# Mobility-Aware TSCH for Reliable IoT Connectivity

Charalampos Orfanidis, Oliver Orla Nielsen, Lucas Koefoed Schnügger, Xenofon Fafoutis
DTU Compute, Technical University of Denmark (DTU), Denmark
{chaorf, s164917, s164924, xefa}@dtu.dk

***Abstract*—Time Slotted Channel Hopping (TSCH) is a Medium Access Control (MAC) protocol widely adopted in Industrial Internet of Things (IIoT) deployments due to its high reliability in the presence of noise and interference. However, TSCH becomes suboptimal in mobile scenarios, where nodes frequently move in and out of coverage. This paper introduces a compact, mobility-aware framework based on an enhanced TSCH joining process, by embedding topology information into beacons to reduce mobility-related downtime. In a previous work, we examined basic functions of this framework, this paper presents new results on re-joining performance, power consumption, channel-set behavior, and drift-mitigation. The approach achieves up to an 8× power reduction during re-join, 30–40%, and a tenfold increase in de-synchronization time through intentional time-shifting.**

***Index Terms*—Internet of Things, TSCH, Reliability, Dependable IoT**

## I. Introduction

Internet of Things (IoT) technology has been widely used in the last decade in numerous applications such as smart home [1], smart health [2], industry 4.0 [3], and smart agriculture [4]. Especially in industry and agriculture, the IoT has a great potential to automate production through massive data collection and management to monitor the entire production cycle and ultimately boost productivity and profit.

Medium Access Control (MAC) protocols are an essential part of the IoT since they have an impact on various performance aspects such as the lifetime of the network, the reliability and robustness of the application, and a number of delays which can occur at different levels. Time Slotted Channel Hopping (TSCH) [5] is a MAC protocol that has received a lot of attention, especially for applications in Industrial Internet of Things (IIoT) [6]–[8]. It is able to provide reliability and robustness in low power, through a channel hopping function which mitigates the impact of interference and multi-path fading. All the nodes are synchronized because they have to follow a schedule which dictates a predefined channel offset, and a time offset for each node, to avoid collisions between them and any source of interference.

The popularity of TSCH led researchers to start using it outside of the IIoT context, where mobile nodes might be part of the network. For instance, Elsts et al. utilize TSCH on wearable sensors for healthcare applications, which require high reliability in heterogeneous environments with the presence of external interference [9]. Thus, they introduce a new TSCH schedule and an adaptive channel selection mechanism to increase the performance of TSCH for this application. Yet, the application assumes that the mobile nodes are always within the TSCH network.

One major challenge for TSCH, is to maintain high level of reliability even for mobile scenarios where the nodes may be moving in and out of coverage. The mobility patterns that nodes follow in mobile applications within IoT domain are heterogeneous, and in a previous work we demonstrated that TSCH, even with different scheduling schemes, cannot operate reliably in such cases [10]. The main reason for the low performance is that the mobile nodes might go out of coverage and disconnect from the TSCH network. Then, if the node moves within coverage again, the process to join the TSCH network includes a beaconing procedure which can add significant overhead. TSCH is also common to be organized as a tree data structure, and if a mobile node changes parent frequently, the corresponding schedule to serve this node will be outdated most of the time.

In this paper we present an approach to improve TSCH mobility by enhancing the process of joining and re-joining a TSCH network. The proposed framework makes this process quicker, more efficient, and overall more reliable. This is achieved by introducing a mechanism that encapsulates topology data within the TSCH beacons to decrease the joining time. Early results on these contributions have appeared in a conference publication [11]. The current paper extends further the approach with the following contributions:

- We examine the impact of the length of the channel hopping sequence for TSCH joining mechanism.
- We quantify the benefits of the enhanced joining mechanism for TSCH in terms of average power consumption.
- We introduce a drift mitigation mechanism that improves the time a node can take advantage of the benefits of fast re-joining by an order of magnitude.

The paper is organized as follows: Section II presents background information and the system design, Section III discusses the state of the art, Section IV goes through the evaluation, and Section V concludes the paper.

## II. Background & System Design

This section provides an overview of the operational mechanisms of TSCH and the system design.

### A. TSCH Overview

The TSCH protocol, part of the IEEE 802.15.4 [12] standard and amended in IEEE 802.15.4e-2015 [13], operates at the link layer and features frequency hopping and node synchronization. The nodes follow a duty cycle dictated by a schedule, allowing them to sleep, transmit, or receive packets to conserve power. To prevent collisions, simultaneous transmissions on the same channel are prohibited. The schedule is structured in a two-dimensional table called the slotframe, with time offset on the x-axis and channel offset on the y-axis. Each cell, or timeslot, typically lasts 10 ms, sufficient to send an IEEE 802.15.4 packet and receive an acknowledgment.

The TSCH coordinator sets up the network, defining the network ID, slotframe size, and Frequency Hopping Sequence (FHS), and initializes the Absolute Slot Number (ASN) to 0, which increments with each new timeslot. Nodes join the network by receiving an Enhanced Beacon (EB) packet from existing nodes, which contains essential synchronization information.

### B. Mobility-Focused TSCH Joining Mechanism

The key idea of the approach is that once a joining node has obtained basic network parameters, it can predict when and on which channel the next EB will be transmitted. This prediction relies on knowledge of the network's ASN, the channel hopping sequence, channel offset, and the TSCH schedule. With this information, the node can limit its radio activity to only the expected time slots and channels of EB transmissions, thereby significantly reducing idle listening and scanning overhead.

To enable this mechanism, the approach extends EB frames to include lightweight topology information, such as node identifiers, parent-child relationships, channel offset, ASN, and link status. This information is embedded using vendor-specific information elements and is disseminated through normal EB transmissions, avoiding the need for additional control slots or protocol changes. Each node maintains a local topology dataset that is periodically updated based on received EB information, allowing the network structure to be tracked and updated over time.

A key supporting mechanism is the local ASN tracking. Instead of relying on continuous synchronization from the network, a joining node estimates the ASN using timestamps derived from received EBs and a local real-time clock, compensating for processing and transmission delays. This enables the node to maintain a sufficiently accurate view of network timing for EB prediction. Using this timing information, the node can further reduce energy consumption by disabling its radio during periods where no EB is expected.

The proposed mechanism extends EB with Essential Synchronization Information (ESI), including the current ASN, channel offset, and compact topology metadata required for EB prediction. Assuming an IEEE 802.15.4 PHY operating at 250 kbps, adding $L_{ESI}$ bytes increases the transmission time by only

$$T_{OH} = \frac{8L_{ESI}}{250 \times 10^3},$$

which corresponds to approximately 32 $\mu$s per additional byte. Since EBs are transmitted infrequently (typically once every advertisement interval), the resulting increase in channel occupancy is negligible compared to regular data traffic and does not alter the TSCH schedule or require additional communication cells.

At the network level, the overhead scales linearly with the number of advertising nodes. If $N$ nodes transmit one EB every $T_{EB}$ seconds, the aggregate synchronization overhead introduced by ESI is

$$R_{OH} = \frac{8NL_{ESI}}{T_{EB}} \quad \text{bits/s},$$

which remains a small fraction of the available IEEE 802.15.4 bandwidth for practical network sizes. Consequently, the additional airtime and radio-on time introduced are negligible across the network and are outweighed by the reduction in join and re-join latency, as fewer random channel scans are required and newly joining nodes synchronize substantially faster.

Overall, this method improves both initial network association and re-joining performance, particularly for mobile nodes that may frequently disconnect and re-connect. By combining EB-based topology dissemination with local time synchronization, the approach enables more efficient discovery of potential parent nodes while reducing radio-on time during the joining process.

## III. Related Work

This section provides an overview of the state of the art including several attempts to add mobility features to the TSCH protocol. Beacon advertising is a typical method to advertise the network to a new node. There are numerous techniques that connection-based MAC protocols like TSCH utilize for network advertising.

Al-Nidawi et al., in [14] evaluate the performance of TSCH for mobile scenarios, and quantify the delay and energy overhead caused by disconnecting and re-joining TCSH network. An approach for improvement is to select appropriate slot frame size and scheduling scheme. Moreover, if adequate number of nodes are available to provide sufficient network coverage, the energy consumption is not increased. In an attempt to provide a solution, the same authors introduce a design in [15], based on the ACK packets which behave like beacon packets to advertise the TSCH network on a fixed channel in order to decrease the delay and the energy consumption. Mobile6TiSCH is based on the Single-hop Multiple Gateway (SHMG) architecture for enabling mobility in TSCH networks [16]. The authors present a tool that can configure TSCH for application scenarios involving several mobile nodes and border routers. The border routers, which act as gateways between the TSCH network and other networks, are used alongside different mobility patterns to gather insights and statistics on network performance. In our previous work [10], we evaluated existing TSCH scheduling approaches under heterogeneous mobility scenarios and showed that network performance deteriorates primarily because mobile nodes repeatedly lose synchronization and require expensive re-association. The present work builds upon these observations by addressing one of the identified bottlenecks, namely the TSCH joining and re-joining procedure. In [17], we presented the Active Connectivity framework which is able to provide a solution for mobile nodes to regulate characteristics like the speed, the direction of the mobile nodes to preserve connectivity in TSCH networks. This effort generates an impact towards the application, the time to move within a certain distance might be increased which in turn might affect the overall cost. The average joining time is modeled by means of a Discrete Time Markov chain, and then it is treated as an optimization problem in [18]. Based on the derived model an optimal TSCH schedule is calculated which minimizes the joining time. In a similar spirit to improve the TSCH joining process a random-based advertisement mechanism is presented in [19]. Based on the network conditions and the number of available channels, the algorithm determines in which order the nodes should transmit their beacon messages. In [20] the authors propose random TSCH advertisement algorithms. They propose two algorithms where the first broadcasts all the beacon messages in a single timeslot to take advantage of all the available channels. The second one broadcasts all beacons in every timeslot but in a single channel. Both algorithms have an enhanced version which is more complex and more energy demanding, assuming a node with more resources available. Algora et al. introduce a rendezvous-based algorithm [21] designed to reduce the TSCH joining time. More specifically the nodes that are more than one hop away from the coordinator can exchange information while their predecessors along the path to the coordinator are trying to associate.

Overall, existing approaches primarily improve mobility by modifying the advertisement mechanism, adapting the routing protocol, or relying on mobility-aware scheduling. While these methods improve connectivity or reliability, they typically require additional control traffic, dedicated protocol modifications, or assumptions about node mobility. In contrast, our approach leverages the existing EB dissemination mechanism to provide synchronization information that enables faster joining and re-joining with minimal protocol changes.

## IV. Evaluation

This section presents and discusses the results obtained by comparing the *Enhanced* TSCH joining mechanism we implemented with the *Original* one. To compare the performance of the two approaches, we used the Cooja simulator [22]. We observed and quantified how a new node scans and joins an existing TSCH network. We also evaluated a mobile node that temporarily disconnects and later re-connects with the network. For all experiments, the Minimal Scheduling Function [23] was used, and each simulation was executed for 10 minutes. The Zolertia Z1 [24] mote was selected as the simulation platform within Cooja. Each test case described in this section was executed 15 times using different random seeds to minimize statistical bias.

We assume a scenario where a node attempts to join an existing TSCH network via EB reception and select the most appropriate parent. An important aspect to investigate is how the size of the channel hopping sequence can impact the reception of EBs and discovery of potential parents. The channel hopping sequences we evaluate the Enhanced and Original TSCH versions are: 4, 8, 12, 16 channels. As the goal of this test is to evaluate the performance of parent discovery, both TSCH versions were configured to scan for an unlimited amount of distinct parents and with a scanning timeout of 16 s, which was determined as the optimal number for previous experiments in [11]. Fig. 1 shows the results and illustrate for various network densities, how many distinct parents were discovered on average. A trend we see for the Original version is that as we increase the number of channels, the number of discovered parents decreases since the probability for a node to receive an EB when it is randomly scanning decreases as well. On the contrary the performance of the Enhanced version is not affected when we increase the number of channels because the *EB prediction* mechanism is not dependent on the number of channels once the first EB has been received. For the rest of the evaluation the channel hopping sequence length is 16.

Next we quantify the power it takes a single node to join to an existing TSCH network for a set of topologies with different amount of nodes. To do so, we first quantify the time it takes for that. The scenario is therefore set up such that the joining node is out of range of all other nodes, while the network forms. Once all other network nodes have associated to the network, the joining node is moved within range of all nodes in the network – for this reason the test is performed on networks consisting of various amount of nodes. Then, from this moment the time required for the joining node to associate to the network is measured. For the scan timeout we used 16 s for the Enhanced version, and 180 s for the Original, as it is recommended by IETF. After receiving the first EB, a node may continue scanning for up to the duration defined by the scan timeout, or until it has received EBs from a maximum number of distinct neighbors. The value for the amount of distinct neighbors is 2 for both TSCH versions and is also recommended by IETF. Then, consulting the datasheet of the platform we use in Cooja (Zolertia Z1 [24]) we computed the power consumed based on the time it took for this process. Therefore in Fig. 2 we see that the Enhanced version consumes almost half of the power compared to the Original version, which is expected as it is analogous to the join time. The only difference which is not easily observable in the figure, is that in the Enhanced version the power consumption is not completely analogous to the join time due to a modification we made to deactivate its radio and conserve energy after it has received the first EB and know the next available time slot in which an EB may be broadcast.

The last part of the evaluation includes a scenario where

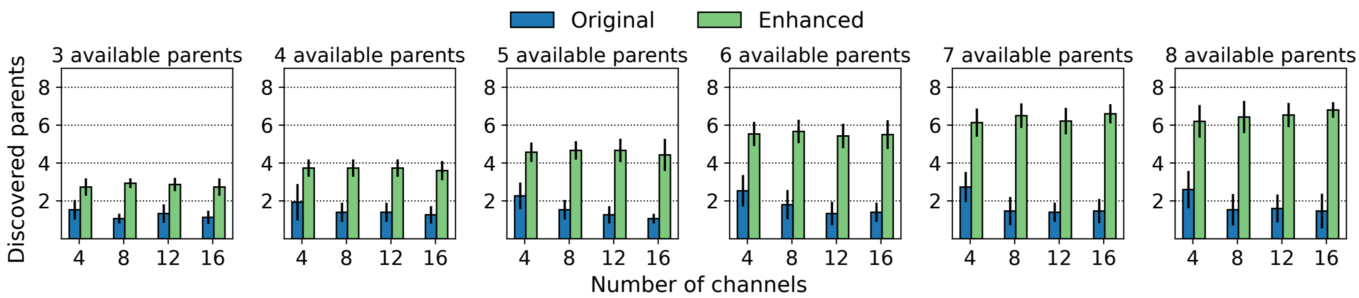


Fig. 1. The average number of parents considered during the scanning process, before a node associates to existing networks for different number of available parents, given various channel hopping sequences for Original and Enhanced TSCH versions.

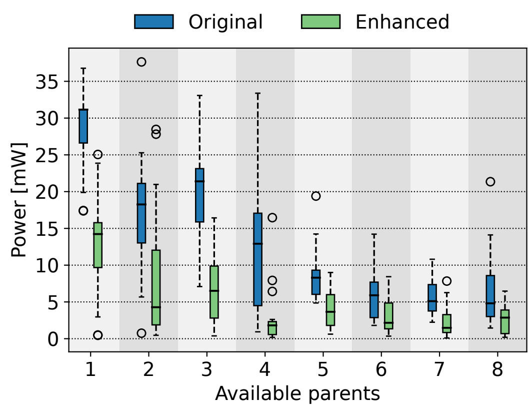


Fig. 2. The average power consumption of the scanning and joining process in a joining node for both the Original and Enhanced TSCH versions in various network densities.

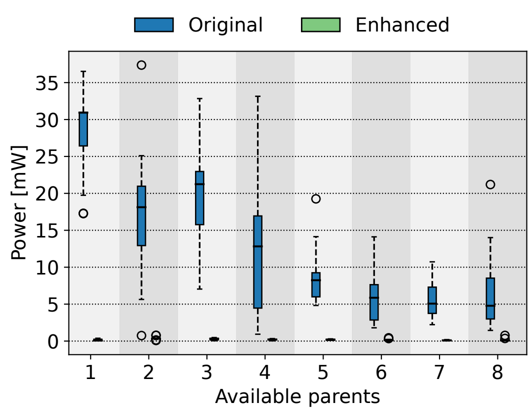


Fig. 3. The average power consumption during the re-joining process for the Enhanced and Original TSCH versions across varying network densities.

we examine the tolerated drift level. The method to track the ASN locally can resolve the issues of the time shift caused by transmission delay but it cannot reduce, or even determine, if the joining node has a continuous time drift, which is usually caused by clock drift in the microcontroller. If a node experiences clock drift its local clock will shift either positively or negatively, i.e. backwards or forwards in time. Given that the joining node should be able to re-synchronize its local ASN-timing to a network node at least once every 16 sec, after receiving the initial EB and given that the guard time (a time period a receiver wakes up before an expected transmission) in this implementation is 1.1 ms, the maximum supported clock drift can be calculated as follows:

$$D_{max} = \frac{S_{max}}{P_{sync}} * 10^6 \quad (1)$$

where the $D_{max}$ is the maximum drift in PPM, the $S_{max}$ is the guard time, and the $P_{sync}$ is the interval in which the time source will be refreshed. The maximum supported clock drift is 68.75 ppm.

If a significant *positive* drift is experienced, it is less likely to cause issues. For instance, if the positive drift of the joining node is so large that it exceeds the finishing guard time of the previous timeslot, the joining node may pick up some data from the previous timeslot, however, it will be discarded due to link layer safety mechanisms such as the checksum. Theoretically, as it is mentioned in above, the custom version should be able to tolerate up to $\pm 68.75$ ppm. To investigate this further we defined a scenario where a joining node is never associated to the network and instead its node re-synchronization feature was disabled such that its drift would not be reset every time an EB is received. The node will synchronize its ASN only to the first EB received and continue listening for EBs without associating or re-synchronizing its ASN timing. The node will do so until it has shifted its timing so much that it is not able to receive an EB when it is broadcast. The time between the first and last received EB was then measured, indicating the duration it would take for the drift of the joining node to reach such a level that it would be unable to receive future EBs. As it was mentioned already, the mechanism is much more tolerant towards positive drift which is reflected in the results listed in Table I. The same scenario was repeated with an introduced intentional shift of 6.5 ms, which is half of the duration of a time slot minus the guard time. This was expected to provide a more balanced resistance to drift regardless of direction. The overall shift window size has not increased particularly, however, it has been balanced such that the mechanism can tolerate negative and positive drift equally. Given that the drift direction cannot

| | De-synchronization time | |
|---|---|---|
| **Drift** | without intentional shift | intentional shift |
| −10 PPM | 57 s | 711 s |
| +10 PPM | 1271 s | 723 s |

TABLE I
DE-SYNCHRONIZATION TIME WITHOUT AN *intentional shift* AND WITH AN *intentional shift* OF 6.5 MS.

be predicted, having an equal tolerance on both sides enables the solution to provide drift tolerance guarantees that are more than 10 times higher than without intentional shift.

## V. CONCLUSION

This paper presented an enhanced TSCH joining framework designed to address the performance degradation of TSCH in mobile IoT scenarios. By embedding lightweight topology information into EBs and enabling local ASN tracking, the proposed mechanism allows mobile nodes to predict EB transmissions and minimize unnecessary radio-on time. The evaluation, conducted with the Cooja simulator across a range of network densities and channel hopping sequence lengths, demonstrated that the Enhanced version consistently outperforms the Original TSCH joining procedure.

Specifically, the Enhanced mechanism discovers a stable number of parent nodes regardless of channel hopping sequence length, while the Original presents a decreasing trend as the number of channels increases. In terms of energy efficiency, the Enhanced version achieves 30–40% reduction in power consumption during both initial joining and re-joining. Finally, the introduction of an intentional time-shift of 6.5 ms, equal to half the timeslot duration minus the guard time, balances tolerance to both positive and negative clock drift, increasing de-synchronization time by more than an order of magnitude compared to the unshifted case.

These results suggest that the proposed framework offers a practical, low-overhead approach to robust TSCH connectivity for mobile IoT deployments, such as autonomous agricultural vehicles and smart warehouse robotics, without requiring modifications to the underlying TSCH schedule or additional control traffic.

## REFERENCES


[1] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous internet of things build our future: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, 2018.

[2] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[4] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.

[5] S. Duquennoy, A. Elsts, B. A. Nahas, and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 11–18, 2017.

[6] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LDSF: Low-Latency Distributed Scheduling Function for Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8688–8699, 2020.

[7] M. Mohamadi, B. Djamaa, and M. R. Senouci, "RAST: Rapid and energy-efficient network formation in TSCH-based Industrial Internet of Things," *Computer Communications*, vol. 183, 2022.

[8] A. R. Urke, Ø. Kure, and K. Øvsthus, "A Survey of 802.15.4 TSCH Schedulers for a Standardized Industrial Internet of Things," *Sensors*, vol. 22, no. 1, 2022.

[9] A. Elsts, X. Fafoutis, G. Oikonomou, R. Piechocki, and I. Craddock, "TSCH Networks for Health IoT: Design, Evaluation, and Trials in the Wild," *ACM Trans. Internet Things*, vol. 1, Apr. 2020.

[10] C. Orfanidis, A. Elsts, P. Pop, and X. Fafoutis, "TSCH Evaluation Under Heterogeneous Mobile Scenarios," *IoT*, vol. 2, no. 4, pp. 656–668, 2021.

[11] O. Nielsen, L. Schnugger, C. Orfanidis, and X. Fafoutis, "Mobility-Focused Joining in TSCH Networks," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, (Los Alamitos, CA, USA), pp. 271–274, IEEE Computer Society, sep 2022.

[12] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.

[13] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, 2012.

[14] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the IoT MAC infrastructure: IEEE 802.15.4e TSCH and LLDN platform," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015.

[15] Y. Al-Nidawi and A. H. Kemp, "Mobility Aware Framework for Timeslotted Channel Hopping IEEE 802.15.4e Sensor Networks," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7112–7125, 2015.

[16] M. Pettorali, F. Righetti, and C. Vallati, "Mobile6tisch: a simulator for 6tisch-based industrial iot networks with mobile nodes," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 614–618, 2022.

[17] C. Orfanidis, P. Pop, and X. Fafoutis, "Active Connectivity Fundamentals for TSCH Networks of Mobile Robots," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2022.

[18] 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, 2016.

[19] 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/actuator networks," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2014.

[20] 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.

[21] 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, pp. 9005–9020, 2018.

[22] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pp. 641–648, 2006.

[23] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)." RFC 9033, May 2021.

[24] Zolertia, "Z1 Datasheet." http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf, 2010. [Online; accessed 15-Feb-2024].