



Piyare, R., Oikonomou, G., & Elsts, A. (2020). TSCH for Long Range Low Data Rate Applications. *IEEE Access*, 8, 228754-228766.
<https://doi.org/10.1109/ACCESS.2020.3046769>

Publisher's PDF, also known as Version of record

License (if available):
CC BY

Link to published version (if available):
[10.1109/ACCESS.2020.3046769](https://doi.org/10.1109/ACCESS.2020.3046769)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the final published version of the article (version of record). It first appeared online via IEEE at <https://doi.org/10.1109/ACCESS.2020.3046769>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Received November 24, 2020, accepted December 13, 2020, date of publication December 23, 2020, date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046769

TSCH for Long Range Low Data Rate Applications

RAJEEV PIYARE¹, (Member, IEEE), GEORGE OIKONOMOU², AND ATIS ELSTS³, (Member, IEEE)

¹Amber Agriculture, Chicago, IL 60610, USA

²Department of Electronic and Electrical Engineering, University of Bristol, Bristol BS8 1TL, U.K.

³Institute of Electronics and Computer Science, LV-1006 Riga, Latvia

Corresponding author: Atis Elsts (atis.elsts@edi.lv)

ABSTRACT The TSCH (Time Slotted Channel Hopping) protocol from the IEEE 802.15.4-2015 standard is known to be suitable for highly reliable applications in low-power networks of severely constrained wireless embedded devices. Most of the research on TSCH has focused on the 2.4 GHz frequency band. The present work extends the TSCH protocol to low data rate applications using the sub-GHz frequency bands for an increased link budget. We introduce multiple improvements on top of the standard TSCH, namely, a special schedule for the network's root nodes and their direct neighbors, as well as the option to have multiple root nodes in a single TSCH network. Experimental results in a testbed and a real-world deployment show that after applying the improvements, the network meets application requirements and provides reliable and energy-efficient operation.

INDEX TERMS TSCH, Internet of Things (IoT), Contiki-NG, IEEE 802.15.4, MAC protocol, low-power and lossy network (LLN).

I. INTRODUCTION

The TSCH (Time Slotted Channel Hopping) protocol from the IEEE 802.15.4-2015 standard [1] is known to be suitable for highly reliable applications in low-power wireless networks of constrained embedded devices. Most of the existing research on TSCH has focused on the crowded 2.4 GHz frequency band [2]–[5]. For applications that require long-range links, such as agriculture applications, using a sub-GHz physical layer is a more attractive option.

In this paper, we focus on an environmental monitoring application for the grain industry. The goal of the application is to sense temperature and humidity inside grain during its storage and provide real-time information to the farmer about these parameters, as they impact the price of the grain and quality of the food eventually prepared from the grain. Gateway nodes placed outside the grain storage pick up this information and forward it to the Internet over a 3G/4G connection. The grain is a poor radio propagation medium, as the individual grains scatter the radio signal a lot, and also shadow the signal due to their significant water content. To make within-grain sensor networks feasible, specific technological solutions must be applied, including the use of multihop networks and sub-GHz communications for better signal penetration.

The associate editor coordinating the review of this manuscript and approving it for publication was Sathish Kumar^{1b}.

However, reducing the communication frequency not only increases the link budget, but also opens up a number of challenges:

- Transmitting each TSCH packet takes longer; consequently, a new TSCH timeslot timing template must be designed and evaluated.
- The physical layer data rate is reduced; consequently, the link capacity is smaller as well.
- Multiple data collection points may be within the range of a single network and serve as root nodes.

In this paper, we show that a widely used TSCH scheduler called Orchestra [2] has problems when operating under limited link capacity conditions, especially in regions close to the network's root node. Furthermore, Orchestra does not support multiple root nodes in a single network, motivating further research for alternative solutions.

In this context, the contributions of this work are:

- We develop and experimentally evaluate a Contiki-NG TSCH port to the Texas Instruments CC1310 System-on-Chip operating in a sub-GHz frequency band and we provide a TSCH timing template for this platform;
- We design and evaluate a rule for a new slotframe in the Orchestra TSCH schedule, which we call the *root rule*;
- We adapt the “virtual root node” concept to TSCH networks and quantify its benefits;

- We discuss the suitability of TSCH for low-data rate applications using a sub-GHz frequency band, and make suggestions for improvements.

Experimental results in a 29-node testbed illustrate that sub-GHz TSCH networks utilizing the proposed root rule achieve low energy consumption (1.6 active slots for reception per second on non-root nodes) and high end-to-end packet delivery rate (99.0%). Results from a pilot deployment confirm the suitability for this application and show >95% end-to-end packet delivery rate in real-world conditions.

The paper is structured as follows. Section II provides background on the relevant technologies and related work. Section III discusses our target application, its requirements, and our technological solutions to cover these requirements. Section IV includes the evaluation results, Section V provides a discussion of the results and sketches future work, and finally, Section VI concludes the paper.

II. BACKGROUND

A. TSCH

The IEEE 802.15.4-2015 standard [1] was published in early 2016. Among other new MAC modes of operation, it introduces the Time-Slotted Channel Hopping (TSCH) protocol. TSCH is a Time Division Multiple Access (TDMA) MAC protocol that combines frequency diversity with scheduled communication. TSCH is designed for industrial low-power wireless network applications that need both high reliability and high energy efficiency. TSCH utilizes pseudo-random channel hopping to combat narrow-band interference and signal fading.

All devices in a TSCH network are time-synchronized to a common source: the network coordinator node. The basic element of a TSCH communication is a *timeslot*; timeslots can be used for packet reception, transmission, or be idle to conserve energy. If an acknowledgment is required, it is sent in the same timeslot as the packet. The nodes in a TSCH network need to agree on a common communication *schedule*, which defines which actions to perform in which timeslots. The schedule of a node consists of one or more periodically repeating *slotframes*. A slotframe is a two-dimensional collection of *cells*; a cell is characterized by its timeslot (time offset relative to the start of the slotframe) and its channel offset. The latter is mapped to a physical channel during run-time of the network using the following formula from the standard [1]:

$$\text{channel} = HS[(ASN + \text{ChannelOffset}) \bmod |HS|],$$

where *HS* is a pseudo-random channel hopping sequence shared by all nodes in the network.

The standard leaves the construction of the schedule to the implementers. In recent years, there have been many proposals for the construction of TSCH schedules. Approaches to scheduling can be either *centralized*, where the schedule is created on a computation element outside the low-power network [6], *distributed*, where the schedule is created via

negotiations between neighbor nodes [7], or *autonomous*, where the schedule is decided by each node autonomously based on the information it already possesses from other network layers. For low-overhead autonomous scheduling, Orchestra [2] is an attractive proposal. It utilizes routing information to allocate and de-allocate TSCH cells. The original Orchestra article [2] describes two modes: sender-based and receiver-based Orchestra. In the sender-based variant, a node allocates a transmission cell for itself and a reception cell for its parent node and each of its direct child nodes. In the receiver-based variant, the node itself has a reception cell, while the other nodes get transmission cells.

Variants and enhancements of Orchestra have been proposed that achieve better performance, for example, through different cell allocation strategies [8] and through using multiple channel offsets [9].

B. RPL

RPL (Routing Protocol for Low-Power and Lossy Networks) protocol [10] is a proactive routing protocol for low-power wireless networks. It is primarily designed for data collection applications (upstream traffic), although it does provide support for downstream traffic as well. RPL networks can have one or more root nodes, although implementations for embedded devices tend to only support a single root device. RPL constructs and dynamically updates the routing DODAG (Destination Oriented Directed Acyclic Graph). To select the parent nodes, RPL uses link quality as estimated by traffic and path quality, combined in a specific objective function, as selected by the configuration. A typical objective function used by RPL is the MRHOF (Minimum Rank with Hysteresis Objective Function) [11], which selects the parents based on the estimated transmission count (ETX) metric.

RPL features several modes of operation: *storing* mode, where the network nodes store their routing tables, and *non-storing* mode, where the routing table is only kept on the root node, while the network nodes only keep a default route with their routing parent node(s) as the next hop(s). Downstream traffic in the non-storing case is sent by source routing. There is also a mode-of-operation with code 0, where no downward routes are maintained by RPL.

RPL forms the DODAG by periodically broadcasting DIO (DODAG Information Object) messages. The period of the DIO messages is dynamically adjusted by a Trickle [12] timer. Downstream routes are created by DAO (Destination Advertisement Object) messages, sent from child nodes to their parents or to the root.

C. SUB-GHz COMMUNICATION BANDS

1) WHY SUB-GHz BAND?

For many Internet of Things (IoT) application such as asset tracking, smart agriculture, environmental monitoring, and manufacturing, the license-free Industrial, Scientific, and Medical (ISM) frequency bands have become a preferred choice of communication as these applications only need to

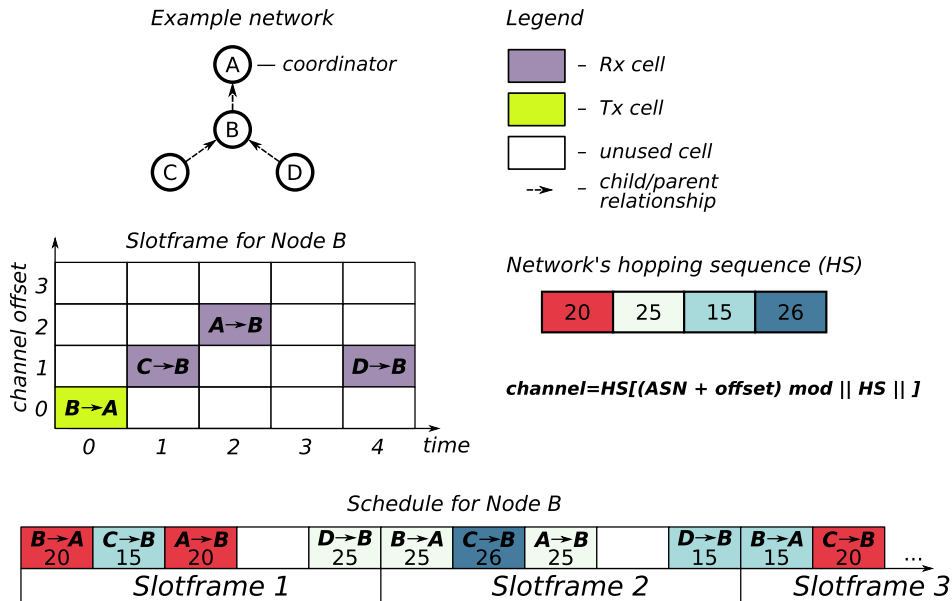


FIGURE 1. Overview of TSCH concepts.

send small amounts of data periodically at low data rates. While there is a range of ISM bands to choose from, the decision often boils down to 2.4 GHz vs. sub-GHz frequencies, i.e., 433, 868, and 915 MHz bands. Firstly, due to its free access, co-existing devices such as Bluetooth and Wi-Fi operating in the same frequency band have led to spectrum crunch making 2.4 GHz more susceptible to interference. In contrast, the sub-GHz band is less congested resulting in more reliable links. The quieter spectrum means fewer communication retries, making sub-GHz more energy efficient for battery-powered systems. Secondly, sub-GHz band frequencies also have higher wavelengths than its counterpart, offering better range and more reliable operation in environments where direct line of sight is difficult to achieve, including agricultural environments and structurally dense environments such as cities. As studied in [13], [14], the sub-GHz band demonstrates longer reachability requiring less number of hops to reach the destination. As a result, requiring fewer base stations to cover the same area and number of sensing devices, driving the overall solution cost down. Today many standardized as well as proprietary wireless technologies like IEEE 802.15.4, IEEE 802.11ah, Z-Wave, LoRa, and Sigfox already leverage the sub-GHz band as a primary form of communication [15].

2) REGULATORY COMPLIANCE

The use of the radio spectrum is regulated in most countries around the world and plays a critical role when choosing wireless connectivity solutions for the IoT. Here, we will briefly look at the regulatory compliance imposed by the Federal Communications Commission (FCC) in the United States and the European Telecommunications Standards Institute (ETSI) in Europe for the ISM bands.

In the US, the frequency band of 902–928 MHz is one of the license-free ISM bands and has no restrictions in terms of the duty cycle and allows a much higher output power limit, making it popular for short-range applications. On the other hand, the 433 MHz band is not a general-purpose band in North America and is regulated by FCC Part 15 [16]. Mainly there are two main restrictions for devices utilizing 433 MHz. First, Part 15.231.a.3 restricts scheduled periodic data transmissions, however, it allows polling data that cannot exceed 2 s of transmission time per hour. Second, under Part 15.231.e the device can only transmit for a maximum of 300 ms every 10 s, i.e., a duty cycle of 3%. This restriction of one transmission every 10 s is currently the major hurdle in the wide adoption of 433 MHz for IoT applications in the US.

In Europe, duty cycles are regulated by section 7.2.3 of the ETSI EN300.220 standard where 863 to 870 MHz bands have been assigned as license-free with a duty cycle restriction of 0.1% (868.7 – 869.2 MHz), 1% (863.0 – 868.6, 869.7–870.0 MHz), or 10% (869.4 – 869.65 MHz). For the 433 MHz band which is specifically allocated for low-power medical implants, there are no duty cycle restrictions.

D. RELATED WORK

Using TSCH in sub-GHz frequency bands is first described by Brachmann *et al.* [13]. The authors are interested in enabling multi-band devices to support both 2.4 GHz operation and longer sub-GHz timeslots in a single TSCH schedule. However, their work does not focus on low-data-rate applications and ignores routing and converge-cast scheduling issues.

Haubro *et al.* [17] investigate the operation of TSCH atop LoRa PHY layer. Their work opens up a whole new

sub-field by merging these two promising network protocols. However, their work is just a first step in this direction and falls short of practical application of TSCH on top of a low-rate communication layer, as demonstrated in the present paper.

TSCH has been used for agriculture applications previously, in particular by utilizing the SmartMesh, a commercial closed source implementation [3], [4]. The authors report 100% PDR and overall excellent performance. However, they use the 2.4 GHz frequency band, which is poorly suited for achieving long-distance links in non-line-of-sight conditions. Due to the closed-source nature of the network stack they used, most details about the scheduling and routing functionality of the network remain a black box to other researchers.

Support for multiple roots in RPL networks via a single “virtual” root was first investigated by Carels *et al.* [18]. We take this idea and apply it to TSCH networks, where additional difficulties are created by the fact that all the nodes need to synchronize their time to a single global clock.

Our paper is the first to introduce special rules for root in TSCH networks that use the Orchestra scheduler. However, the idea that the root and nodes close to the root should be treated in a special way is certainly not new. The implications of tweaking MAC protocol operations in this neighborhood have been explored before, for example, by Ahn *et al.* [19], who suggest extending the CSMA-based network with TDMA operation in the area close to the root.

III. APPROACH

A. TARGET APPLICATION

Many IoT monitoring applications report data that change slowly in time, such as temperature and humidity. This work targets such an application with low data rate requirements where sensor nodes are primarily battery-powered and demand ultra-low-power operation. Specifically, we consider the micro-climate monitoring of grain. The moisture content of grain directly translates to the price the farmer will be paid for it. Our goal is to capture and transmit key quality conditions of grain throughout the storage season which can last from 3 up to 12 months.

However, utilizing wireless sensors for in-grain monitoring presents its own challenges and must adhere to the following requirements:

- The size of these sensors needs to be small in order to flow with the grain and go wherever the grain goes, restricting the size of the battery that can be utilized to power these sensors and the radio transceiver.
- Operating in an environment where sensors are submerged in-grain causes extensive multi-path fading and shadowing of wireless links, thus impeding the overall wireless connectivity and network reliability.
- Device lifetime needs to be up to 12 months.
- The minimum data delivery rate is 90%.

We do not have specific requirements for latency; it can be in the minute range, as temperature and humidity inside the grain are slow-changing environmental variables.

As such, it is critical to design a dependable wireless system that can guarantee the delivery of grain sensor data to farmers with high quality of service (availability) and reliability. Due to high RF signal attenuation caused by the grain as a signal propagation medium, frames transmitted from the wireless sensors can only be received within a maximum distance of a few meters, at best up to 2 m at the 915 MHz frequency band [20]. To provide greater coverage, for instance, to be able to report moisture levels at different locations within a grain storage silo or transport carriage, data need to be relayed between sensor nodes traversing a multi-hop path before reaching the gateway node. In addition, wireless sensor lifetime of up to 12 months when powered by a single battery is required to provide continuous sensing throughout the storage season.

As a consequence, proprietary long-range wireless technologies such as SigfoxTM or LoRaTM cannot be employed due to their high peak current demand. For instance, SX1276, a widely used LoRa transceiver from Semtech Corporation [21] consumes 120 mA peak current while transmitting w.r.t. CC1310 [22], a Zigbee transceiver utilized in this work that consumes a maximum of 25 mA. Additionally, these proprietary long-range wireless technologies lack support for multiple topology options such as multihop data collection. Although several proposals have been made by various researchers [23], [24], none are standardized. In contrast, IEEE 802.15.4 provides both low-power communication and the ability to configure different network topologies such as star, tree, and mesh networks.

Nevertheless, the approach proposed in this article can also benefit other application scenarios such as machine monitoring in an industrial setting where there are high signal multi-path and fading effects and other sensing applications with deterministic traffic requirements.

B. TSCH SCHEDULE

We build the schedule for the application on top of the receiver-based Orchestra mode. For low data rate applications, the receiver-based variant is expected to be more energy-efficient, as it allocates fewer reception slots per node. In doing so, nodes have to spend less time in idle listening. Existing research finds the sender-based variant more reliable at the same energy consumption level [9]; however, the traffic rate evaluated in that study was higher than the rate required by our application. Furthermore, the sub-GHz frequency band uses a longer PHY layer header compared to the 2.4 GHz band. As a result, the receiver must be on for a longer time, leading to increased idle listening cost and therefore making the receiver-based mode even more energy efficient in comparison.

We apply the multi-channel Orchestra approach described in [9], as our initial results showed that using multiple chan-

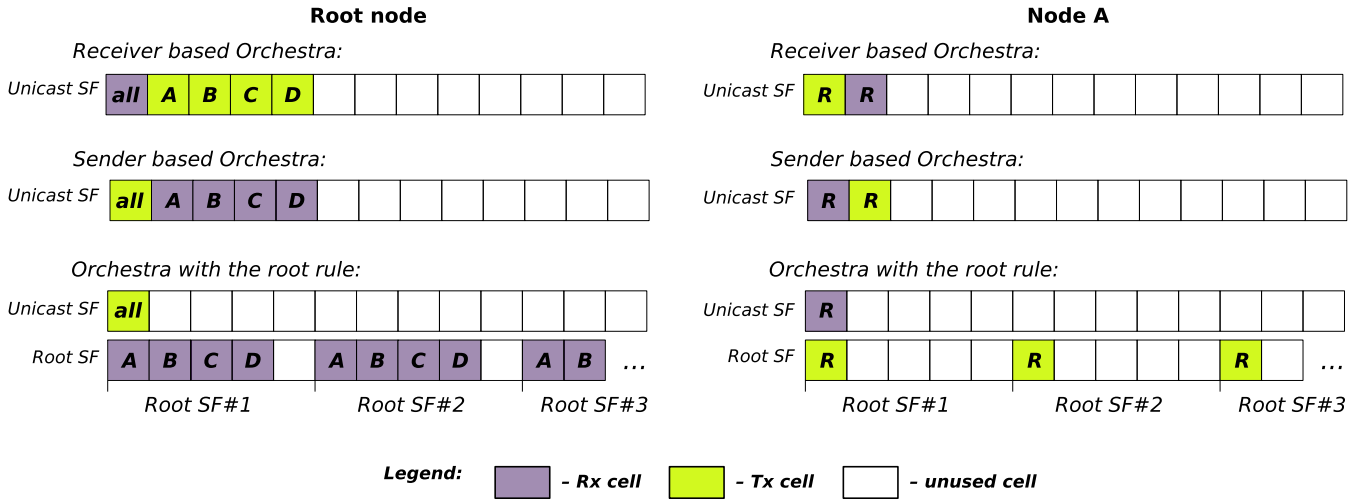


FIGURE 2. Schedules in the example network in Fig. 3. Left: for the root node; right: for the node A.

nel offsets for unicast cells greatly reduces intra-network interference.

Nevertheless, the network performance initially failed to meet the combined reliability and energy efficiency requirements (Section III-A). Part of the problem is that the RPL routing protocol is characterized by a fast packet exchange while building the network. This creates problems for schedules that do not have a sufficient number of active cells; on the other hand, increasing the number of active cells leads to poor energy efficiency. It is possible to tune the RPL routing protocol to make it send fewer packets. However, this leads to increased time required to construct the network. The time for all nodes to join the network is critically important to increase energy efficiency, as before joining the network a node is in a continuous channel scan mode, under which the radio alone consumes 5.4 mA current [22]. The time required for nodes to rejoin the network after an accidental reboot (e.g., because of a hardware issue or a watchdog timer reset) must also be minimized. While joining the RPL network is not strictly necessary to be energy efficient – a node can be part of the TSCH network without being in the RPL DODAG – joining RPL is required to send TSCH Enhanced Beacon messages and to allow other nodes to join the TSCH network through the local node.

In summary, there is a trilemma:

- A small number of active cells leads to a large number of packet collisions unless the RPL traffic is slowed down;
- A large number of active cells leads to energy inefficient operation due to idle listening;
- Slow RPL traffic leads to a long time required to build the network.

C. TSCH ROOT RULE

The requirement to handle heavy RPL traffic is especially severe for nodes that have many direct neighbors. Sub-GHz networks are expected to have long-range links, hence, the number of hops is expected to be low. As a result, the root

node is expected to have a large number of neighbors. We experimentally verified that this is the case and that it makes it difficult to build an RPL DODAG in the network when an energy-efficient schedule is used.

To circumvent this limitation, we introduce a special Orchestra scheduler rule for the root node. We observe that the root node is usually not as energy-constrained as the other nodes; for instance, in our application, it is expected to have a solar cell and a large battery. As a result, the root node can spend most of the time in idle listening.

In light of this, we propose the *root rule* for the Orchestra scheduler. This rule adds a dedicated slotframe for communication between the root node and the nodes directly adjacent to the root node.

If a schedule with the root slotframe (Fig. 2) is used:

- The root node uses all cells in the root slotframe for reception, as well as a normal-sized unicast slotframe with one active slot for transmission;
- Direct neighbors to the root node add the root slotframe with one cell active for transmission, as well as the normal unicast slotframe for communication with other nodes, including one active slot for reception from the root;
- The rest of the nodes operate as before, e.g., using Orchestra’s receiver based or sender based modes.

A network node is expected to install the root slotframe if it has any root nodes as direct neighbors. For instance, nodes A – D in Fig. 3 add the root slotframe to transmit packets to the root node R. For reception from the root node, and for communication with nodes E and F they use the normal Orchestra unicast slotframe. With nodes E and F, two situations are possible. The first is that they are out of the direct communication range of the root, and do not receive any direct packets from it. In this case, these nodes do not add the root slotframe at all. The second situation is that they do receive some packets directly from the root, but do not select the root node as their RPL parent node due to bad link quality.

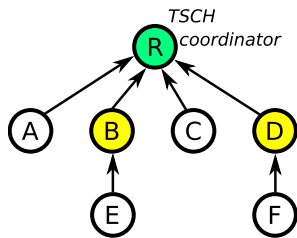


FIGURE 3. Example network.

In this case, these nodes do not add the root slotframe but do not normally use it for data transmission, as their packets to the root are routed through an intermediate node. Finally, if the nodes do not receive any more direct packets from the root for a configuration-dependent timeout, the root slotframe is removed again.

The root slotframe should be shorter than the Orchestra's unicast slotframe to allow a higher traffic rate.

Next, we explain how the root slotframe works in two potentially problematic situations. The first situation is when a network node attempts to communicate with a root node before it is aware that it is a root node. In this situation, the packets are still going to be received correctly due to the nature of the root slotframe (all its cells are for reception). The communication will simply not be as efficient, because the normal unicast slotframe will be used instead, with fewer transmission slots available.

The second situation is when cells from multiple nodes collide in the root slotframe. A node gets its cell offset in the root slotframe by calculating the Orchestra hash function on its MAC address and taking the result modulo of the root slotframe size. The default Orchestra hash function may be used for all slotframes, including the root one. An unlucky choice of the hash function together with the size of the root slotframe may create cell collisions. In the worst case, all nodes will attempt to use the same cell. To mitigate the number of resulting packet collisions, the IEEE-802.15.4 back-off mechanism is helpful and is enabled by default in TSCH. It increases reliability at the expense of latency, and latency is not critical for our application. Furthermore, even in this worst-case the performance will still be better than in the worst-case of the unicast slotframe (which is vulnerable to the same problem), as the root slotframe is shorter, so the network nodes have more opportunities for transmission.

Conflicts between cells in multiple slotframes are resolved as described in the IEEE 802.15.4-2015 standard [1]: first, transmission takes priority over reception, then a slotframe with a lower handle takes priority. For instance, in the first slot in Fig. 2, node A would check its packet queues, and send a packet to the root if its queue is not empty; otherwise, it would use the slot to listen for messages from other nodes.

D. MULTIPLE RPL ROOTS

The low-power networks in some of the target applications may have multiple connection points to the cellular

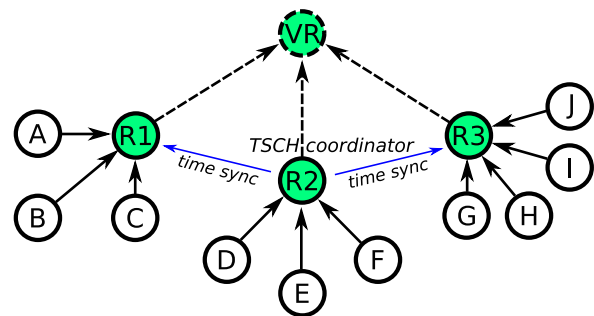


FIGURE 4. Example network with multiple root nodes R1, R2, and R3, where the node R2 is also the TSCH coordinator. VR is the virtual root node.

network or the Internet. For example, each grain silo and grain transport carriage may have its own gateway node. One solution would be to create multiple TSCH & RPL networks, rooted in each of these gateway devices. However, this is going to lead to sub-optimal routing in case devices close to one gateway accidentally join a network rooted at a remote gateway.

To support multiple RPL root devices in a single network, we apply the virtual root approach from Carels *et al.* [18] (Fig. 4). The root devices advertise a connection to a virtual device VR with a fixed, known address. The link cost between the root devices and the virtual root VR is treated as zero. The network nodes generate data packets addressed to VR. The network stack on a root node, when receiving a packet addressed to VR, does not attempt to forward it, but instead passes it to the interface connecting the device with a gateway device to an external network. This configuration allows all network nodes to route their data packets to their closest root device (Fig. 4).

The root nodes are all equal from the RPL point of view, but they cannot be equal from the TSCH point of view. Multi-root TSCH networks are not possible in the general case unless external time synchronization is used between the roots. The presence of non-root nodes in the routing paths between root nodes would lead to conflicts between the desired direction of TSCH time synchronization and the desired upstream direction of routing. However, in a typical case for our application, the root devices, when considered on their own are expected to form a graph that is not partitioned. This is feasible as a root device typically has an external antenna and uses a higher transmit power due to access to external power or a bigger battery. It may even have line-of-sight links to other root devices. This assumption is true for grain sensing in particular, as root nodes are placed out of the grain, and have hundreds of meters of communication range, while other nodes are placed in the grain, and have only meters of communication range (Section III-A). Based on this assumption, we require that a root node joins the RPL network only through another root node as their parent. To handle this requirement, the root nodes are configured to ignore other nodes in the joining process and to never switch to an RPL parent that is not a root node.

As a result, we do not assume access to an external time synchronization mechanism but instead, rely on the TSCH protocol itself to provide time synchronization between the roots. This setup requires that (1) each root node is directly connected at least to one other root node and that (2) the connectivity graph between the root nodes is not partitioned. Otherwise, multiple TSCH networks will be created, leading to potentially lower efficiency and reliability.

The TSCH coordinator is dynamically selected among the root nodes by running this algorithm on each root node:

- (1) A root node starts up.
- (2) It scans channels for *timeout* and a random jitter factor minutes, waiting for a TSCH network to join, where *timeout* is a configuration constant.
- (3a) If a TSCH network is found and joined, the node becomes a regular RPL root node.
- (3b) If a TSCH network is not found, it starts its own TSCH network by becoming a coordinator as well as an RPL root node.

For example, if the node *R2* in Fig. 4 starts the TSCH network first, the nodes *R1* and *R3* join the network created by it. If the node *R1* starts first, the node *R2* joins directly, and then the node *R3* joins through *R2*. This algorithm does not guarantee the formation of a single network: for example, all nodes may decide to start their TSCH networks at the same time. However, the appropriate selection of a jitter parameter makes this situation unlikely. The jitter should be sufficiently large relative to the TSCH Enhanced Beacon (EB) packet and RPL DIO packet intervals, e.g., larger than the sum of EB period and DIO packet periods.

If there are multiple root nodes, they all are expected to use the same format for the root slotframe. This is enforced in the default configuration, as the root slotframe size is a constant that is shared by all nodes, and the Orchestra default hash function only takes the local node's MAC address as a parameter, not the remote node's MAC; hence, a node's slot number in the root slotframe is not affected by the MAC address of the root. As a result, a network node can seamlessly switch to another root node without modifying its TSCH schedule. The switching between root nodes, or sub-networks sourced by different root nodes, is done by standard RPL mechanisms. The RPL objective function does not have to be modified, as from a network node's perspective, there is only a single root node (i.e., the virtual root), and the actual root nodes are merely its direct neighbors. Consequently, the standard RPL parent selection mechanisms remain suitable for the task.

IV. EVALUATION

A. TEST SETUP

We base our evaluation on a testbed consisting of 29 nodes deployed in an office environment. The nodes are controlled by a Texas Instruments CC1310 System-on-Chip (SoC), comprising a Cortex-M3 MCU and a sub-GHz IEEE 802.15.4 radio transceiver. The physical layer (PHY) has been

TABLE 1. Experimental settings and parameters.

Parameter	Value
Number of nodes	29
Routing	RPL
MAC	TSCH
Experiment duration	30 mins settling, 120 mins of actual
Traffic pattern	1 pkt / 5 mins
Application layer payload size	40 bytes
TSCH	Value
EB frequency	32 s
Timeslot length	40 ms
Max packet retransmissions	5
Channels used	15, 25
Orchestra	Value
Mode of operation	Receiver based
EB slotframe	397 slots
Unicast slotframe	49, 101, or 199 slots
Broadcast slotframe	43 slots
Root slotframe	31 slots
RPL	Value
Objective function	MRHOF
Mode of operation	No downward routes
Minimal DIO interval	64 sec
Maximal DIO interval	512 sec
Probing interval	120 sec
Hardware	Value
Data rate	50 kbps
TX power	0 dBm
Frequency	433 MHz

configured to use a 50 kbps data rate with an external antenna tuned to operate at the 433 MHz band.

In each of our experimental runs, every source node generates 40 byte data packets randomly. The average interval between packets is 5 minutes. The packets are forwarded it to the root node. We use a node with ID $0 \times 5d32$ as the network root node in the office testbed. Each experiment is run for 150 minutes with the first 30 minutes serving as a burn-in time for the RPL routing protocol to establish routes. The subsequent 120 minutes are analyzed and results reported for (i) power consumption, (ii) packet delivery ratio (PDR), and (iii) network stability. We evaluate power consumption of the network using the amount of time a node keeps its radio on in different states, i.e., the average radio duty cycle (RDC). Experiments in the testbed are executed at random times of the day, but back-to-back to ensure fairness. The details of the different parameter settings used in our evaluation are summarized in Table 1.

B. TESTBED RESULTS

Fig. 5 shows the main results from the testbed experiments: radio duty cycle (Fig. 5b), packet delivery ratio (Fig. 5a), and parent stability (Fig. 5c). Only the configuration with the root rule and slotframe size of 49 achieves good results in all metrics. In particular, it meets the application reliability requirements defined in Section III-A, as it shows 99.0 % end-to-end reliability and low energy consumption with only

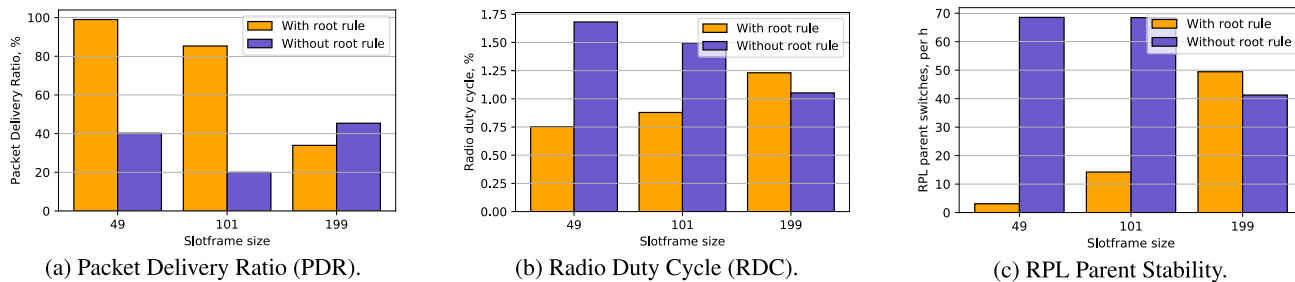


FIGURE 5. Comparison of performance metrics for different network settings. Testbed results.

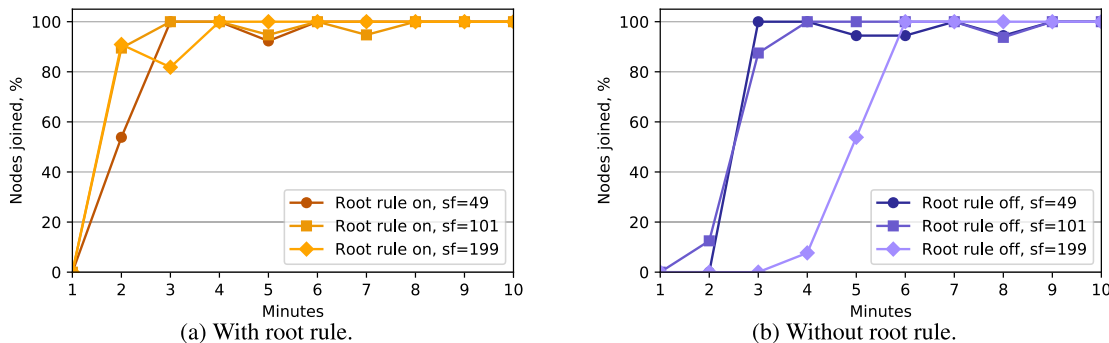


FIGURE 6. Nodes joined over time. Testbed results.

1.6 active slots for reception per second (as calculated from Table 1).

When looking at the experiments with the root rule, it can be seen that a longer slotframe translates into worse results: fewer active slots lead to a higher number of collisions. This in turn reduces PDR and RPL parent stability, and causes more packet re-transmissions, therefore failing to provide a reduced duty cycle. In contrast, without the root rule, even the shortest slotframe fails to show acceptable results. Due to a large number of packet collisions close to the root, the RPL routing tree is very unstable (Fig. 5c). Under longer slotframe sizes, it even fails to form completely.

These results are supported by the other figures. Fig. 6 shows nodes joining over time, measured from the moment the root node becomes a TSCH coordinator. Fig. 6b shows longer joining times compare to Fig. 6a, especially when the slotframe size is 199, as well as increased instability: Some nodes keep leaving the network after having joined it.

Fig. 7 shows the RPL routing tree topologies formed in the different experiments. Only the root rule in combination with short slotframe size is able to take advantage of the flat physical network topology (Fig. 7a). When the slotframe size increases, the number of packet collisions at the root node also increases, therefore some nodes move a hop further away from the root in the RPL routing tree (Fig. 7b). The number of hops in the network increases even more if the root rule is disabled (Fig. 7c). Without the root rule and with a longer slotframe some nodes fail to join the RPL routing tree altogether. This is shown in Fig. 7d as fewer joined nodes. (We clarify that Fig. 6 shows the percentage of nodes

joined relative to the maximum number of nodes joined in the experiment.)

In sum, the root rule has a significantly positive impact on the performance of our testbed network, due to the root node having many direct neighbors resulting in a shallow network.

C. SIMULATION RESULTS

1) EFFECT FROM THE NUMBER OF ROOT NODES

To evaluate the effect from multiple root nodes, we additionally perform simulation experiments using COOJA, the network simulator distributed as part of the Contiki OS. COOJA allows us to have full control over network conditions and conduct simulations in a repeatable manner. The network is randomly generated and consists of 28 regular nodes and between one and three root nodes. Other settings are as in Table 1; the root rule is enabled. In the multiple root node case, each root node is directly connected to at least one other root node, as required for the root rule.

Simulation results are shown in Fig. 8. Adding the extra root nodes increases the PDR (Fig. 8a), and the positive effect is proportional to the number of roots. The extra roots also reduce the RPL churn rate (Fig. 8c); however, it is already small with two roots and the addition of a third root does not result in any visible subsequent improvement. The effect on the RDC appears to be random and insignificant (Fig. 8a).

Adding more roots would be possible, but would lead to diminishing gains and would not be cost-effective for our target applications, partially due to the additional logistic challenges created by the need to set up the extra gateway devices and solar panels for power.

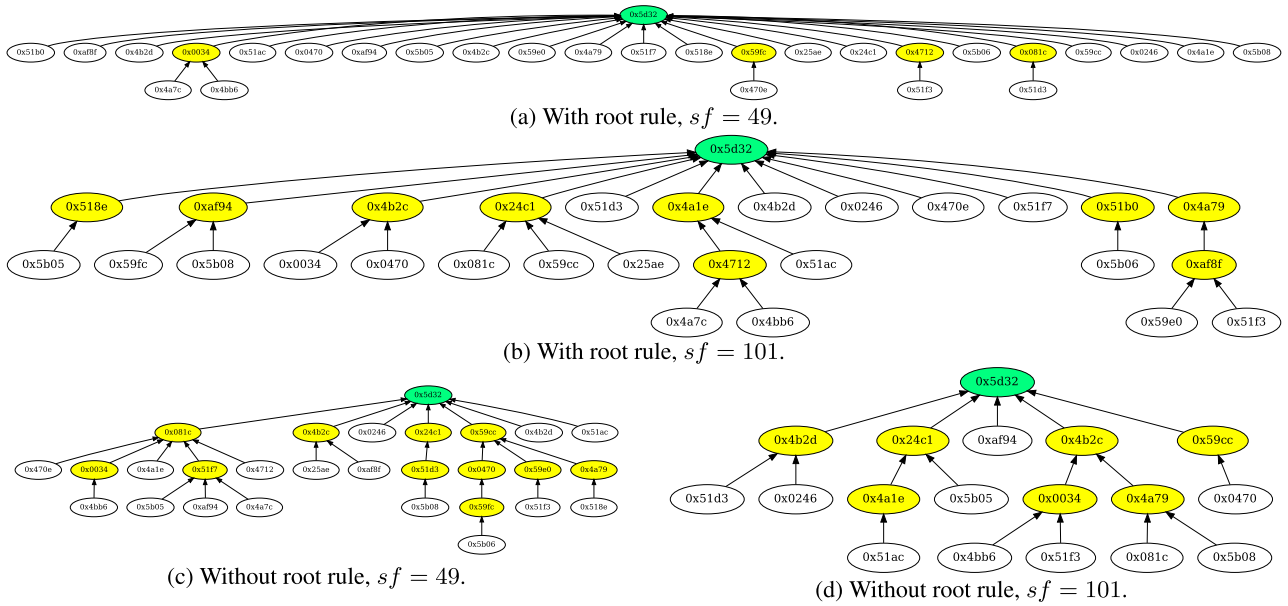


FIGURE 7. RPL network topologies formed under different network settings. Testbed results.

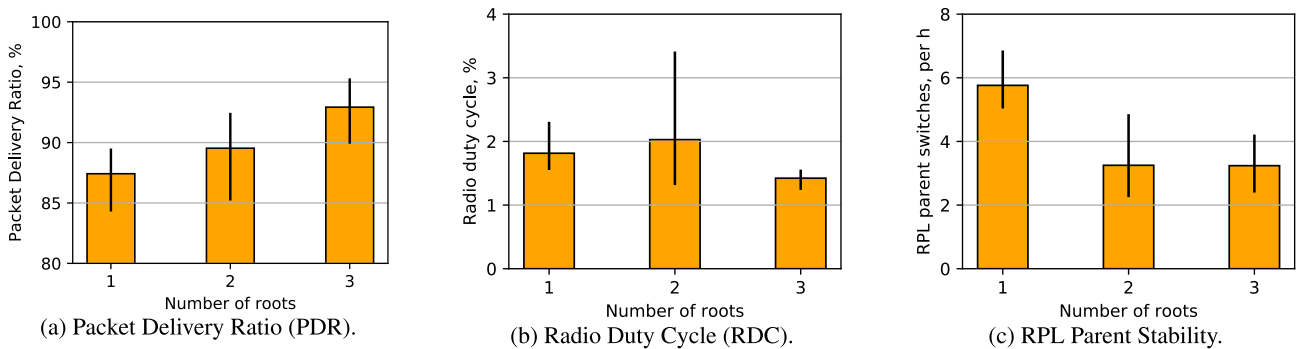


FIGURE 8. Comparison of performance metrics depending on the number of root nodes. Simulation results.

2) FAULT TOLERANCE OF NETWORKS WITH MULTIPLE ROOT NODES

In another set of experiments, we measure the fault tolerance of the network by disabling some root nodes in an operational network and observing the impact on network performance metrics. We compare the standard multi-network approach with our virtual root approach. In the multi-network approach, the network has three root nodes, each of which starts its own TSCH network as a TSCH coordinator. In the virtual root approach, there are three root nodes as well, but they all are in a single TSCH network; only one of them is a TSCH coordinator (Section III-D).

Each experiment is conducted in the following way: the simulation is configured to run for an hour. During the first 20 minutes, all root nodes are operational. At the 20 min mark, one root node is disabled, and at the 40 min mark another one is disabled, leaving only one active root node. Other settings are as in Table 1.

With the multi-network approach, network nodes that are associated with a disabled root node have to leave their previous TSCH network first and join a new one; then, join a completely new RPL network, learn about the set of suitable RPL parent nodes, and select one parent from them. If a non-coordinator root node is disabled in a network using the virtual-root approach, a different situation arises: all other nodes remain in the same TSCH and RPL network. The nodes that were routing their packets to the disabled root node simply have to pick a new RPL parent from their already discovered sets of candidates.

The results are shown in Fig. 9. In both sets of experiments, the network displays relatively high fault tolerance due to nodes being able to switch to another root. However, the virtual root approach leads to a higher PDR (13 percentage points higher on the average) and increased routing stability (the number of RPL parent switches is 2.5 times lower on the average), when compared with the standard multi-network

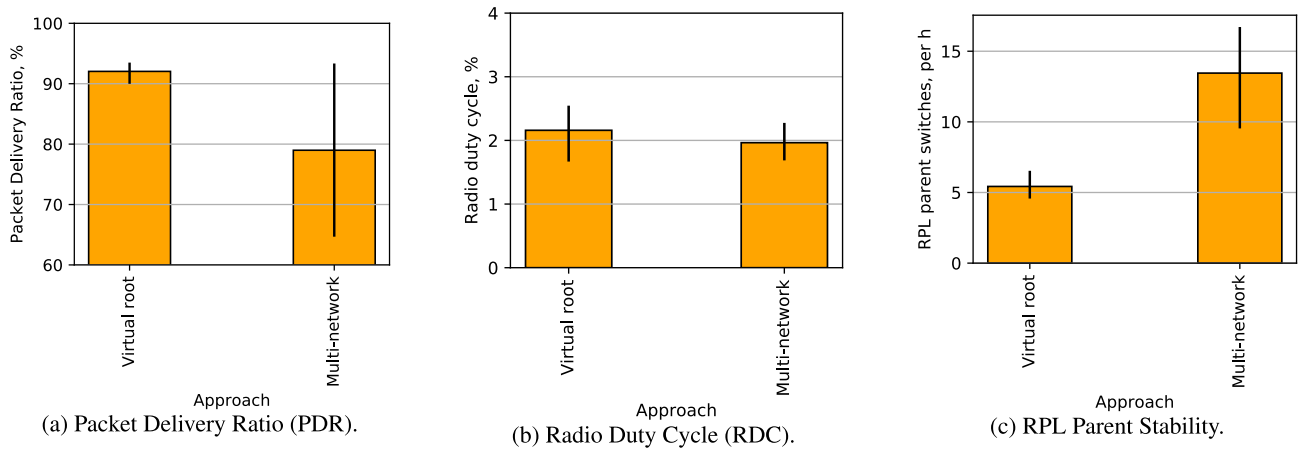


FIGURE 9. Performance comparison in networks with three root nodes, two of which are disabled over time. “Virtual root” – our virtual RPL root approach in a single TSCH network; “multi-network” – standard TSCH that creates multiple TSCH & RPL networks. Simulation results.



FIGURE 10. Wireless sensor inside a custom made enclosure.

approach. This demonstrates that the fault tolerance of the network is increased by applying the virtual root approach.

D. DEPLOYMENT RESULTS

To verify that the results obtained in the testbed is achievable in a real-world environments, we now move from an in-office testbed to a real-world deployment.

Location: We deploy 28 wireless sensor nodes (Fig. 10) in an outdoor grain pile as illustrated in Fig. 11. The bottom sensors are plunged inside the grain a meter apart from each other vertically and 2 m apart horizontally, with the topmost sensors placed on the surface. The root node with an external antenna is installed close to the middle of the grain pile fixed to a 3 m long galvanized steel pole. The experimental campaign lasts for 21 days, after which the sensors are retrieved from the pile.

Hardware and Software: Each sensor node is powered by a single 3 V AA Lithium battery running a custom application that measures and reports temperature and humidity of the grain environment to the sink node. The sampling period of the application is configurable and is set to 30 minutes. On the sink node, the radio is always kept on as it is powered by a larger battery pack. Furthermore, all nodes transmit at a 10 dBm power and use IEEE 802.15.4 channels 15 and 26 as TSCH hopping channels. The Orchestra unicast slotframe size is fixed at $sf = 49$. All remaining network configuration settings are kept the same as reported in Table 1.

Results: At the end of the experimental campaign these main results are analyzed: packet delivery ratio and the network topology, as captured at the back-end. The analysis reveals that all sensors reported grain climate data at a packet delivery ratio of above 95 %, exceeding the application reliability requirements defined in Section III-A.

Next, Fig. 12 shows the RPL routing tree topology formed in the outdoor experiment. The network has the root rule enabled and is able to take advantage of the flat physical network topology, similar to the one obtained in the indoor testbed (Fig. 7a). Despite being buried in the grain, most of the sensors are able to directly reach the root node within a single-hop, while a few sensors that are further from the root and at the bottom of the chain move a hop away from the root taking advantage of a intermediate nodes to relay their data. These results are attributed to the facts that:

- The network utilizes a short slotframe size of $sf = 49$ and the root rule is enabled. This combination reduces the number of packet collisions at the root node.
- The PHY layer with low-data rate operating in 433 MHz band is able to extend the communication range, resulting in most nodes being within a single hop of the root.

V. DISCUSSION

A. WHAT WENT WELL

We were able to successfully port the TSCH protocol to the CC1310 platform, which was not available in Contiki-NG before our work. We have also enabled the 433 MHz physical layer support for the experiments. As a result, now Contiki-NG TSCH supports 433 MHz, 868/915 MHz and 2.4 GHz bands for short and long-range applications.

Even though the TSCH standard [1] was mainly designed for the 2.4GHz frequency band with 250 kbps data rates, it was easy to adapt it to the sub-GHz band with five times slower bit-rates. The main change required was a custom TSCH timing template, which replaces the standard 10 ms slot and increased the time for most actions in the slot such as the transmission of a packet and an ACK, as well

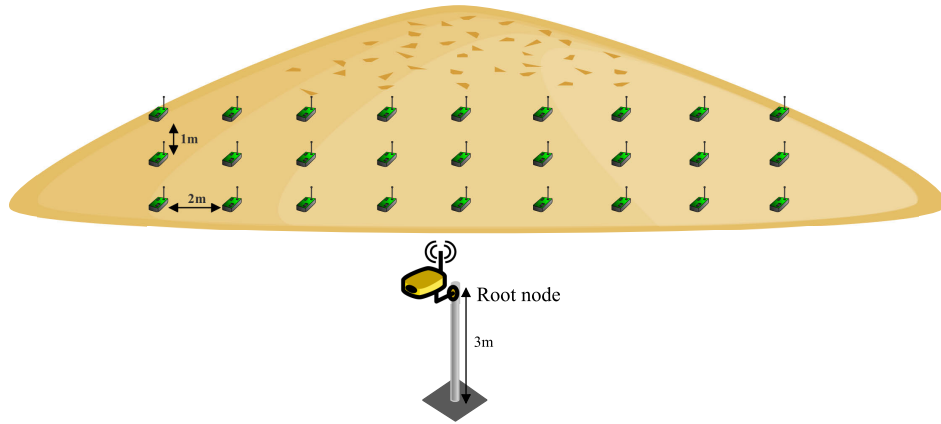


FIGURE 11. Illustration of our 28 node sensor deployment for condition monitoring in a grain pile.

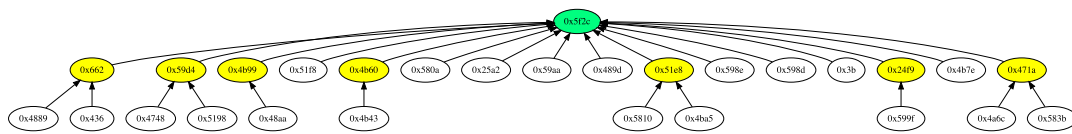


FIGURE 12. RPL network of a 28 node micro-climate sensing deployment with root rule enabled, $sf = 49$, one root node.

as the pre-amble reception time. The guard time in TSCH timeslot may be left unchanged – time synchronization is not inherently less accurate in the sub-GHz bands –, but the time reserved to receive the preamble and the synchronization header must be longer. First, the preamble and the synchronization are longer: together they take 6 bytes with our settings, compared with 4 bytes in the 2.4 GHz band. Second, the reception of each byte takes five times as long.

B. DRAWBACKS AND PROBLEMS

In low-rate applications, a large proportion of the traffic sent over the air is network maintenance traffic. As the data packet frequency is low, the overhead of both TSCH and especially RPL becomes significant. The TSCH time synchronization frequency can be reduced by applying techniques such as adaptive synchronization [25], [26]. However, there is no easy way to reduce the number of packets required to construct and maintain the forwarding DAG. As a result, the RPL routing protocol was a constant source of frustration during testbed experimentation as well as during the real deployment. Two issues should be highlighted in particular. The first one has to do with the difficulty of modeling RPL control packet frequency due to the Trickle timers. The second is the fact that the parent selection done by RPL is highly non-deterministic. This is due to the randomness in probing and data packet generation, and the hysteresis that causes nodes to persist with using as their preferred parent the first candidate parent they happened to hear from. While these problems are not unique to low data rate networks, the sparse schedule and the high proportion of RPL packets relative to data packets exacerbates these problems in low data rate settings. These

problems lead to difficult-to-debug problems in routing tree formation, difficulties in selecting a good TSCH schedule that is both energy efficient and able to handle Trickle activity bursts, and difficulties in predicting the performance and battery lifetime of real-world deployments.

C. FUTURE WORK

1) REPLACING RPL

It is tempting to replace RPL with upward-directed flooding for our application, given that the amount of nodes and links in our target networks is envisioned to be relatively small, and the data rate to be low. Existing research shows that even simple flooding beats RPL in some real-world applications [27].

However, a mechanism is required in order to set up the TSCH time synchronization tree, preferably one that is guaranteed to avoid long-lasting loops. RPL fits this role, and there are no obvious alternatives. The Contiki-NG TSCH implementation does support time-source auto-selection using TSCH EB packets on their own, however: (1) no experimental evaluation data is available on the performance of the mechanism; (2) the approach would not work with the Orchestra scheduler that supports EB reception only from the time source.

2) EXTERNAL TIME SYNCHRONIZATION

If one can replace the TSCH time synchronization tree with something else, one can then also substitute RPL for an alternative. This naturally suggests a second idea for an improvement: Adopting an external synchronization mechanism. In this case, the strengths of TSCH would be preserved, namely, its pseudo-random channel hopping along with scheduled

communications, while the cumbersome time source selection process would be eliminated.

While using a GNSS (Global Navigation Satellite System) for this purpose is not feasible, as it would consume too much energy on the network nodes, even if the signal was able to penetrate the grain, several other candidate approaches exist:

- A *continuous* transmission of a time signal from the gateway device on a dedicated channel.
Benefits: simplicity of this approach; no additional radio hardware may be needed.
Drawbacks: depending on the frequency and the signal strength, this approach may prove unsuitable to use in unlicensed ISM bands.
- A *periodic* transmission of a time signal from the gateway device, which is picked up by a wake-up radio [28] on the network nodes.
Benefits: the approach allows the gateway's signal to have low duty cycle.
Drawbacks: another radio chip is required on the device, and it must interface with the rest of the system.
- Relying on a source external to the network, for example, a regional low-frequency transmitter such as DCF77 [29] in Europe or WWVB [30] in the USA.
Benefits: no additional hardware and maintenance required on the gateway side.
Drawbacks: on the nodes, a separate hardware solution is required for each region.

Each approach must be experimentally verified to determine its suitability, in particular, the level of shadowing by the grain may be a critical problem for all single-hop approaches.

VI. CONCLUSION

This paper describes the design and experimental evaluation of the TSCH protocol on a Sub-GHz frequency band using Texas Instruments CC1310 System-on-Chip based sensor nodes. Evaluation shows that the network performance is increased by adapting a custom TSCH schedule where the root nodes have an increased number of slots for reception. This is crucial for reliable operation in some scenarios; in our experiments, it increases the network packet delivery ratio more than twice, up to 99.0% PDR. We also investigate a "virtual root" approach, previously seen in literature about the RPL protocol, and show that it is possible to apply it to TSCH networks. Having more than once root node increases the network packet delivery ratio and also makes RPL parent selection more stable. In our experiments, the PDR increases by around 3% for each additional root node. Sensor networks based on this work are currently being deployed and tested for smart agriculture applications.

REFERENCES

- [1] *IEEE Standard for Local and metropolitan area networks—Part 15.4*, Standard 802.15.4-2015, 2015.
- [2] S. Duquenooy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.*, 2015, pp. 337–350.
- [3] T. Watteyne, A. L. Diedrichs, K. Brun-Laguna, and J. Chaar, "Peach: Predicting frost events in peach orchards using iot technology," *EAI Endorsed Trans. Internet Things*, vol. 2, no. 5, pp. 1–12, 2016.
- [4] K. Brun-Laguna, A. L. Diedrichs, D. Dujovne, C. Taffernaberry, R. Léone, X. Vilajosana, and T. Watteyne, "Using SmartMesh IP in smart agriculture and smart building applications," *Comput. Commun.*, vol. 121, pp. 83–90, May 2018.
- [5] 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, no. 2, pp. 1–27, Apr. 2020.
- [6] D. Chen, M. Nixon, and A. Mok, *WirelessHART: Real-Time Mesh Network for Industrial Automation*. New York, NY, USA: Springer, 2010.
- [7] Q. Wang, X. Vilajosana, and T. Watteyne, *6TiSCH Operation Sublayer (6top) Protocol (6P)*, document RFC 8480, Nov. 2018.
- [8] S. Kim, H.-S. Kim, and C. Kim, "ALICE: Autonomous link-based cell scheduling for TSCH," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 121–132.
- [9] A. Elsts, S. Kim, H.-S. Kim, and C. Kim, "An empirical survey of autonomous scheduling methods for TSCH," *IEEE Access*, vol. 8, pp. 67147–67165, 2020.
- [10] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *IPv6 Over Low Power Wireless Personal Area Networks (6LoWPAN)*, document RFC 4944, IETF, 2007.
- [11] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document RFC 6719, Sep. 2012.
- [12] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, document RFC 6206, Mar. 2011.
- [13] M. Brachmann, S. Duquenooy, N. Tsiftes, and T. Voigt, "IEEE 802.15.4 TSCH in sub-GHz: Design considerations and multi-band support," in *Proc. IEEE 44th Conf. Local Comput. Netw. (LCN)*, Oct. 2019, pp. 42–50.
- [14] R. Sandoval, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, and J. Garcia-Haro, "Evaluating the more suitable ISM frequency band for IoT-based smart grids: A quantitative study of 915 MHz vs. 2400 MHz," *Sensors*, vol. 17, no. 12, p. 76, Dec. 2016.
- [15] E. De Poorter, J. Hoebeke, M. Strobbe, I. Moerman, S. Latré, M. Weyn, B. Lannoo, and J. Famaey, "Sub-GHz LPWAN network coexistence, management and virtualization: An overview and open research challenges," *Wireless Pers. Commun.*, vol. 95, no. 1, pp. 187–213, Jul. 2017.
- [16] *Part 15-Radio Frequency Devices; 15.231 Periodic Operation in the Band 40.66-40.70 MHz and Above 70 MHz*, FCC, Washington, DC, USA, 2011.
- [17] M. Haubro, C. Orfanidis, G. Oikonomou, and X. Fafoutis, "TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things," *Internet Technol. Lett.*, vol. 3, p. e165, Aug. 2020.
- [18] D. Carels, N. Derdaele, E. D. Poorter, W. Vandenberghe, I. Moerman, and P. Demeester, "Support of multiple sinks via a virtual root for the RPL routing protocol," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, pp. 1–23, Dec. 2014.
- [19] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 293–306.
- [20] P. Armstrong, "Wireless data transmission of networked sensors in grain storage," in *Proc. ASAE Annu. Int. Meeting*, 2003, Art. no. 036157.
- [21] *SX1276/77/78 LoRa Modem*, Texas Instruments, Dallas, TX, USA, Jul. 2017.
- [22] *CC1310 SimpleLink Ultra-Low-Power Sub-1GHz Wireless MCU*, Texas Instruments, Dallas, TX, USA, Jul. 2018.
- [23] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, "Enabling RPL multihop communications based on LoRa," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.
- [24] J. Dias and A. Grilo, "LoRaWAN multi-hop uplink extension," *Procedia Comput. Sci.*, vol. 130, pp. 424–431, Dec. 2018.
- [25] A. Elsts, S. Duquenooy, X. Fafoutis, G. Oikonomou, R. Piechocki, and I. Craddock, "Microsecond-accuracy time synchronization using the IEEE 802.15.4 TSCH protocol," in *Proc. IEEE 41st Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Nov. 2016, pp. 156–164.
- [26] A. Elsts, X. Fafoutis, S. Duquenooy, G. Oikonomou, R. Piechocki, and I. Craddock, "Temperature-resilient time synchronization for the Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2241–2250, May 2018.
- [27] T. Istomin, C. Kiraly, and G. P. Picco, "Is RPL ready for actuation—A comparative evaluation in a smart city scenario," in *Proc. Eur. Conf. Wireless Sensor Netw.* Cham, Switzerland: Springer, 2015, pp. 291–299.

- [28] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2117–2157, 4th Quart., 2017.
- [29] D. Piester, A. Bauch, J. Becker, and A. Hoppmann, "Time and frequency broadcast with DCF77," in *Proc. 43rd Annu. Time Time Interval (PTTI) Syst. Appl. Meetings*, 2011, pp. 185–196.
- [30] G. K. Nelson, M. A. Lombardi, and D. T. Okayama, "NIST time and frequency radio stations: WWV, WWVH, and WWVB," *NIST Special Publication*, vol. 250, no. 67, p. 161, 2005.



RAJEEV PIYARE (Member, IEEE) received the Ph.D. degree in telecommunication from the University of Trento, Italy, in 2019, and the M.Eng. degree in electronics engineering from Mokpo National University, South Korea, in 2014. He was a Lecturer in electronics with Fiji National University from 2009 to 2012. From September 2017 to May 2018, he was a Visiting Research Scholar with the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He has coauthored more than 15 scientific articles. He is a member of several Technical Program Committees of conferences in the field of the IoT and energy management.



GEORGE OIKONOMOU received the M.Sc. degree in information systems and the Ph.D. degree in computer science from the Athens University of Economics and Business, Athens, Greece, in 2002 and 2009, respectively. He has worked as a Research Associate with the Faculty of Engineering, University of Bristol, and with the Computer Science Department, Loughborough University, U.K. He is currently a Lecturer with the Department of Electrical and Electronic Engineering, University of Bristol. His current research interests include energy-efficient networking for networks of severely constrained wireless embedded devices and the Internet of Things. He is a Co-Founder, a Steering Group Member, and a Maintainer of Contiki-NG, and an Active Developer of the Contiki open source embedded operating system for the Internet of Things.



ATIS ELSTS (Member, IEEE) received the Ph.D. degree in computer science from the University of Latvia, in 2014. He was with the Digital Health Engineering Group, University of Bristol, from 2016 to 2018, with the Swedish Institute of Computer Science (SICS), in 2015, and a Researcher with Uppsala University, from 2014 to 2015. Since December 2018, he has been a Researcher with the Institute of Electronics and Computer Science (EDI), Riga, Latvia. He is also a Maintainer of the Contiki-NG operating system for the Internet of Things (IoT). His scientific interests include experimental research in networked embedded systems, including network protocols, wearable devices, and embedded machine learning.

• • •