Network Formation for Industrial IoT: Evaluation, Limits and Recommendations

Dario Fanucchi
Department of Computer Science
University of Augsburg
Augsburg, Germany
d.fanucchi@informatik.uni-augsburg.de

Barbara Staehle
Department of Computer Science
University of Applied Sciences Konstanz
Konstanz, Germany
barbara.staehle@htwg-konstanz.de

Rudi Knorr
Fraunhofer Institute for Embedded Systems and Communication Technologies ESK
Munich, Germany
rudi.knorr@esk.fraunhofer.de

***Abstract*—An open standardized protocol stack proposed by IETF is nowadays emerging in industrial wireless communication. Main building blocks are TSCH at MAC layer and RPL as routing protocol. This standard architecture is able to replace proprietary technology and to guarantee a timely, reliable and energy efficient communication. However, IETF can not offer a one-size-fits-all solution. Therefore, implementers of industrial IoT have to correctly set the parameters of the protocols in this stack, adapting it to the application requirements and on the physical topology. This paper focuses on the network formation procedure proposed by IETF 6TiSCH working group, a mandatory phase before nodes may transmit any sensed data. We evaluate through simulations the impact of TSCH- and RPL-parameters on the duration of and the energy consumed for the network formation process. We describe how to avoid an unsuccessful network formation and we give guidelines for an appropriate parameter setting, depending on typical network topologies.**

***Index Terms*—TSCH, RPL, 6TiSCH, performance, network formation**

I. Introduction

Wireless technology and especially wireless sensor networks offer several advantages over traditional wired communication solutions, such as lower installation cost, flexibility, mobility and energy awareness. Today's WSNs differ from those designed and implemented about 15 years ago. An analysis of the product portfolio of leading manufactures reveals the leading presence of WirelessHART [1] and ISA100 [2] nowadays. In both standards, low power wireless communication is based on time slotted channel hopping (TSCH), to guarantee determinism of channel access and to enhance resilience to interference. [3] and [4] are only some example of research papers or real deployment, which report on fulfilment of industrial requirements with TSCH technology, i.e., wire-like 99.999% reliability, 5+ year battery lifetimes.

However, a survey [5] highlights a wrong perception of wireless communication solutions, since the participants still identify reliability issue as a primary objection to the use of wireless in industrial applications. One reason may be the existence on the market of several independent and competing proprietary technologies, supported by different industry players. The IETF has already understood this issue and has proposed an open standardized protocol stack for the industrial internet of things (IIoT), denoted in the following as IIoT-stack [6]. In this architecture, TSCH at MAC layer and IPv6 connectivity are main building blocks, and they are glued together by the IETF 6TiSCH standardisation efforts and recommendations [7]. However, using this stack with its default parameters is not reasonable and there is also not the only one-size-fits-all technology in this context. So it is also crucial to correctly set up the parameters of the protocols present in the IIoT-stack. Otherwise, operators will obtain unsatisfactory results.

The goal of this paper is to explore the procedure proposed by IETF 6TiSCH working group (6TiSCH-WG) for the network formation phase, the so-called *6TiSCH Minimal Configuration* (6TiSCH-MC) [8]. This initial phase is mandatory, before nodes may transmit any sensed data, and the coordination between medium access control (MAC) and network layer is here crucial. An improper tuning of the their protocol parameters may lead to a very long time for (or even unsuccessful) network formation. We evaluate through simulations the delay and the charge consumed for the network formation. Implementers of IIoT-solutions will benefit from our guidelines for an appropriate parameter setting, depending on some typical physical topologies.

To this end, Section II introduces the IIoT-stack, and gives an overview how a 6TiSCH-network is formed. Section III reviews work related to the 6TiSCH-network formation. Section IV then describes our extensive simulative study of the network formation phase. We discuss there the results obtained from the simulative study and we offer guidelines for improvements. Finally, Section V draws some conclusions and gives an outlook to future work.

II. Overview of IIoT-stack

The IETF, concerned with the evolution of the Internet architecture, nowadays also looks into the industrial automation processes. The contributions of a variety of IETF activities, initiated during the last ten years, enable now the replacement of proprietary standards by means of an open source protocol stack, as depicted in Fig. 4. Low-power and wireless devices form a multi-hop network using TSCH and the IPv6 Routing Protocol for Low power and Lossy Netwoks (RPL) [9]. 6LoWPAN works as adaptation layer for transmitting IPv6

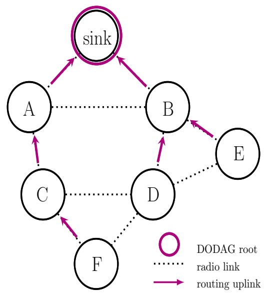


Fig. 1. Multi-hop network

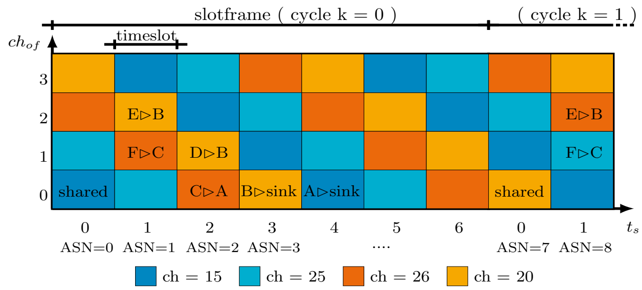


Fig. 2. A possible TSCH-schedule

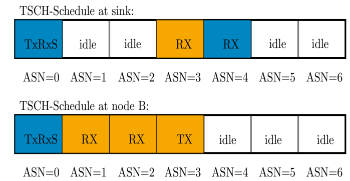


Fig. 3. Locally scheduled activities

packets in an IEEE 802.15.4 network and CoAP acts a UDP-based web transfer protocol between constrained end-points. In the following we describe those building blocks in detail.

A. *Time Slotted Channel Hopping*

Time Slotted Channel Hopping (TSCH) is a MAC mode specified in IEEE 802.15.4-2015 [10] to offer industrial performance regarding power consumption and reliability. TSCH borrows key elements of WirelessHART and ISA100.1 as explained below. In TSCH, time is organised as a continuously repeating sequence of slotframe formed by several timeslots, typically 10 ms long. In each timeslot, a node may transmit or receive a frame, or it may turn its radio off for saving energy. A value shared by all nodes in the network, called Asynchronous Sequence Number (ASN), labels each timeslot. In particular, $ASN = k \cdot N_s + t_s$ counts the total number of timeslots elapsed since the start of the network, where k defines the slotframe cycle, N_s is the slotframe size and t_s points out a timeslot in one slotframe. Up to $N_c \leq 16$ different physical frequencies are available for transmission at each timeslot. As a result, TSCH provides a matrix of links (or cells) for scheduling communications in the network, where each link can be identified by a pair, $[t_s, ch_{of}]$, specifying the timeslot t_s in the slotframe and the channel offset ch_{of} used in that timeslot. The channel offset translates into a physical frequency as follows:

$$f = F\left[(ASN + ch_{of})\, mod N_c\right] \tag{1}$$

The function F can be implemented as a lookup table. Simultaneous communications can take place without interfering in the same timeslot and so the network capacity is increased. In addition, Eq.(1) returns a different frequency for the same link at successive slotframes, following a pseudo-random hopping pattern. This is an efficient way to minimise the effect of multipath fading and external interference. Each link allows a node to send a frame, and if expected, to receive the related acknowledgement (ACK). Links can be *dedicated* or *shared*. Dedicated links are allocated to a single sender-receiver couple and are contention free. On the other hand, CSMA-CA regulates the transmission on shared links.

Let us consider a possible network TSCH schedule illustrated as a matrix of links in Fig. 2. The number of channel offset, i.e., the height of the matrix, is equal to the number of available frequencies ($N_c = 4$) and $N_s = 7$ is the number of timeslots in a slotframe, i.e. the width of the matrix. In particular, $[0, 0]$ is a shared link, allocated for broadcasting frame and used by more than one transmitter node. Furthermore, simultaneous communications happen at timeslot $t_s = 1$ and $t_s = 2$. Each node can translate this TSCH schedule into a local slotframe, where scheduled activities (transmit, receive or sleep) repeat over time, as shown in Fig. 3 for the sink and node B. In particular, node B is only active in four of seven timeslots, resulting in a (atypical) duty-cycle of about 57%.

B. *Routing protocol RPL*

RPL is a distance-vector protocol designed by IETF ROLL working group to operate on top of low power and lossy networks (LLNs), where energy, computation and bandwidth resources are very constrained, and communication is prone to high error rates [9]. RPL is a gradient-based routing that organises nodes as a Destination Oriented Directed Acyclic Graph (DODAG). The DODAG is a directed tree, rooted at the sink, which is usually responsible for data collection. The gradient is called *rank*, and it encodes the distance of each node from the sink, as specified by an Objective Function. The Objective Function offers a flexible way to optimise the network topology, defining which metrics and how these are used to compute a node's rank. Exchanging signaling information, each node can choose a set of parents (nodes with lower rank) among neighbours and select one as its preferred parent, which is the next-hop on a path towards the sink. Section II-D explains the DODAG formation procedure and its interplay with TSCH networks synchronisation, as defined by 6TiSCH in [8].

C. *IETF 6TiSCH*

The 6TiSCH-WG of the IETF aims to face the problem of building and maintaining multi-hop schedules for RPL-organized, TSCH-based networks. To this end, it is necessary to enhance the IEEE standard, which just defines how a node executes a given TSCH schedule, and not the link allocation mechanism, i.e., how to build a TSCH communication schedule [7].

A 6top protocol is being currently defined as a sublayer to fill this gap. Such sublayer is responsible for (1) negotiating the (de)allocation of communication links between nodes,

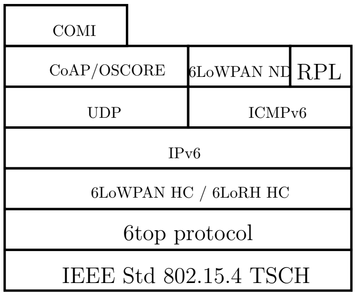


Fig. 4. IETF-Stack for IIoTs

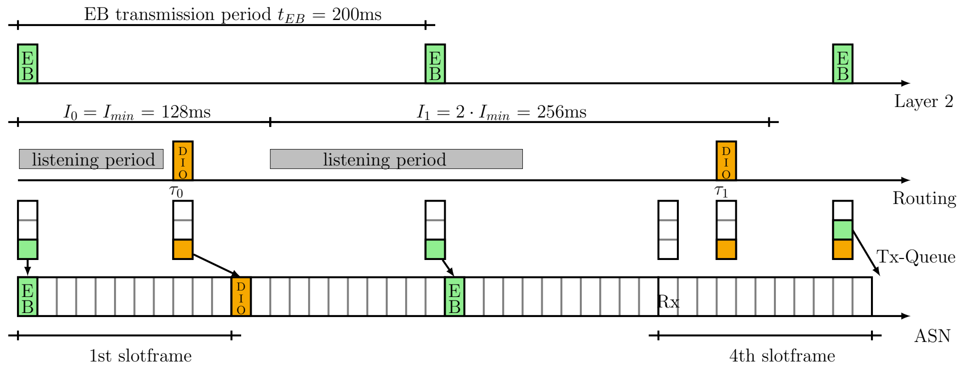


Fig. 5. A timeline of the network formation on an advertiser node

(2) monitoring performance and (3) collecting statistics. It provides a set of commands to support decentralised, centralised and hybrid scheduling solutions [11].

Furthermore, standardisation activities in the 6TiSCH community cover the definition of a bootstrapping protocol, by which a new mote is (optional securely) admitted into the multi-hop network. In Section II-D, we describe the way a 6TiSCH-network is formed. The secure join protocol, by which a node requests the admission into the network and sets up keys used to authenticate and encrypt subsequent transmission [12], is out of scope for this paper.

D. Network formation procedure

When nodes run TSCH and RPL protocols, they have at least to synchronise on a slotframe structure and to join a DODAG, before they can send data packets. We call this process network formation. It starts with the sink (or root), which advertises the network presence, ends when every node has selected its preferred parent, and involves MAC and routing layers. For network advertising, at least two kind of signaling information are sent: Enhanced Beacon (EB) frame and DODAG Information Element (DIO) packets. An EB contains all the necessary time information to allow the initial synchronisation among nodes. A DIO packet is an ICMPv6 control message, which announces the DODAG-ID, the sender's rank and other configuration parameters used for DODAG construction and maintenance.

IEEE 802.15.4-2015 specifies the frame format and the function of EBs. However, it does not detail a transmission strategy for these messages. 6TiSCH has been recently working on the definition of a set of conventions to build the network, assuring essential interoperability between nodes, and standardised the 6TiSCH-MC in [8]. A static communication schedule, referred as *minimal schedule* should enable the network formation, using only a single shared slot and a tunable number of timeslots, set by the sink. Besides, 6TiSCH-MC defines how TSCH interacts with RPL during the network formation, as hereafter described.

The root starts the process broadcasting EBs and DIOs in its neighbourhood. EB are emitted every t_{eb}, a tunable parameter which influences the network formation time and its energy consumption. In RPL, the Trickle algorithm controls the generation of DIO messages as specified in [13]. The time is split into intervals of size I and the root schedules transmission of DIO messages at a random instant τ in the second half of each interval. The size I is varied over the time. In particular, the root doubles I at the end of each interval, starting from the minimum size I_{min}, until a maximum value $I_{max} = 2^M \cdot I_{min}$ is reached. The sink exploits the shared slots, offered every $N_s \cdot t_s$ by the *minimal schedule*, to transmit EBs and DIOs located in the outgoing queue. This process is exemplified in Fig. 5, where $t_{eb} = 200ms$, $I_{min} = 128ms$, and the *minimal schedule* are used. After being generated, EBs and DIOs are put in the transmit queue and consumed by TSCH in the next active slot. As illustrated, EBs are queued with a priority higher than DIOs.

When a node wishes to join the network, it uses preferably passive scanning, i.e., it turns the radio on to a randomly chosen frequency, and it listens for EBs. While waiting for a valid EB, the joining node keeps its radio always on and changes frequency every t_{scan}. After hearing a valid EB, a node learns the *minimal schedule* in the network (i.e., ASN, the timeslot timing, slotframe length, number of available frequencies), so it knows when to wake up for receiving or sending signalling frames related to the network formation procedure. The joining nodes will not start advertising the network presence straight away, but only after it has received a DIO message, computed its rank and selected its preferred parent among a set. This expedient assures a multi-hop time synchronisation with a loop-less structure since it reuses the DODAG structure with no extra signalling at MAC layer [14]. Nodes regularly re-synchronize their clock upon receiving a frame or an ACK from their parents. When that does not happen within a t_{ka} period, a node sends keep-alive messages (KAs) to trigger a clock update from its time source. Optionally, a newly synchronised node may transmit a DODAG Informational Solicitation (DIS) to ask advertisers in the neighbourhood for a DIO and to speed up the joining.

This process spreads gradually to cover the whole network: each node may become an advertiser node after hearing an EB and after joining a DODAG. Each advertiser nodes will periodically send EBs. Besides, the Trickle timer may suppress some scheduled transmission of DIO messages, e.g. when a node has received more than c DIO messages with redundant information, or it resets the interval size I to the minimal value in case of reception of an inconsistent message or a DIS.

All control traffic listed above is transmitted within the single shared timeslot provided by the *minimal schedule*. As more and more nodes join, collisions are likely to occur.

Our description of those interacting mechanism illustrate that answering the question *how long the 6TiSCH-network formation process will take?* is not trivial. Obviously, the size of the network has an impact. Moreover, there is an interaction between TSCH and RPL protocols and their parameters in every phases (scanning for EBs, synchronising, joining a DODAG). In Section IV we study this interplay and aim at identifying the factors with the highest impact using simulations.

III. Related Works

The classic IEEE 802.15.4 MAC modes, specified in [15] - [16], RPL and the 6TiSCH-architecture have already been subjects of several studies in the literature in the last years, and various aspects have been analysed so far. To reasonably bound the number of related works, we review in this section only works that focus on or examine (in the evaluation) the initial network formation. We consider related work from three different areas: (1) Network formation with IEEE 802.15.4 and RPL, (2) Initial TSCH synchronisation, and (3) 6TiSCH.

Network formation with classic IEEE 802.15.4 and RPL Authors in [17] evaluate the control overhead of RPL using the Cooja simulator. Results show a transient phase for the network bootstrap of about 10 minutes, considering the time required to let the overhead significantly drop. That work considers fixed values of the Trickle timer parameters (e.g., $I_{min} = 4, 1s$), random topologies composed of 20 or 100 nodes and a UDGM propagation model. A more comprehensive insight on the influence of RPL parameters and network characteristics on the network formation time is presented in [18], using OMNeT++ as simulator. We use a similar methodology, however considering different network scenarios and only a fixed value $c = 5$ as RPL redundancy threshold. The limitations of a strict separation between the IEEE 802.15.4 MAC and routing protocol RPL, especially during the formation of a multi-hop wireless network, are presented in [19], [20] and [21]. In those works the advantages using a cross-layer optimization are highlighted. Pavkovic et al. [19] and Vučinić et al. [20] address the incompatibility of the IEEE 802.15.4 cluster-tree and the DODAG. To resolve this, a new cluster-DAG structure for IEEE 802.15.4 is stated in [19] and an interplay scheme between IEEE 802.15.4 and RPL during network formation is proposed in [20]. The work of Iova et al. [21] recommends among other things that the MAC protocol does not impose a topology to the network, but only filters out bad links. The 6TiSCH-MC, evaluated in this paper, follows this recommendation, since the DODAG construction relies on broadcast control packets, which are exchanged over shared slots, where all node stay awake.

Initial TSCH synchronisation The initial global synchronisation in a TSCH network is reached when all nodes have received an EB. The standard IEEE 802.15.4-2015 [10] does not detail any beacon scheduling strategy, although it strongly influences the time and energy devoted for the synchronisation process. The authors of [22], [23], [24] propose different scheduling algorithms, which are characterized by a given set of advertising slot, i.e. the number of shared slot per slotframe, used for broadcasting traffic, is greater as one. Where the N_b advertising slots are located and how these are selected by each advertiser node depends on the specific algorithm. In [22] the PAN coordinator initially calculates the links to be used for EB transmission solving an optimisation problem. In [23] multiple consecutive slotframe are grouped to form a multi-slotframe, and the advertised slots are located either in the first timeslot of a multi-superframe (using different channel offsets) or in the first timeslot of every slotframe in the multi-superframe (but only using $ch_{of} = 0$). In both works, each advertiser node randomly chooses a link from the set, and there is no guarantee, that a slot will not be taken twice by neighbouring nodes, causing EB collisions. Instead, the algorithm proposed in [24] ensures that beacon transmissions take place on all frequencies used by the TSCH network, regularly and without collision. However, those three approaches are not compliant with the 6TiSCH-MC, impose a higher duty-cycle, and require additional management overhead. Wang et al. [25] examine how quickly a joining node synchronises to an existing TSCH network when the 6TiSCH-MC is used. They present results achieved by simulations with Cooja, and they highlight the impact of EB period, namely t_{eb}, and neighbourhood size on the synchronisation time. In our work, we extend this study considering the parameters listed in Table I and the activation of all nodes at the same time.

6TiSCH. Righetti et al. [26] evaluate the performance of the *6top* protocol, defined in 6TiSCH, considering the exchange of the messages to allocate dedicated timeslots at bootstrap. As in our contribution, this work gives guidelines on tuning TSCH parameters, e.g. at least two shared timeslots for a successful allocation by each node, and considers the interplay with RPL. However, the Trickle parameters are fixed to the default value. The limit of the 6TiSCH-MC in dense networks is considered as well as a factual issue by Vučinić et al. [27] and motivates there the adaptation of the Bayesian broadcast algorithm. This algorithm optimises the network formation process, setting (locally and dynamically) different transmission probabilities for each type of traffic, i.e. EBs, DIOs or bootstrapping. Our work, in contrast, aims to extensively investigate the issue with the *minimal schedule* in relation to crucial parameters of TSCH and RPL protocols and to different typical network topologies. In this sense, it complements and confirms the performance evaluation recently carried out in [28], where only the slotframe size is varied.

IV. Performance Evaluation

We present here the methodology used to evaluate the network formation process and our principal findings.

A. Methodology

Contiki [29] is an open source operating system created to run on constrained nodes. A world-wide active community of developers in both academia and industry has been contributing to the project and regularly updates the code for supporting IETF 6TiSCH protocols stack. Since the end of 2015, an implementation of TSCH access mode and of the 6TiSCH-MC is available and it has been extensively evaluated [30], [31][2]. Contiki includes a network simulator, Cooja, which allows the emulation of binary code produced by the Contiki toolchain, before burning it into real hardware, and to simulate different wireless propagation models. For reasons explained above, we evaluate the performance of the initial 6TiSCH-network formation using Cooja.

In the simulations, we use the Cooja mote type, i.e., a virtual hardware without limitations concerning memory and computation capabilities. As wireless propagation model, we use the Unit Disk Graph Medium (UDG). Links are symmetric, the packet delivery ratio (PDR) is 100% in a transmission range of 50m, and the interference range is 100 m. We did not consider a realistic hardware and radio setting to narrow down the side effects and to study the protocol mechanisms alone.

Similarly to [26] and [18], we consider different network sizes ($N_{size} \in \{9, 16, 25\}$) and three categories of network topology:

- *Grid*: the sink is placed in the left-high corner and the distance between nodes is set, so that the degree of the central node is $d = 4$ with $N_{size} = 9$, or $d = 6$ with $N_{size} = 16$ and 25.
- *Ellipse*: for every node is degree $d = 2$, i.e., each node has a PDR of 100% only with the left and right one-hop neighbours, and the multi-hop path from a sensor node to the sink has a maximum depth $maxD = \lfloor N_{size} \ / \ 2 \rfloor$.
- *Random*: nodes are randomly placed in a square area of $100 \, m^2$. The topology features a portion of the network with high density and other some nodes with a minimum degree $d = 2$.

For each simulated setting, i. e. a possible combination of parameter's value, we run 50 independent replications with different seeds, and we report the average value of each metric with its 95% confidence interval.

The configuration setting of TSCH and RPL are reported in Table I. For the evaluation, we study the effects of TSCH EB period t_{eb}, TSCH number of channel N_c and RPL minimum interval I_{min} on the following performance metrics:

- *TSCH synchronisation time*, defined as the time between the transmission of the first EB by the PAN coordinator and the first network-wide TSCH synchronisation. At this time, all nodes have learned the minimal schedule, and each node is synchronised with its time source neighbour.
- *DODAG formation time*, defined as the time until all nodes are simultaneously in the DODAG. The period between TSCH synchronisation time and DODAG formation time is not only characterised by the waiting for the first DIO message at the last synchronised node, but also by possible TSCH desynchronisations, so that some nodes may temporarily leave the 6TiSCH network.
- *Number of exchanged control frames*, defined as the number of EBs, DIOs, DIS and KAs transmitted during the DODAG formation time.
- *Charge consumed* during the network formation, estimated summing the different charge consumed by the radio component of each node when they are active (e.g., transmission or reception activities) and inactive.

[2]The development team is now working on full support for IETF 6TiSCH specifications in the project Contiki-NG [32].

TABLE I
SETTING FOR TSCH- AND RPL-PARAMETERS

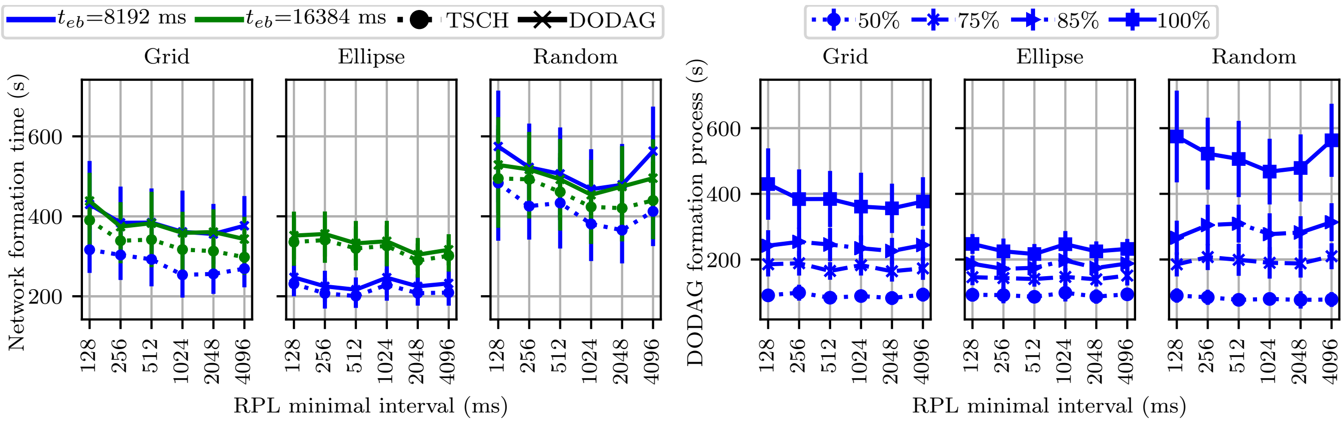
Parameter	Symbol	Value
TSCH slotframe length	N_s	101
TSCH timeslot duration	t_s	10 s
TSCH scan interval	t_{scan}	1 s
TSCH number of channel	N_c	$4, 16$
TSCH KA period	t_{ka}	$\{12, ..., 60\}$ s
TSCH EB period	t_{eb}	$\{2048, 4096, 8192, 16384\}$ ms
RPL redundancy constant	c	5
RPL interval doubling	M	8
RPL minimum interval	I_{min}	$\{128, 256, ..., 4096\}$ ms

TABLE II
SUCCESSFUL (%) DODAG FORMATIONS WITHIN 30 MIN

		Grid			Ellipse			Random		
		N_{size}			N_{size}			N_{size}		
t_{eb}	N_c	9	16	25	9	16	25	9	16	25
2048	4	0	0	0	70	16	0	0	0	0
	16	0	0	0	2	2	0	0	0	0
4096	4	84	0	0	100	100	86	46	0	0
	16	12	0	0	100	100	26	2	0	0
8192	4	100	20	0	100	100	100	100	2	0
	16	96	0	0	100	98	40	66	0	0
16384	4	100	70	0	100	100	100	100	14	0
	16	98	4	0	94	34	0	92	0	0

B. Simulation of 6TiSCH-MC

This first set of simulations were performed to evaluate the performance and the limits of using *minimal schedule* in different topologies. As stated in Table I, we do not strictly use default values for the Trickle timer. Specifically, the default value $I_{min,RFC} = 8ms$, together with $t_s = 10ms$ and $N_s = 101$, would cause queue drops in the firsts Trickle intervals. Besides, we set the redundancy constant $c = 5$, i. e. the half of the default value, since almost all nodes have a degree lower than $c = 5$ in the considered topologies, and hence $c \geq 5$ does not significantly affect the network formation time (see also [18]). Furthermore, advertiser nodes generate EB frame with a random EB period in the value ranges between $t_{eb} \cdot [0.75, 1)$. Although the 6TiSCH-MC does not mention this mechanism, open-source implementations of the TSCH (e.g. OpenWSN and Contiki) include it, and [25] shows its advantages in the network synchronisation. In configurations with $N_c = 4$, we assume that only a subset of channels is available in the network (see e.g. Fig. 2) and that the joining nodes are aware of this subset.



(a) TSCH synchronisation time and DODAG formation time

(b) DODAG formation progress

Fig. 6. With $N_c = 4$ and $N_{size} = 9$, we recognised the lowest average DODAG formation time of about 216 s in the ellipse topology. For the grid and the random scenarios the average lowest DODAG formation time are respectively 355 s and 467 s.

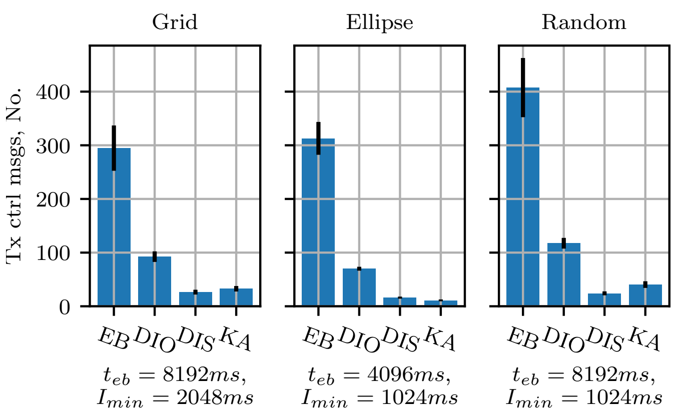


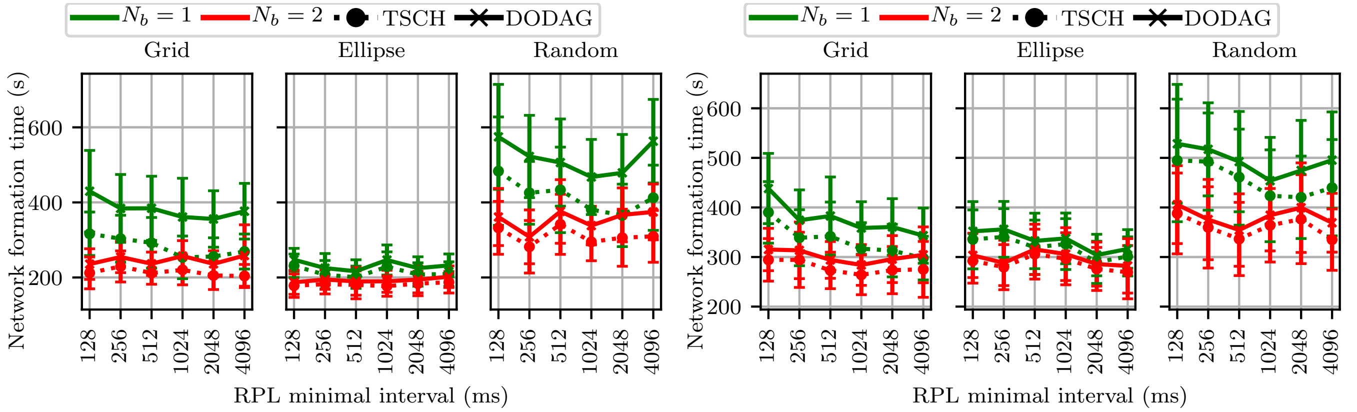
Fig. 7. Art and average number of control frames exchanged until all nodes are simultaneously in the DODAG. For each topology ($N_{size} = 9$), we select the best parameter setting concerning the network formation time.

Table II reports on the number of simulations in which the network formation phase is completed within 30 minutes. As can be seen, an improper choice of t_{eb} and the number of channel offset N_c in relation to the network density (i.e.topology) can lead to an unsuccessful network formation, where at least one node is even after 30 minutes not yet operational and cannot transmit any sensed data to the sink. With only one shared slot for broadcasting control frame and $t_{eb} < 4 \cdot N_s \cdot t_s$, the positive effect of applying randomness to the EB periods cannot take place, i.e. advertiser nodes choose with high probability the same slotframe for sending EBs, causing their collision. Besides, also local contentions between EB and other control packets (e.g. DIO, DIS and KA) are critical with this aforementioned setting. These two problems become more and more apparent as the network density increases. Compared to a setting with $N_c = 4$, we observed a linear increase of the network formation time, when $N_c = 16$ is used. For these reasons, we will consider only results obtained with $N_c = 4$ and $N_{size} = 9$ in the reminder of the section.

In Fig. 6a the *TSCH synchronisation time* and the *DODAG formation time* are expressed as a function of the RPL minimum interval. As it can be seen, I_{min} shows an influence on the network formation time of random and grid topologies, which is worth investigating. The behaviour is pretty different from the results presented in [18] for IEEE 802.15.4-2011, where decreasing the RPL minimum interval yields to a reduction of DODAG formation time. One reason is that the configuration of the TSCH slotframe interferes with the Trickle mechanism and implicitly sets a range for the optimal value of I_{min}. As above mentioned, when $I_{min} << N_s \cdot t_s$ several DIO messages are dropped in the transmission queues. We do not appreciate the influence of the RPL minimum interval when we deal with a sparse topology as the ellipse. This fact can be explained considering the limited effect of the polite gossip policy (used by Trickle) on this type of network.

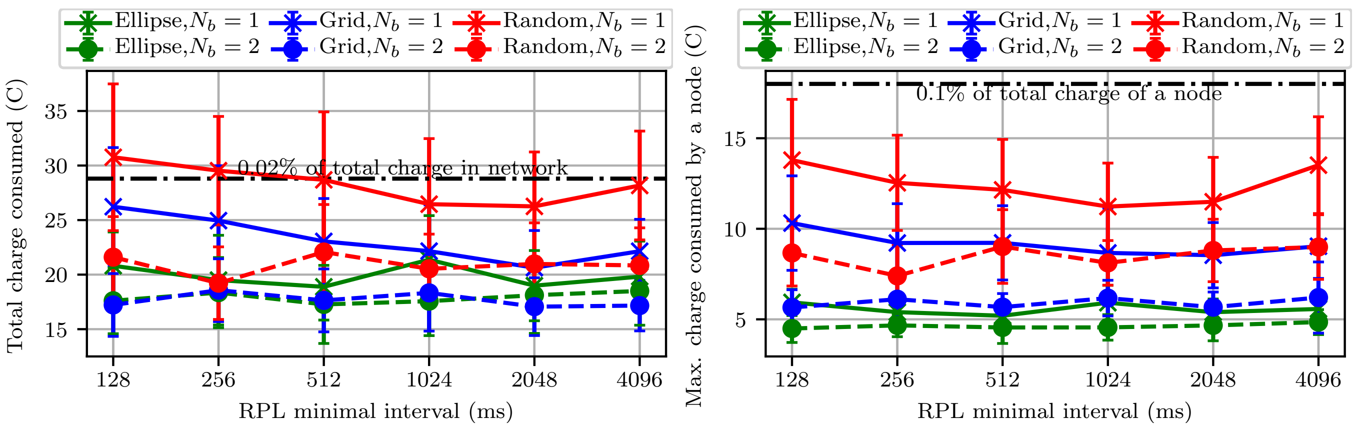
Another insight from Fig. 6a is the significantly different elapsed time between the network-wide TSCH synchronisation and the DODAG completion, which can be observed in the three topologies. The higher physical density in the random or grid topologies makes collision of EB or KA frames more likely to occur than in the ellipse network, especially when the number of joined nodes increase. Consequently, desynchronisation events are more likely to happen (and they was observed) in such topologies and cause an additional delay of about 100 s , i.e. circa 100 minimal slotframe, after the network-wide TSCH synchronisation. Fig. 6b, which reports on the dynamic of the DODAG formation process in the three scenarios, confirms this behaviour. The time, until half of all nodes have joined the DODAG, is approximately the same in all scenarios. In the second half of the network formation, the results show how the minimal schedule slows down the completion of the process in dense network.

We can see in Fig. 7 how EBs are the dominant element in all networks. The introduction of an EB transmission strategy, where the EB period is mapped to the Trickle interval, has the potential of reducing the control overhead, but also the drawback of hindering the synchronisation of new nodes [27]. The valuable number of KA in the grid and random scenarios can be explained considering the collisions of frames from



(a) (b)

Fig. 8. Time saving in the network formation process by allocating $N_b = 2$ shared slots for the transmission of EBs and RPL messages with $N_c = 4$, $N_{size} = 9$ and $t_{eb} = 8192\,ms$ (a) or $t_{eb} = 16384\,ms$ (b).



(a) Total charge consumption

(b) Highest charge consumption in a node

Fig. 9. We assume powering every node (except the sink) with a 2xAA battery pack, which carries 18000C (2x2500 mAh) and we select as parameter set $N_{size} = 9$, $t_{eb} = 8192\,ms$, $N_c = 4$. We use as current consumption the values inferred from the datasheet of the TI's CC2538 [33], a wireless microcontroller System-on-Chip commonly used for IEEE 802.15.4 applications.

neighbouring nodes in these topologies.

C. *Adding additional links to 6TiSCH-MC*

In this experiment, we investigate the influence of an additional slot in the *minimal schedule* on the time and energy consumed for the network formation. We follow the guideline fixed in [26] allocating $N_b = 2$ shared slots in the basic schedule. The second slot will be used, if any frame in the transmission queue is present after the first timeslot, but none scheduling algorithm applies. The first remark is that the number of configurations afflicted by unsuccessful network formation decreases. This is an expected behaviour since with a higher value of N_b there is a reduction of local contentions between control packets. Fig. 8b shows a valuable reduction in the average time spent for the network formation. It is also interesting to observe, how the time gap between TSCH synchronisation and DODAG completion shrinks. The 6TiSCH-MC recommends that EBs are queued with a higher priority than messages coming from higher layers. When only one shared slot is available, the transmission of RPL packets, generated by the Trickle algorithm, may be significantly delayed or dropped, causing the behaviour described in Section IV-B. With $N_b = 2$, there is a more efficient messaging and, therefore, a quicker transition from the state "synchronized" to "advertiser" in every joined node.

The drawback of adding links to the *minimal schedule* is a higher "basic" duty-cycle, since every node is active at least in these broadcast slots. For example, with $N_b = 2$ and $N_s = 101$ results a duty cycle of about 0.2%. However, the significant reduction of the time spent for network formation causes valuable energy savings, especially in grid and random topologies, as we can see in Fig. 9a and 9b, which present an estimation of the charge consumed respectively for the whole network and for the node with the highest consumption. This fact is consequence of the less time spent in waiting for EB by an activated node and of the reduced number of collisions

D. *Recommandations*

This simulation study allows us to derive a set of guidelines on how to choose the TSCH and RPL parameters for a given topology. We recommend implementers of 6TiSCH-network to set the I_{min} parameter slightly over the minimal slotframe duration, while the t_{eb} should be set to a value $t_{eb} \geq 4 \cdot N_s \cdot t_s$, so that the effect of applying randomness to the EB periods can take place. In dense topologies, the number of shared slots used for broadcasting is a crucial factor, and the use of at least $N_b = 2$ is recommended. The duplicated duty-cycle is compensated by a reduced average joining time of each single node, and by a more likely successful delivery of frame in the whole network.

V. CONCLUSIONS

The protocol architecture proposed by IETF activities has the potential of revolutionising the communication technology adopted for industrial automation. In this architecture, TSCH for medium access control and RPL for routing are core concepts. We described how a low power wireless multi-hop network is formed, i.e., how its presence is announced and how each mote joins to it, as defined by the IETF 6TiSCH-WG.

Through simulation, we reported on the limits of the proposed 6TiSCH-MC and on the risk of its blind adoption. Even with small network size, a wrong parameter configuration would cause an unsuccessful synchronisation of nodes within 30 minutes, that means lost operational time and discharged nodes. Among other offered guidelines, we recommend implementers to allocate additional shared slots in the TSCH-schedule responsible for network formation. This approach shows significant time and energy savings during the network formation phase, especially in grid and random topologies. That is a consequence of (1) a reduced number of collisions, (2) a less local contention and (3) a better interplay between TSCH and RPL.

In future works, we plan to simulate a more realistic wireless propagation model and to experiment with testbeds. We aim at investigating the trade-off between the network formation time and the duty-cycle of nodes, and at defining a solution for a dynamic, decentralised and collision-avoiding allocation of the $N_b \geq 2$ TSCH-links used for sending broadcasting frames.

REFERENCES

[1] *Industrial networks - Wireless communication network and communication profiles - WirelessHART*, IEC Std. IEC 62 591:20 016, 2016.

[2] *Wireless systems for industrial automation: Process control and related applications*, ANSI/ISA Std. ANSI/ISA-100.11a-2011, 2011.

[3] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli *et al.*, "Industrial wireless ip-based cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1025–1038, 2016.

[4] R. Yu. (2017) Security and reliability are key in wireless networks for industrial iot. [Online]. Available: http://www.analog.com/media/en/technical-documentation

[5] FraunhoferESK. (2015) Survey: Utilization of wireless technologies in german industry. [Online]. Available: https://www.esk.fraunhofer.de/en/publications/studies/surveywireless.html/

[6] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia *et al.*, "Standardized protocol stack for the internet of (important) things," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[7] P. Thubert and T. Watteyne. (2013) Ipv6 over the tsch mode of ieee 802.15.4e. [Online]. Available: https://datatracker.ietf.org/wg/6tisch/about/

[8] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal ipv6 over the tsch mode of ieee 802.15. 4e (6tisch) configuration," RFC 8180, 2017.

[9] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," RFC 6550, 2012.

[10] *IEEE Standard for Low-Rate Wireless Networks (WPANs).*, IEEE Std. 802.15.4-2015, 2016.

[11] Q. Wang, X. Vilajosana, and T. Watteyne, "6top protocol (6p)," IETF Draft, 2018.

[12] M. Vučinić, J. Simon, K. Pister, and M. Richardson, "Minimal security framework for 6tisch," IETF Draft, 2018.

[13] P. Levis and T. H. Clausen, "The trickle algorithm," RFC 6206, 2011.

[14] T. Chang, T. Watteyne, K. Pister, and Q. Wang, "Adaptive synchronization in multi-hop tsch networks," *Computer Networks*, vol. 76, pp. 165–176, 2015.

[15] *IEEE Standard for Low-Rate Wireless Networks (WPANs).*, IEEE Std. 802.15.4-2006, 2006.

[16] *IEEE Standard for Low-Rate Wireless Networks (WPANs).*, IEEE Std. 802.15.4-2011, 2011.

[17] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the rpl routing protocol," in *Mechatronics (ICM), 2011 IEEE International Conference on*. IEEE, 2011, pp. 767–772.

[18] H. Kermajani and C. Gomez, "On the network convergence process in rpl over ieee 802.15. 4 multihop networks: Improvement and trade-offs," *Sensors*, vol. 14, no. 7, pp. 11 993–12 022, 2014.

[19] B. Pavkovic, A. Duda, W.-J. Hwang, and F. Theoleyre, "Efficient topology construction for rpl over ieee 802.15. 4 in wireless sensor networks," *Ad Hoc Networks*, vol. 15, pp. 25–38, 2014.

[20] M. Vučinić, G. Romaniello, L. Guelorget, B. Tourancheau, F. Rousseau, O. Alphand *et al.*, "Topology construction in rpl networks over beacon-enabled 802.15. 4," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–7.

[21] O. Iova, F. Theoleyre, T. Watteyne, and T. Noel, "The love-hate relationship between ieee 802.15. 4 and rpl," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 188–194, 2017.

[22] D. De Guglielmo, S. Brienza, and G. Anastasi, "A model-based beacon scheduling algorithm for ieee 802.15. 4e tsch networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*. IEEE, 2016, pp. 1–9.

[23] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast join and synchronization schema in the ieee 802.15. 4e mac," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*. IEEE, 2015, pp. 85–90.

[24] I. Khoufi, P. Minet, and B. Rmili, "Beacon advertising in an ieee 802.15. 4e tsch network for space launch vehicles," in *EUCASS 2017-7th European Conference for Aeronautics and Aerospace Sciences*, 2017.

[25] L. Wang and A. Reinhardt, "A simulative study of network association delays in ieee 802.15. 4e tsch networks," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*. IEEE, 2017, pp. 1–3.

[26] F. Righetti, C. Vallati, G. Anastasi, and S. Das, "Performance evaluation the 6top protocol and analysis of its interplay with routing," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[27] M. Vučinić, T. Watteyne, and X. Vilajosana, "Broadcasting strategies in 6tisch networks," *Internet Technology Letters*, vol. 1, no. 1, p. e15, 2018.

[28] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving network formation in 6tisch networks," *IEEE Transactions on Mobile Computing*, in press.

[29] A. Dunkels. (2002) Contiki: The open source os for the internet of things. [Online]. Available: http://www.contiki-os.org/

[30] S. Duquennoy, A. Elsts, A. Nahas, and G. Oikonomou, "Tsch and 6tisch for contiki: Challenges, design and evaluation," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2015)*, 2017.

[31] M. R. Palattella, X. Vilajosana, T. Chang, M. A. R. Ortega, and T. Watteyne, "Lessons learned from the 6tisch plugtests," in *International Internet of Things Summit*. Springer, 2015, pp. 415–426.

[32] S. Duquennoy. (2017) Contiki-ng: the next generation contiki. [Online]. Available: http://www.contiki-ng.org/

[33] TexasInstruments. (2015) Cc2538: A powerful system-on-chip for ieee 802.15.4 applications. [Online]. Available: http://www.ti.com/product/CC2538/