus, using short slotframes increased the network capacity, reduced congestion and accelerated the association process. Overall, FAN recorded the lowest association times

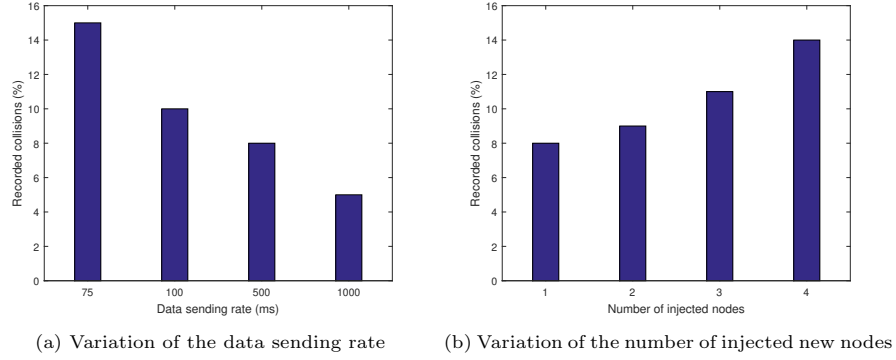


Figure 15: Percentage of suppressed EBRs (avoided collisions)

compared to the passive scan-based solutions. Furthermore, the association time in our proposal is not strongly affected by the SF length. On the contrary, longer SFs may advantage FAN since they allow to add several dynamic random RX slots.

Concerning the number of exchanged EBs, Fig. 16b shows that increasing SF sizes led to an increase in the number of exchanged EBs during the association process. Indeed, in the minimal configuration, long SFs reduced the traffic capacity and increased network latency. This was manifested by sending more EBs before getting associated. Comparing FAN to the state-of-the-art solutions, we can notice that it reported the lowest number of EBs since they were mainly triggered by EBRs and sent adaptively using Trickle.

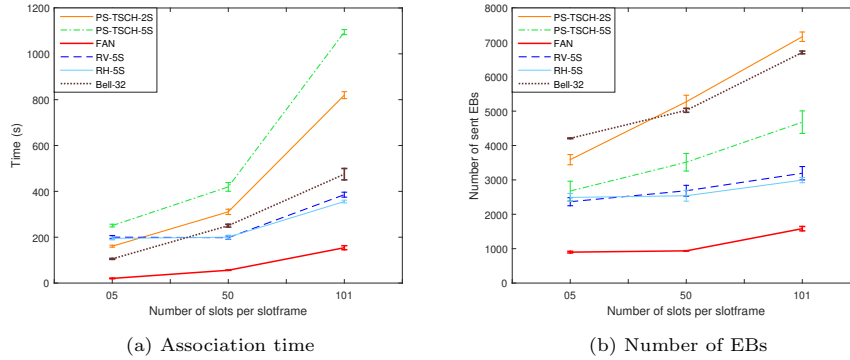


Figure 16: Slotframe length effect

5.2.6. Code Size and Memory Usage

Finally, and in order to evaluate the complexity of the code, we run a test under Contiki to measure the required memory resources by each evaluated

algorithm on Z1 motes. The obtained results are reported in Table 4. As can be seen from this table, FAN required some additional RAM and ROM resources compared to the other solutions, which is due to the Trickle timers implementations, EBR processing, and collision avoidance. Nevertheless, these additional memory requirements are acceptable considering the benefits brought up by FAN.

Table 4: RAM and ROM usage

	FAN	RV/RH	PS-TSCH	Bell-X
ROM (byte)	62 937	57 433	57 323	59 289
RAM (byte)	6 760	6 596	6 596	6 684

5.2.7. Summary and Discussion

The simulation results show that FAN outperforms state-of-the-art solutions under several aspects, making it efficient and powerful in forming industrial TSCH networks. It accelerates the association process and reduces the network overhead even under network churn, mobility, and/or de-association/re-association of nodes. This is due to the fact of using stateless Trickle timers, which are already implemented in Contiki OS, local EBR transmissions, and practical collision avoidance mechanisms. Nevertheless, while these mechanisms are efficient and practically simple to implement, they require a small extra memory footprint.

6. Public Testbed Validation

In the aim to validate our simulation results and consolidate FAN’s efficiency, we present in this section real world experimentation over the FIT IoT-LAB platform [33], an open large-scale infrastructure dedicated to IoT and IIoT experiments.

We have mainly run two sets of experiments in the Inria Grenoble site of IoT-LAB in which the average density is about 2 to 4 neighbors per node (M3 open node). In the first set, we fixed the number of channels to 8 and varied the number of nodes per network as follows: 28, 99, 153, and 290 nodes, to study the effect of the network size. In the second set, we fixed the number of nodes to 209 and varied the number of available channels in order to study their effect on network formation. To do so, we used 2, 4, 8, 12, and 16 channels in each experiment. Obtained results are reported in figures 17, 18.

Fig. 17a shows the time elapsed from the network startup until all the nodes receive a valid EB and join the network on different network sizes when using FAN and PS-TSCH. It is clear from the figure that the time needed to form the network increases with the network diameter in both approaches. Obviously, large networks require more time to be fully formed. However, FAN recorded

a very reduced association time compared to PS-TSCH. This is a proof of the efficiency of the active scan combined with a Trickle advertising policy of EBs, which achieves a very low association time even in large networks.

Concerning the number of exchanged EBs, Fig. 17b shows that the number of generated EBs increases with the number of nodes per network. Nonetheless, our scheme reported a reduced number of EBs when compared with PS-TSCH. This is due to the Trickle advertising approach, which accelerated the network formation, increased throughput, and reduced overhead.

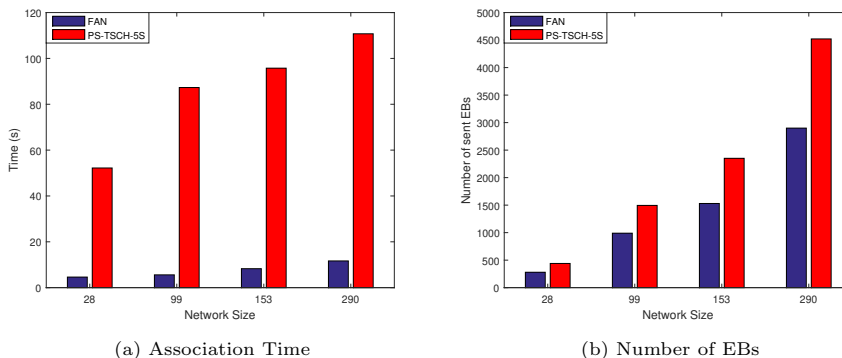


Figure 17: Network size effect

For the second set, Fig. 18a illustrates the association time in a 209-nodes network when varying the number of available channels. It is clear from this figure that using a reduced number of channels (e.g., 4) led to a rapid association time in the two approaches. This is strongly related to network density. Hence, a reduced number of communication channels promoted the alignment of the sender’s and receiver’s communication channels. Nevertheless, FAN revealed better values than PS-TSCH, even when using a large number of channels. This is achieved due to FAN’s explicit EB solicitation via EBRs and the dynamic random RX slots.

Finally, Fig. 18b presents the resulting number of transmitted EBs during the network formation process for different communication channels. It is clear from the bar charts that the number of sent EBs increased with the number of available channels. As explained before, a high number of communication channels in a sparse network can lead to long scanning periods, especially in PS-TSCH procedures, because of channel misalignment. This, in turn, can lead to sending additional EBs over all channels before getting associated. Despite that, if we compare the total number of exchanged EBs in PS-TSCH with those of FAN, it is clear that it is reduced by FAN.

7. Conclusion and Future Work

In this paper, we investigated the network formation process in TSCH networks and presented a fast joining scheme dubbed FAN. This latter is based on

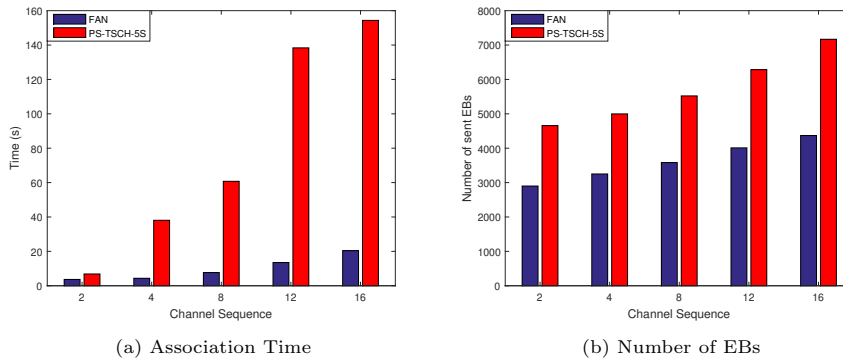


Figure 18: Channel hopping sequence effect

CCA-based active scan and Trickle transmissions to reduce the joining time, improve the network capacity, and establish a reduced and balanced energy consumption among the nodes. FAN was evaluated through extensive simulations and experimentation on the IoT-LAB testbed. Obtained results show that FAN achieves significant improvements in terms of association time and overhead when compared with state-of-the-art passive scan-based solutions. As future work, we plan to devise an adaptive EBR transmission mechanism to optimize the active scan procedure and further reduce collisions. We also plan to enable the suppression mechanism of EBs to increase the network capacity while keeping the network tightly synchronized. Besides, we consider devising a mechanism that takes into account the quality of available channels to improve FAN operations in harsh and radio interfering environments.

References

- [1] IEEE, *IEEE802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks. Part 15.4: Low-Rate Wireless Personal Area Networks Amendment 1: MAC Sublayer*. IEEE Std., April 2012.
- [2] IEEE, “Ieee standard for low-rate wireless networks,” *IEEE Std 802.15.4-2015*, pp. 1–709, 2016.
- [3] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, “Fast join and synchronization schema in the ieee 802.15. 4e mac,” in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 85–90, IEEE, 2015.
- [4] X. Vilajosana, K. Pister, and T. Watteyne, “Minimal ipv6 over the tsch mode of ieee 802.15. 4e (6tisch) configuration,” tech. rep., IETF, 2017.
- [5] D. De Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, “A performance analysis of the network formation process in ieee 802.15. 4e tsch wireless sensor networks,” in *ISCC*, pp. 1–6, IEEE, 2014.

- [6] J. Vera-Pérez, D. Todolí-Ferrandis, and J. Silvestre-Blanes, “Bell-x, an opportunistic time synchronization mechanism for scheduled wireless sensor networks,” *Sensors*, vol. 19, no. 19, p. 4128, 2019.
- [7] J. Vera-Pérez, D. Todolí-Ferrandis, S. Santonja-Climent, and J. Silvestre-Blanes, “A joining procedure and synchronization for tsch-rpl wireless sensor networks,” *Sensors*, vol. 18, no. 10, p. 3556, 2018.
- [8] L. Wang and 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)*, pp. 1–3, IEEE, 2017.
- [9] D. Fanucchi, B. Staehle, and R. Knorr, “Network formation for industrial iot: Evaluation, limits and recommendations,” in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 227–234, IEEE, 2018.
- [10] C. M. G. Algora, E. O. Guerra, S. Montejo-Sánchez, E. M. G. Fernández, and K. Steenhaut, “A theoretical association time model for ieee 802.15. 4 tsch networks,” *IEEE Communications Letters*, 2020.
- [11] T. Watteyne, M. Palattella, and L. Grieco, “Using ieee 802.15. 4e time-slotted channel hopping in the iot: Problem statement,” tech. rep., IETF, 2015.
- [12] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, “Fast network joining algorithms in industrial ieee 802.15. 4 deployments,” *Ad Hoc Networks*, vol. 69, pp. 65–75, 2018.
- [13] T. P. Duy and Y. Kim, “An efficient joining scheme in ieee 802.15. 4e,” in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 226–229, IEEE, 2015.
- [14] T. P. Duy, T. Dinh, and Y. Kim, “A rapid joining scheme based on fuzzy logic for highly dynamic ieee 802.15. 4e time-slotted channel hopping networks,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, 2016.
- [15] D. De Guglielmo, S. Brienza, and 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)*, pp. 1–9, IEEE, 2016.
- [16] I. Khoufi, P. Minet, E. Livolant, and B. Rmili, “Building an ieee 802.15. 4e tsch network,” in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–2, IEEE, 2016.

- [17] I. Khoufi, P. Minet, and B. Rmili, “Beacon advertising in an ieee 802.15.4e tsch network for space launch vehicles,” *Acta Astronautica*, vol. 158, pp. 76–88, 2019.
- [18] I. Khoufi and P. Minet, “An enhanced deterministic beacon advertising algorithm for building tsch networks,” *Annals of Telecommunications*, vol. 73, no. 11-12, pp. 745–757, 2018.
- [19] J.-Y. Kim, S.-H. Chung, and Y.-V. Ha, “A fast joining scheme based on channel quality for ieee802.15.4 e tsch in severe interference environment,” in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 427–432, IEEE, 2017.
- [20] 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)*, pp. 588–599, IEEE, 2018.
- [21] A. Karalis, D. Zorbas, and C. Douligeris, “Collision-free broadcast methods for ieee 802.15.4-tsch networks formation,” in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 91–98, ACM, 2018.
- [22] C. M. G. Algora, V. A. Reguera, E. M. G. Fernández, and K. Steenhaut, “Parallel rendezvous-based association for ieee 802.15.4 tsch networks,” *IEEE Sensors Journal*, vol. 18, no. 21, pp. 9005–9020, 2018.
- [23] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, “Improving network formation in 6tisch networks,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 98–110, 2019.
- [24] A. Kalita and M. Khatua, “Channel condition based dynamic beacon interval for faster formation of 6tisch network,” *IEEE Transactions on Mobile Computing*, 2020.
- [25] A. Kalita and M. Khatua, “Opportunistic priority alternation scheme for faster formation of 6tisch network,” in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, pp. 1–5, 2020.
- [26] 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, p. 1303, 2020.
- [27] D. Fanucchi, B. Staehle, and R. Knorr, “On the suitability of 6tisch for industrial wireless communication,” in *Kommunikation und Bildverarbeitung in der Automation*, pp. 34–48, Springer Vieweg, Berlin, Heidelberg, 2020.
- [28] A. Kalita and M. Khatua, “Opportunistic transmission of control packets for faster formation of 6tisch network,” *ACM Transactions on Internet of Things*, vol. 2, no. 1, pp. 1–29, 2021.

- [29] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The trickle algorithm,” *Internet Engineering Task Force, RFC6206*, 2011.
- [30] S. Duquennoy, A. Elsts, B. Al Nahas, and G. Oikonomo, “Tsch and 6tisch for contiki: Challenges, design and evaluation,” in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 11–18, IEEE, 2017.
- [31] T. Chang, T. Watteyne, K. Pister, and Q. Wang, “Adaptive synchronization in multi-hop tsch networks,” *Computer Networks*, vol. 76, pp. 165–176, 2015.
- [32] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust mesh networks through autonomously scheduled tsch,” in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 337–350, ACM, 2015.
- [33] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, *et al.*, “Fit iot-lab: A large scale open experimental iot testbed,” in *2015 IEEE 2nd World Forum on Internet of Things*, pp. 459–464, IEEE, 2015.