IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2024, accepted 22 November 2024. Date of publication 00 xxxx 0000, date of current version 00 xxxx 0000.

Digital Object Identifier 10.1109/ACCESS.2024.3508540

Energy-Aware Adaptive Scheduling for Battery-Less 6TiSCH Routers in Industrial Wireless Sensor Networks

DRIES VAN LEEMPUT[1], JEROEN FAMAEY[2], (Senior Member, IEEE), JEROEN HOEBEKE[1], AND ELI DE POORTER[1]

[1]IDLab, Department of Information Technology, Ghent University-imec, 9052 Ghent, Belgium
[2]IDLab, Department of Computer Science, University of Antwerp-imec, 2000 Antwerp, Belgium

Corresponding author: Eli De Poorter (eli.depoorter@ugent.be)

This work was supported by the Flemish Sustainable Internet of Batteryless Things (IoBaLeT) Project under Grant FWO SBO S001521N.

ABSTRACT Industrial Wireless Sensor Networks (IWSNs) using the 6TiSCH standard offer a scalable and cost-effective alternative to wired solutions in industrial environments, especially in hard-to-reach areas. Mains-powered devices face high installation costs and cable vulnerabilities, while battery-powered devices are limited by lifespan and maintenance challenges. Energy harvesting and supercapacitors offer promising alternatives with longer lifetimes and reduced maintenance. However, only battery-less end devices are usually considered. Due to the intermittent energy availability of battery-less devices and demanding network requirements, routers are assumed to be continuously powered. Therefore, this paper proposes, to the best of our knowledge, the first solution to integrate battery-less routers into 6TiSCH networks, building on previous work that used real-time traffic prediction models. We extend this by developing an energy consumption and storage prediction mechanism, enabling adaptive scheduling based on a node's available energy. The proposed adaptive algorithm dynamically modifies the Time Slotted Channel Hopping (TSCH) Scheduling Function to reduce the energy consumption of battery-less routers while triggering topology changes to ensure network reliability and adaptively route data based on dynamic energy availability. Evaluation of the algorithm in both small- and large-scale topologies demonstrates its effectiveness in reducing energy consumption and improving network performance by dynamically adjusting timing schedules. This approach significantly enhances the reliability and uptime of battery-less networks, although it introduces latency. Overall, the solution advances the development of fully energy-autonomous IWSNs, suitable for non-critical building automation and similar applications.

INDEX TERMS Industrial wireless sensor networks (IWNSs), energy harvesting, battery-less IoT, 6TiSCH, time slotted channel hopping (TSCH).

I. INTRODUCTION

In Industrial Wireless Sensor Networks (IWSNs), the IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) network stack [1] has emerged as a promising alternative to traditional wired industrial solutions, offering scalability, cost-effectiveness, and flexibility in environments where wired installations are impractical or impossible, such as hard-to-reach or hazardous areas. While mains-powered devices are commonly used in industrial networks, they come with several limitations. The installation of mains-powered devices is costly, and their reliance on physi cal cabling reduces both scalability and flexibility. Furthermore, these devices are susceptible to cable damage, which can lead to network failures in critical systems. In contrast, battery-powered devices, made feasible by the energy-efficient nature of 6TiSCH networks, offer greater flexibility in deployment. However, they present challenges of their own. Batteries

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Wang.

typically need to support a lifespan of over 10 years, as frequent replacements can be costly and disruptive. In highly regulated environments, such as those requiring ATEX certification, battery maintenance is even more complex and expensive. Moreover, the environmental impact of recycling batteries poses sustainability concerns, further complicating their long-term use in industrial networks [2], [3], [4].

Given these challenges, there is growing interest in powering IoT devices using energy harvesters and supercapacitors as storage elements. Supercapacitors offer significant advantages over batteries, including a longer operational lifespan, greater resilience to current peaks, superior performance in extreme temperatures, and a reduced risk of explosion in the event of a short circuit. Additionally, they are essentially maintenance-free, allowing for a ''deploy-and-forget'' approach. Energy harvesters convert energy sources, such as solar, kinetic, thermal, or Wireless Power Transfer (WPT), into usable power. However, their effectiveness can vary greatly depending on the deployment environment. By combining energy harvesters with supercapacitors, battery-less devices can be deployed in IWSNs, particularly in hazardous or hard-to-reach areas, where maintenance and power supply are major concerns.

The importance of integrating battery-less devices into multi-hop Industrial IoT (IIoT) networks was also recognized in [5]. However, this integration into IWSNs like 6TiSCH introduces significant challenges. The variable nature of energy harvesting, combined with the lower energy density and higher self-discharge rates of supercapacitors, creates an intermittent power supply. This intermittency stands in contrast to the stable power that IWSNs typically require. Especially the joining procedures, frequent synchronization, and idle listening can be problematic for battery-less devices, as they require more consistent energy availability than these devices can provide [6], [7], [8].

Previous research has explored various strategies to integrate battery-less end devices into IWSNs, including duty-cycled joining mechanisms [7] and hierarchical network management [8] to improve the likelihood of successful network participation and synchronization. Despite this progress, integrating battery-less routing devices, crucial for multi-hop networks, has not been thoroughly investigated. Our earlier work proposed strategies to integrate battery-less devices based on energy source, storage capacity, mobility, and network reliability [6]. It became evident that in multi-hop battery-less IWSNs, synchronization remains essential to minimize the energy consumption of routers.

This paper builds upon that foundation by presenting a novel solution for integrating battery-less routers into 6TiSCH networks. The proposed approach extends previous work that predicted the future traffic of 6TiSCH devices using an analytical model with real-time telemetry [9]. This work uses traffic prediction as a basis for an energy consumption model and a storage prediction mechanism for battery-less devices, allowing for real-time adaptation of energy consumption according to available resources. The TSCH Scheduling Function (SF) is locally modified using an energy-aware adaptive algorithm, which minimizes the impact on other nodes while reducing the energy consumption of the battery-less routers. Additionally, the algorithm enables dynamic topology changes to adapt the routes based on the availability of intermediary routers, thereby improving both the energy efficiency of these nodes and the reliability of child nodes. As a result, this work contributes to the broader goal of realizing fully energy-autonomous IWSNs.

To summarize, our main contributions are:

- Prediction of the future voltage level of a battery-less node's supercapacitor, based on energy predictions.
- Design of an adaptive TSCH scheduling algorithm that (i) adapts the unicast slotframe size of battery-less routers locally to control energy consumption based on the predicted voltage level and (ii) ensures child nodes switch to another router when nearing energy depletion. This does not affect control traffic or the unicast slotframe of other nodes.
- Implementation of an open-source[1] Cooja extension based on Energest to simulate battery-less nodes using a supercapacitor as a storage element.
- Implementation of the scheduling algorithm in Contiki-NG [10], provided open-source[1].

The remainder of this paper is structured as follows. First, Section II gives an overview of related works and Section III provides technical background on 6TiSCH, the analytical traffic model, and battery-less devices. Next, Section IV provides an architectural overview of the adaptive scheduling algorithm including the interaction between its components. Subsequently, the different components are described separately in Section V (energy model), Section VI (scheduling algorithm), and Section VII (in-band network telemetry). The adaptive scheduling algorithm is evaluated in Section VIII and Section IX concludes the paper.

II. RELATED WORK

Table 1 lists an overview of existing works in IWSNs that cover (i) monitoring, (ii) energy modeling, (iii) energy saving, and (iv) the integration of battery-less devices. As can be seen, while several works exist that cover each aspect individually, our work combines these techniques to provide a solution to integrate battery-less routers into 6TiSCH IWSNs.

In terms of monitoring solutions for 6TiSCH, Karaagac et al. [11] propose a lightweight telemetry solution for monitoring 6TiSCH using the Alternate Marking Performance Monitoring (AMPM) concept, with minimal overhead but limited to end-to-end and hop-by-hop monitoring. Graf et al. [15] provide a survey on Application Performance Monitoring (APM) for low-power wireless networks, focusing on TSCH systems. They demonstrate an active real-time APM implementation using the SmartMesh IP protocol stack

[1] https://github.com/imec-idlab/6tisch_predictive_algorithm

TABLE 1. Comparison of our work and related works covering (i) monitoring, (ii) energy modeling, (iii) energy saving, and (iv) support for battery-less devices in IWSNs.

Related work	Monitoring	Energy modeling	Energy saving	Battery-less devices	
				End nodes	Routers
[11]–[15]	✓	✗	✗	✗	✗
[16]–[20]	✗	✓	✗	✗	✗
[21]–[24]	✗	✗	✓	✗	✗
[6]–[8]	✗	✗	✓	✓	✗
Our work	✓	✓	✓	✓	✓

integrated with Zephyr, offering a reliable and deployable monitoring solution. Gajica [14] present an approach to monitoring Low-power and Lossy Networks (LLN) networks formed by VESNA sensor nodes running 6TiSCH. They integrate technologies such as MQTT for machine-to-machine communication and Zabbix for data visualization, showcasing a practical deployment with Zabbix graphs displaying various network readings. Gaillard et al. [13] and Karaagac et al. [12] both present methods for piggybacking telemetry in Information Elements (IEs) in 6TiSCH. In [12], IEs are added to data frames, creating an In-band Network Telemetry (INT) solution without extra control packets. In our previous work [9], we extended this by using Destination Advertisement Objects (DAOs) as telemetry carriers, enabling storing-mode compatibility and supporting a broader range of use cases. This formed the basis for an analytical traffic model, which we further expand with an energy consumption model in this work.

In the context of 6TiSCH energy models, Hajizadeh et al. [16] propose a scalable model for evaluating Medium Access Control (MAC) performance in large-scale TSCH-based networks. They consider various probabilities of successful communication in different timeslots of a TSCH slotframe, with performance metrics such as Packet Reception Probability (PRP), latency, and energy consumption. Kubaszek et al. [18] introduce an energy consumption model for 6TiSCH nodes operating in the 863-870 MHz band. Their model, verified through experimental measurements with 98% accuracy, includes layers for hardware current consumption, slot type modeling, and traffic analysis to determine the probability of each slot type. Scanzio et al. [20] present a mathematical model for TSCH based on real testbed measurements, focusing on reliability, power consumption, and latency. An experimental campaign on OpenMote B devices was conducted to better characterize power consumption, and results show trade-offs between the three performance indicators when optimizing for any one of them. Ouanteur et al. [19] propose a two-dimensional Markov chain model for the TSCH channel access mechanism. They provide theoretical expressions for metrics such as collision probability, data packet loss rate, energy consumption, throughput, delay, jitter, and reliability, while analyzing the effects of traffic conditions and the number of devices sharing a link. Daneels et al. [17] also develop an accurate energy consumption model for devices using TSCH, capturing all network-related Central Processing Unit (CPU) and radio state changes. In contrast to other works, we estimate the energy consumption by combining our existing analytical traffic model [9] with the energy consumption model of [17], using INT to collect network, traffic, and energy information.

Several studies have focused on optimizing energy consumption in 6TiSCH networks by reducing idle listening during normal TSCH operation. Fafoutis et al. [21] propose Adaptive Static Scheduling, selectively activating timeslots to avoid idle listening in unused slots. Scanzio et al. [22] introduce Proactive Reduction of Idle Listening (PRIL), which allows nodes to skip RX slots where idle listening is expected, using a MAC sleep command to suspend unnecessary listening. Hannachi et al. [23] improve energy efficiency during 6TiSCH network formation by optimizing control packet handling and parent switching, reducing energy usage, joining time, and jitter. In our previous work [24], we introduced a 6TiSCH Low-Power Node (6LPN) to optimize the energy-intensive joining process and eliminate idle listening by queueing downlink traffic through a 6TiSCH Friend Node (6FN). This achieved significant power savings, up to 94% during operation, while maintaining backward compatibility with 6TiSCH. In contrast to these works, we reduce idle listening of battery-less nodes to limit energy consumption when their energy storage level is nearly depleted. In addition, parent switching is triggered to relieve nearly depleted routers of forwarding and to enhance reliability.

Other studies explore integrating battery-less devices using energy harvesting in IWSNs networks. Chew et al. [7] propose a duty-cycled network joining strategy for energy-harvesting nodes, where nodes join during gaps between other joining attempts, minimizing energy wastage while waiting. By limiting advertisement channels, the initial scanning time is shortened, especially helpful when multiple nodes attempt to join simultaneously. Das et al. [8] introduce a hierarchical management approach, where a central system manager allocates resources for initial joining, while local routers manage constrained devices. This method also reduces scanning time by limiting advertisement channels, ensuring efficient network integration for energy-harvesting nodes. Our previous work [6] tackles the challenges of integrating battery-less devices with supercapacitors in multi-hop industrial wireless sensor networks. Three strategies were proposed to address intermittent power issues, focusing on

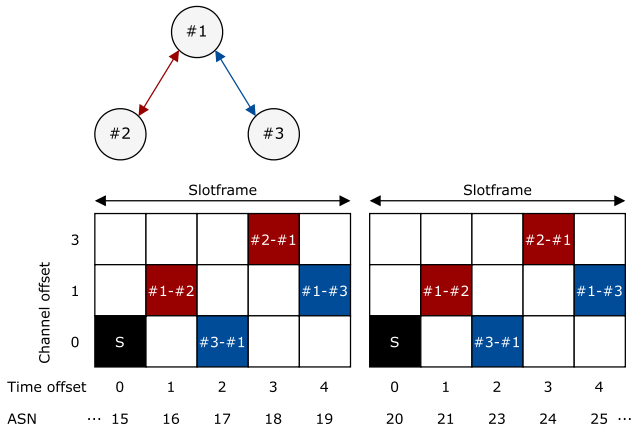


FIGURE 1. Example TSCH schedule for a three-node network. Black slots indicate shared slots, whereas blue and red slot represent dedicated slots.

energy source type, storage capacity, device mobility, and network reliability, to enhance the integration of energy-harvesting devices. While the above works focused on the integration of battery-less end devices into IWSNs, we present, to the best of our knowledge, the first solution to integrate battery-less routing devices into synchronized 6TiSCH networks.

III. TECHNICAL BACKGROUND

This Section provides an overview of the 6TiSCH network stack and some details about the analytical traffic model of 6TiSCH [9], which forms the basis of the energy model described later in this work. Finally, a brief introduction on battery-less devices and intermittency concludes this Section.

A. 6TiSCH NETWORK STACK

The 6TiSCH protocol stack [1], developed by the IETF working group, is designed to ensure reliable and deterministic communication in industrial environments. Built on the TSCH MAC layer [25], it supports IPv6 through the 6LoWPAN adaptation layer. TSCH operates by dividing time into slots, which can either be dedicated to a pair of nodes or shared by multiple nodes. Each timeslot allows for sending a data frame and, for unicast transmissions, the reception of an Enhanced Acknowledgement (EACK). IEs in both frames and acknowledgments help nodes share network information. Timeslots are organized into repeating slotframes, with an Absolute Slot Number (ASN) tracking time across the network. TSCH also uses channel hopping to improve reliability, where each time a different frequency is assigned to a specific link. An example TSCH schedule for a simple network of three nodes is depicted in Fig. 1, with a shared slot at channel offset and time offset 0 in black and dedicated slots in red and blue.

Nodes also send Enhanced Beacons (EBs) to synchronize and share network details. Routing in 6TiSCH relies on the Routing Protocol for Low-power and Lossy Networks (RPL), which builds a Destination Oriented Directed Acyclic Graph (DODAG) for routing traffic. Each node is assigned a rank, which increases with distance to the 6LoWPAN Border Router (6LBR). The rank calculation is managed by an Objective Function (OF), of which OF0 [26] and Minimum Rank with Hysteresis Objective Function (MRHOF) [27] are the most frequently used. OFs use link metrics in the rank calculation, which indicate the quality of a link. A popular choice is Expected Transmission Count (ETX), which indicates the average number of transmissions that are needed through this link before successful reception. Depending on the mode, communication may either route through the 6LBR (non-storing mode) or use local routing tables (storing mode). DODAG Information Object (DIO) and DAO messages manage routing for uplink and downlink traffic, respectively.

To manage TSCH links, an SF assigns slot and channel offsets, with popular options being the 6TiSCH Minimal Scheduling Function (MSF) [28] and Orchestra [29]. In this work, we use the Orchestra Receiver Based Scheduler (RBS) SF, which uses three concurrent slotframes: an advertising slotframe for EB messages, a broadcast slotframe for RPL traffic, and a unicast slotframe for data traffic. The latter comprises one RX slot per node, during which neighbors transmit data to the node. To minimize collisions between the three slotframes, their lengths should be mutually prime. If slot collisions do occur, the slotframe with highest priority executes its slot (advertising broadcast unicast). An RBS SF is written as TSCH-RB-X-Y-Z, with X, Y, and Z being the slotframe lengths of the advertising, broadcast, and unicast slotframes, respectively.

B. ANALYTICAL TRAFFIC MODEL FOR 6TiSCH

The analytical traffic model [9] predicts the number of bytes transmitted and received by each node in a 6TiSCH network during a specified prediction interval. Since the model operates centrally, telemetry data, including network topology and timing, is gathered using INT. This approach reduces both computational and communication demands, making it ideal for energy-constrained devices. Furthermore, real-time telemetry eliminates the need for initialization and allows for dynamic adaptation to changes in network topology. The model also takes data traffic information as input and calculates the introduced overhead by the network. This enables the calculation of the number of TX and RX bytes for each node, as well as the slot type used for each interaction (either shared or dedicated TSCH slots).

The model consists of three sub-models, each addressing a different type of overhead: control traffic overhead, single-hop overhead, and multi-hop overhead. The control traffic sub-model calculates the overhead of TSCH and RPL control frames, which includes periodic EBs, DIOs, and DAOs, taking into account the synchronization interval, Trickle timer [30], and Prefix Lifetime, respectively. The

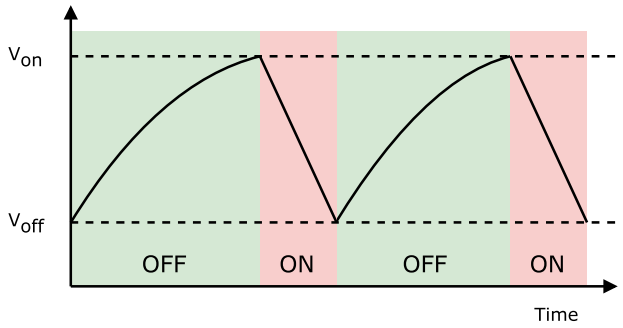


FIGURE 2. Intermittent on-off behavior of a battery-less device. The supercapacitor is charged by the energy source to the turn-on threshold, permitting the device to turn on and execute tasks. This depletes the supercapacitor until the turn-off threshold is reached, forcing the device to power off and allowing the supercapacitor to be recharged by the energy source. Reprinted from [6] with permission of the authors.

single-hop overhead sub-model accounts for 6TiSCH headers at each network layer, depending on the frame type and its routing path, and considers factors like IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) fragmentation and Constrained Application Protocol (CoAP) Block transfer. Finally, the multi-hop overhead sub-model estimates the overhead of nodes along a multi-hop route from source to destination, using the topology information obtained via INT.

For a detailed explanation of the calculations used to solve the analytical model, please refer to [9].

C. BATTERY-LESS DEVICES AND INTERMITTENCY

Battery-less devices rely on supercapacitors to store energy, which is harvested from their environment through energy harvesters. Due to the lower energy density of supercapacitors compared to batteries, and because of the irregular power supply of the energy harvester, battery-less devices experience intermittent on-off behavior. This is depicted in Fig. 2, where the supercapacitor charges to a threshold voltage V_{on}, enabling the device to function for a time before the energy depletes to V_{off}, causing the device to shut down. Once depleted, the supercapacitor can be recharged. Most battery-less devices use a Power Management Unit (PMU), that regulates the energy conversion from the energy source to the supercapacitor and from the supercapacitor to the Internet of Things (IoT) device. As such, it determines the threshold voltages V_{on} and V_{off}. The energy conversion has inevitable losses, which is determined by the efficiency of the PMU. The timing of these power cycles depends on factors such as the device type, application, and available power for energy harvesting. In some cases, the device may turn off upon reaching V_{off}, while in others, it may intentionally shut down or enter a low-power mode before reaching that limit [6]. The capacitor must always store enough energy to support the most power-hungry operation, which is the joining procedure in the case of TSCH-based networks [6], [7], [8].

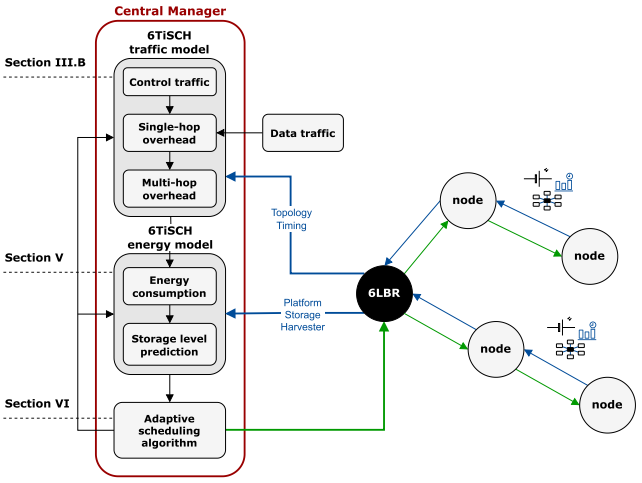


FIGURE 3. Architectural overview of the required components to execute the adaptive scheduling algorithm, which is executed by a central manager, highlighted in red, connected to the 6LBR. Blue lines indicate INT to solve the analytical traffic and energy models, whereas green lines indicate a decision of the scheduling algorithm toward the nodes.

IV. ARCHITECTURAL OVERVIEW

Fig. 3 provides an overview of the necessary components to execute the adaptive scheduling algorithm, enabling the integration of battery-less routers into a 6TiSCH network. To limit computation and energy consumption overhead on the nodes, a central manager, connected to the 6LBR, is responsible for executing the adaptive scheduling algorithm and solving the required models to generate input for the algorithm. Therefore, as mentioned in Section III-B, the INT strategy proposed in [12] is used to provide network telemetry from the nodes to the 6LBR, which forwards this information to the central manager, shown by blue arrows in Fig. 3. The telemetry comprises topology, timing, storage, and energy information.

First, the 6TiSCH traffic model (Section III-B) calculates the number of transmitted and received bytes for every node in the network during a future prediction interval, using topology and timing information of the network nodes. Second, a 6TiSCH energy model estimates the energy consumption of each node during the prediction interval, based on the output of the traffic model and energy consumption parameters of the employed platform of each node. Third, the future storage level of each node is estimated based on the current storage level, current harvesting power, and the estimated energy consumption provided by the energy model. Finally, the adaptive scheduling algorithm takes appropriate action based on the predicted storage levels, mitigating the risk of battery-less nodes disconnecting from the network due to energy depletion. The algorithm adapts the local schedule of these nodes by changing the number of executed RX slots, which allows tuning idle listening and hence the energy consumption. The necessary changes to the local schedule are fed back to the respective nodes (green arrows in Fig. 3), and to the traffic and energy models.

The following Sections will describe the 6TiSCH energy model (Section V) and the adaptive scheduling algorithm (Section VI) individually. In addition, Section VII discusses the use of INT and the associated overhead when employing the adaptive scheduling algorithm.

V. ENERGY CONSUMPTION MODEL FOR 6TiSCH

Using the results of the analytical traffic model, the energy consumption of 6TiSCH nodes during a prediction interval can be estimated, in addition to their storage level after the prediction interval. The energy consumption and storage level estimation will be covered separately.

A. ENERGY CONSUMPTION

The estimation of energy consumption is achieved by integrating an energy consumption model for TSCH, as proposed by Daneels et al. [17], into our analytical traffic model. The model defines various timeslot types and offers a detailed analysis of the distinct radio and CPU states during each timeslot type. The classification of timeslots depends on the traffic type (unicast, multicast, or advertising) and its direction (transmitting or receiving). For instance, unicast traffic requires Acknowledgements (ACKs) and is, therefore, transmitted or received during *TxDataRxAck* or *RxDataTxAck* timeslots, respectively. In contrast, multicast and advertising traffic occupy *TxData* and *RxData* slots as they are not acknowledged. *RxIdle* slots represent timeslots during which a node passively listens for incoming traffic. All other slots are considered as *Sleep* slots, during which the node enters a Low-Power Mode (LPM) until waking up for the next active slot. Note that *TxIdle* slots are not defined, as nodes will skip TX slots without pending frames.

Given the duration (t_{state}) and current consumption (I_{state}) for each timeslot state, the energy consumption during a single timeslot (E_{slot}) can be computed as shown in (1), where V_{ref} represents the reference voltage at which I_{state} is obtained.

$$E_{slot} = \sum_{state \epsilon slot} \Delta t_{state} \times I_{state} \times V_{ref} \tag{1}$$

I_{state} and V_{ref} depend on the chosen platform. Additionally, most state durations t_{state} are platform-specific and slot-type specific, except for data-related states such as transmitting and buffer operations. Therefore, (1) can be written as (2), where E_t represents the energy consumption of a specific timeslot type and S denotes the size of the frame transmitted or received. Naturally, for *RxIdle* slots, S is zero.

$$E_{slot} = E_t(S) \tag{2}$$

As a result, the analytical traffic model in our previous work can be coupled with the TSCH energy model [17] to predict the energy consumption over a specified time interval. Equations in [9] enable the calculation of (i) the number of timeslots for each timeslot type and (ii) the number of transmitted or received bytes for each timeslot type. This leads to (3), where $\mathcal{TS}$ represents the set of timeslot types, $N_t(T_p)$ the predicted number of timeslots of type t in $\mathcal{TS}$ during the prediction interval T_p, and S_i denotes the frame size transmitted or received within timeslot i.

$$E(T_p) = \sum_{t \epsilon \mathcal{TS}} \sum_{i=1}^{N_t(T_p)} E_t(S_i) \tag{3}$$

TxDataRxAck and *RxDataTxAck* timeslots are obtained by considering transmitted and received unicast frames, respectively. In a 6TiSCH network, unicast traffic comprises CoAP and DAO (ACK) messages. *TxData* and *RxData* encompass multicast and advertising traffic, representing DIO and EB messages. For a detailed description on how to calculate the number of timeslots and their associated frame lengths for each timeslot type, we refer to the equations in [9]. However, the number of *RxIdle* slots cannot be inferred directly from the analytical traffic model, as no data exchange occurs. Instead, the number of *RxIdle* slots N_{RxIdle} is calculated in (4), where $\mathcal{SF}$ represents the set of slotframes used in the TSCH SF, $N_{rx,s}$ the number of scheduled RX slots within one slotframe duration T_s of slotframe s, and $N_{rx,act,s}$ denotes the number of active RX slots during the T_p for slotframe s.

$$N_{RxIdle} = \sum_{s \epsilon \mathcal{SF}} \left(\frac{N_{rx,s} T_p}{T_s} - N_{rx,act,s} \right) \tag{4}$$

$N_{rx,act,s}$ relies on the chosen SF, which determines the available slotframes. In most SFs, there is either a single slotframe or multiple slotframes, where one is dedicated for multicast and/or advertising traffic, and another for unicast traffic. As a result, $N_{rx,act,s}$ equals the total number of active RX timeslot types (*RxDataTxAck* and *RxData*) for SFs featuring a single slotframe. For slotframes dedicated to unicast, $N_{rx,act,s}$ equals the number of unicast timeslots (*RxDataTxAck*). For advertising or multicast slotframes, $N_{rx,act,s}$ equals the number of multicast/advertising slots (*RxData*). Certain SFs split multicast and advertising slots into two different slotframes (e.g., Orchestra [29]). In this case, EB traffic is considered within the EB slotframe, while DIO traffic is accounted for in the broadcast slotframe. For an Orchestra TSCH-RB-X-Y-Z schedule, (4) can be written as (5), where *EBS* represents the EB slotframe (size X), *BCS* the broadcast slotframe (size Y), *UCS* the unicast slotframe (size Z), and T_{ts} the timeslot duration.

$$\begin{aligned} N_{RxIdle} = & \left(\frac{N_{rx,EBS} T_p}{X T_{ts}} - N_{rx,EB} \right) \\ & + \left(\frac{N_{rx,BCS} T_p}{Y T_{ts}} - N_{rx,DIO} \right) \\ & + \left(\frac{N_{rx,UCS} T_p}{Z T_{ts}} - N_{rx,CoAP} - N_{rx,DAO} \right) \end{aligned} \tag{5}$$

B. STORAGE LEVEL PREDICTION: SUPERCAPACITOR

By using the energy model, energy storage levels of constrained devices can be anticipated. To predict the future voltage level of a battery-less node's capacitor, we use

the capacitor model introduced in [31] and adapted in [2] and [3]. This model includes realistic phenomena for energy harvesting devices, such as capacitor leakage and PMU efficiency. The capacitor voltage after a prediction interval T_p is determined by (6), where V_0 represents the voltage at the start of the prediction interval, C the capacitance, and V_{ref} the reference voltage of the platform and energy harvester. The consumed energy E'_c during the prediction interval accounts for capacitor leakage and PMU efficiency, as per (7). Here, $\eta_{PMU,l}$ is the efficiency of the PMU when transferring power to the load (platform) and P_{leak} the leakage power of the capacitor.[2] Similarly, the harvested power P'_h is adjusted in (8), where $\eta_{PMU,h}$ denotes the efficiency of the PMU when charging the capacitor.

$$V_p = V_0 e^{\frac{-E'_c}{V_{ref}^2}C} + \frac{V_{ref} T_p P'_h}{E'_c}\left(1 - e^{\frac{-E'_c}{V_{ref}^2}C}\right) \tag{6}$$

$$E'_c = E_c \eta_{PMU,l} + \frac{P_{leak}}{T_p} \tag{7}$$

$$P'_h = P_h * \eta_{PMU,h} \tag{8}$$

VI. ENERGY-AWARE SCHEDULING ALGORITHM

Based on the predicted capacitor voltage level, the adaptive scheduling algorithm determines whether adjustments to the TSCH SF are necessary to prevent a battery-less node from dropping below the turnoff voltage and consequently disconnecting from the network. We use the Orchestra RBS [29] as the basis of our adaptive scheduling algorithm due to several advantages it offers. First, Orchestra is among the most widely adopted TSCH SFs, facilitating integration into existing networks. Second, it operates as an autonomous scheduler, allowing nodes to compute their schedules locally without relying on a central manager or engaging in slot negotiations with neighboring nodes. This reduces control overhead, benefiting energy consumption. Third, Orchestra RBS defines three concurrent slotframes: one for advertising traffic (EBs), one for control and multicast traffic (DIO and, occasionally, DAO), and one for unicast traffic (CoAP and DAO). This separation enables us to adjust the unicast slotframe without affecting control traffic or network stability. Finally, each node has one dedicated RX slot per unicast slotframe, which minimizes energy consumption compared to the Sender Based Scheduler (SBS) alternative, where one RX slot is foreseen for every neighbor.

The size of the unicast slotframe is a trade-off between latency and energy consumption. A shorter slotframe size minimizes latency but increases energy consumption due to frequent idle listening (*RxIdle* slots), especially in networks with limited data traffic. Since battery-less nodes typically send periodic updates to a server with low periodicity, we can reduce energy consumption by increasing their unicast slotframes. However, small slotframes offer low latency, often necessary in IWSNs. Therefore, the slotframe size of non-energy-constrained nodes should not be altered. To balance these factors, we propose adapting the unicast slotframe on a per-node basis by introducing a *divisor* parameter. This parameter allows battery-less nodes to execute only a subset of RX slots while maintaining the original slotframe structure. In practice, this means executing one RX slot followed by skipping multiple RX slots. For a certain ASN, the RX slot is executed if (9) is satisfied, where *time_offset* represents the time offset of the unicast RX slot and S_{sf} denotes the size of the unicast slotframe. An example of this concept is depicted in Fig. 4, which show the unicast slotframe with $time_offset = 1$, $S_{sf} = 4$ and a *divisor* value of 1, 2, and 3. As shown, idle listening is reduced with increasing *divisor*.

$$time\_offset = ASN \pmod{S_{sf} \times divisor} \tag{9}$$

This ensures that the adjustment to the unicast slotframe occurs locally, significantly reducing idle listening for the battery-less node. Furthermore, TX slots remain unaffected, preserving the latency from the battery-less node to the server. However, the end-to-end latency from the child nodes is impacted due to the reduced unicast slotframe size of the battery-less routers.

Algorithm 1 Adaptive Scheduling Algorithm

```
Input: V_th, V_on
Output: divisor
divisor ← 1
V_p ← calculate_vp(divisor)
while (V_p < V_th and divisor ≤ max_divisor) do
    divisor ← divisor + 1
    V_p ← calculate_vp(divisor)
end while
while (V_p > V_on and divisor ≥ 1) do
    divisor ← divisor − 1
    V_p ← calculate_vp(divisor)
end while
return divisor
```

The adaptive scheduling algorithm (Algorithm 1) determines the most suitable *divisor* after voltage prediction (**calculate_vp()** as explained in Section V-B). It initializes the *divisor* to one and increments it until the predicted voltage, associated with the current *divisor*, no longer falls below a predetermined voltage threshold V_{th}. This threshold should be set higher than V_{off} to allow sufficient time for the algorithm to react to a voltage drop. Similarly, the algorithm decrements the divisor until the predicted voltage V_p no longer exceeds V_{on}. This ensures that the *divisor* adapts accordingly to changes in harvested power, improving latency when necessary.

The selection of V_{th} and V_{on} is essential to the functionality of the algorithm, as the voltage will oscillate between these thresholds. A carefully chosen V_{th} must consider both the prediction interval of the energy model and the device's energy

[2]For simplicity, we assume the leakage to be constant, while it actually depends on the voltage level of the capacitor.

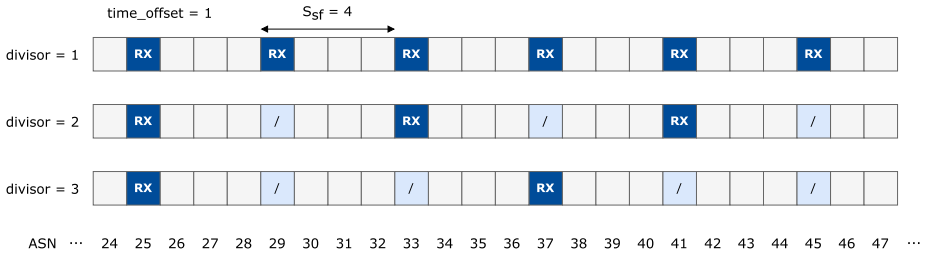


FIGURE 4. Example unicast slotframe schedule for a *divisor* of 1, 2, and 3. With increasing *divisor*, idle listening is reduced to limit energy consumption.

consumption. Given that the algorithm can only respond after each prediction interval, the supercapacitor must hold sufficient energy at V_{th} to sustain the device throughout the prediction interval, in the absence of harvesting power, without dropping below V_{off}. In other words, the energy consumption during a prediction interval should align with the stored energy in the capacitor between V_{th} and V_{off}. This ensures that in the event of an immediate drop in harvesting power after a prediction at V_{th}, the algorithm can increase the *divisor* at the subsequent prediction interval before the voltage descends to V_{off}. V_{th} can be approximated by using (10), which approximates the remaining energy in a capacitor between two voltage levels [2], [32].

$$E_{stored} \approx \frac{C}{2}\left(V_{max}^2 - V_{min}^2\right) \tag{10}$$

Following (10), V_{th} is given in (11), where E_p represents the energy consumption during the prediction interval, with a *divisor* of 1 applied to account for the worst-case scenario.

$$V_{th} \sim \sqrt{\frac{2 * E_p}{C} + V_{off}^2} \tag{11}$$

V_{on} can be inferred from the energy consumption during joining (E_{join}), as joining accounts for the largest energy consumption period for TSCH devices. Therefore, the stored energy between V_{on} and V_{th} should be equal E_{join}. This enables a device to initially join the network and wait for a prediction interval before the algorithm can determine a suitable divisor, without reaching V_{off}. Therefore, V_{on} can be calculated using (12).

$$V_{on} \sim \sqrt{\frac{2 * E_{join}}{C} + V_{th}^2} \tag{12}$$

The choice of V_{on} and V_{th} is illustrated in Fig. 5, which shows a scenario where a node joins the network when powering on (at V_{on}) and reaches V_{off} after a prediction interval T_p in the absence of harvesting power.

The adaptive scheduling algorithm's decision at the central manager is immediately communicated to the battery-less node if the *divisor* is updated. Given the lack of guaranteed downlink frames to the battery-less node, INT cannot be used, and the decision must be relayed quickly. Therefore, a dedicated User Datagram Protocol (UDP) message is sent from the 6LBR to the battery-less node, which immediately

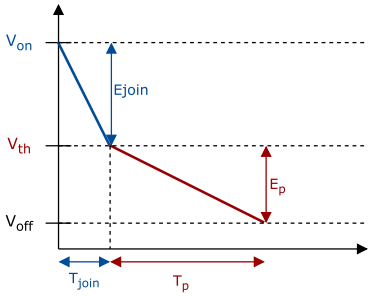


FIGURE 5. Selection of V_{th} and V_{on} depending on the joining energy E_{join} and predicted energy E_p during the prediction interval T_p. V_{on} and V_{off} are chosen to execute a joining procedure and one prediction interval without powering off.

updates its schedule. To keep neighbors informed about the battery-less node's current *divisor*, this value is included in the Reserved field of DIOs (see [33] for more details), which are received by all neighbors of the battery-less node. Additionally, upon receiving a new *divisor* from the central manager, the battery-less node immediately schedules a DIO to notify its neighbors of the updated schedule.

This process is illustrated in Fig. 6, which shows the time schedule of a battery-less router and two neighbors, along with their interactions. When a neighbor receives a DIO with an updated *divisor* (as seen with Neighbor 1 in Fig. 6), it skips the TX slots to the battery-less node that correspond to the battery-less node's skipped RX slots. Furthermore, the link metric toward the battery-less node is adjusted to make it less attractive as an RPL parent based on its *divisor* value. Specifically, the current ETX value is increased by *divisor* − 1, which matches the number of skipped slots. Consequently, child nodes may choose another parent that is either better suited or not nearly depleted, benefiting the battery-less node by reducing the need to forward traffic from child nodes.

However, since DIOs are not acknowledged, some neighbors might not receive a DIO and thus may not update their schedule. This is depicted with Neighbor 2 in Fig. 6. These neighbors will then expect the battery-less node to wake up at all scheduled RX slots, leading to potential retransmissions until the battery-less node wakes up at the next RX slot. Although this decreases link reliability, frames

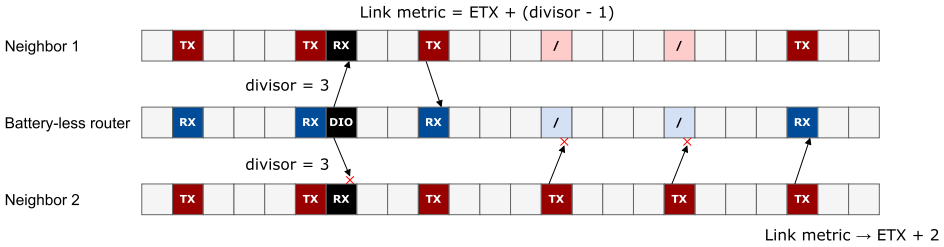


FIGURE 6. Time schedule and interactions between a battery-less router and two neighboring nodes, following a *divisor* update. Neighbor 1 successfully receives a DIO with the updated *divisor* and adjusts its transmission schedule accordingly, while Neighbor 2 continues to expect the battery-less node to wake up at all scheduled RX slots, resulting in potential retransmissions.

TABLE 2. Required telemetry for the analytical traffic and energy model, including size in bytes. Adapted and extended from [9].

Model	Telemetry	Size
Traffic model	Neighbors	N B
	Preferred parent	1 B
	Routing table	$2R$ B
	Time source	1 B
Traffic model (optional)	ETX	2 B
	TX ASN	2 B
	EB ASN	4 B
Energy model	Capacitor voltage	2 B
	Harvesting power	2 B
	Energy ASN	2 B

in TSCH networks are retransmitted until acknowledged within a predefined number of retries. Increasing the number of retries can mitigate the impact on reliability. Moreover, the ETX value from the child node to the battery-less node will increase to a value similar to other neighbors, as an average of $divisor - 1$ extra transmissions will be required, corresponding to the number of skipped RX slots. Eventually, neighbors that do not initially receive a DIO will receive a subsequent one, allowing them to only send updates during active RX slots.

VII. IN-BAND NETWORK TELEMETRY AND SCHEDULER'S OVERHEAD

The INT extension in [9] enabled the use of DAOs to piggyback telemetry to the 6LBR in storing mode, which is being used in this work as well. This results in an average telemetry update every 15 min for a default Prefix Lifetime of 30 min. However, apart from periodical DAOs, they are also triggered upon a topology change, which allows the algorithm to quickly adapt to a topology change while limiting the telemetry overhead during stable periods. Depending on the use case, network configuration, and harvesting capabilities, the Prefix Lifetime may be changed to achieve a higher or lower telemetry update interval.

Table 2 lists the required telemetry for the traffic and energy models. It includes the telemetry size, where N and R represent the number of neighbors and routes, respectively. Some telemetry is required for correct execution of the model, whereas some telemetry is optional to increase the accuracy of the model for shorter prediction intervals, such as additional timing information and the current ETX value. Each node is assigned a 1 B ID by the 6LBR (e.g., by using a hash table of the MAC address), which results in 256 possible nodes in the network. The bottom part of Table 2 lists the required telemetry for the energy model, which includes the voltage over the capacitor and harvesting power. In addition, a timestamp is included that represents the time at which the voltage and harvesting power were obtained.

Although INT minimizes the introduced overhead by piggybacking information onto existing control frames, additional bytes must be transmitted and received by all nodes along the path to the 6LBR. Furthermore, the algorithm's decisions are communicated to battery-less nodes through UDP messages, and new DIOs are triggered upon receiving these decisions. This results in a certain amount of overhead, inherent to the centralized nature of the analytical models and the algorithm. According to Table 2, the number of INT bytes in storing mode can be calculated using (13), where S_{fint} equals 8 and 16 with and without the optional telemetry, respectively. Given the Maximum Transmission Unit (MTU) of 127 B in IEEE 802.15.4 [25], the number of INT bytes must be less than or equal to $127 - S_{DAO}$.

$$S_{int} = S_{fint} + N + 2R \leq 127 - S_{DAO} \tag{13}$$

With a typical S_{DAO} of 86 B in storing mode, this leaves a maximum of 41 B or 33 B for INT. This restricts the number of neighbors and routes a battery-less node can have, thus limiting the solution's scalability. However, this issue can be mitigated by limiting the number of RPL neighbors a node can have, which is already a favorable solution to reduce the multi-hop overhead for battery-less nodes.

The algorithm's decision is sent to the battery-less node using an UDP message with a payload of 1 B. In addition, the battery-less node will broadcast one DIO with length S_{DIO}, which is received by all its neighbors.

The associated energy overhead of sending and receiving UDP decisions and DIOs can be calculated using the energy consumption model, as the number of bytes can be calculated using the equations of the analytical traffic model. However, the energy overhead associated with INT must be calculated

TABLE 3. Current consumption of the Energest states for the nRF52840 platform [34].

	Energest state	nRF52840
CPU	LPM ($\geq$ 32 kHz ST)	3.16 µA
	Active ($\geq$ 16 MHz GPT)	6.3 mA
Radio	RX	6.53 mA
	TX (0 dBm)	6.4 mA

using (14), where E_{int} represents the energy consumption of sending/receiving all DAOs including INT, and E_{dao} represents the energy consumption of sending/receiving the same number of regular DAOs. After all, these DAOs would be sent even without the algorithm.

$$E_{int,dao} = E_{int} - E_{dao} \tag{14}$$

VIII. EVALUATION

This Section assesses the implementation of the adaptive scheduling algorithm in a simulated 6TiSCH network, using both small-scale and large-scale topologies. The algorithm was integrated into the Contiki-NG Operating System (OS) [10] and evaluated through the built-in Cooja simulator, alongside the Energest module, to analyze its performance and energy consumption. Energest precisely tracks the time spent in various CPU and radio states, enabling energy consumption estimation for any platform by combining the platform's current consumption data from its datasheet with Energest's CPU and radio timings. In our simulations, the nRF52840 platform [34] was used, with its current consumption values for each Energest state presented in Table 3. Energest distinguishes two radio states (RX and TX) and two CPU states: LPM and Active. During LPM, Contiki-NG mandates the use of a Sleep Timer (ST) of at least 32 kHz, resulting in a current draw of just 3.16 µA for the nRF52840. Conversely, the Active state requires a more precise General Purpose Timer (GPT) of at least 16 MHz, leading to a current consumption of 6.3 mA. Despite the clear distinction in timing requirements between the LPM and Active states in Contiki-NG, support for STs is currently lacking [24]. Consequently, a GPT is used even during LPM in real-world TSCH implementations on Contiki-NG. Given that this work targets energy-constrained, battery-less devices in 6TiSCH networks, the resulting 2000-fold increase in LPM current consumption (from 3.16 µA to 6.3 mA) is impractical. Therefore, our evaluation is based on simulations using Energest, allowing us to model a much lower LPM current draw as specified in the datasheet. To more accurately simulate battery-less devices, including energy harvesting and supercapacitors, Section VIII-A introduces an open-source extension to the Cooja simulator. This extension helps to evaluate the adaptive algorithm but also enables the evaluation of future solutions aimed at integrating battery-less devices into 6TiSCH networks. Section VIII-B evaluates the algorithm in a small-scale topology, showing the ability of the algorithm to adapt the energy consumption of battery-less routers to their storage level and for children to switch to the best-suited parent. Finally, Section VIII-C evaluates the algorithm in a large-scale battery-less 6TiSCH network. Table 4 lists the network, energy, and simulation parameters for both topologies.

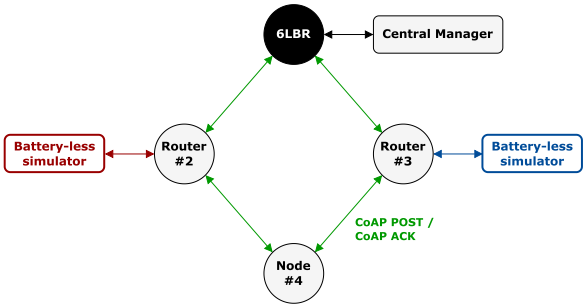


FIGURE 7. Two-router topology to evaluate the adaptive scheduling algorithm. Node #4 periodically transmits a Confirmable CoAP POST message to the 6LBR, which replies with a CoAP ACK. Both messages are forwarded by battery-less Router #3 or battery-less Router #2.

A. SIMULATING BATTERY-LESS 6TISCH NODES IN COOJA

To simulate battery-less nodes in Cooja, each battery-less node is connected to a Python script via a serial connection. This connection enables the node to report its real-time energy consumption data using the Energest module. Following the execution of each non-idle TSCH timeslot, an update containing energy consumption details is sent to the battery-less simulator. The device's CPU is assumed to operate in LPM if no slot is executed, and in Active mode for the entire duration of an active slot, except for RX idle slots, where the CPU enters LPM after scanning for frames.

The simulator employs the same capacitor model used in Section V to calculate the voltage over a supercapacitor, albeit with the actual energy consumption instead of the predicted consumption. The Python script configures the supercapacitor and can provide a variable harvesting power. Before transmitting a telemetry update to the 6LBR, the node queries the battery-less simulator for updates on voltage and harvesting power. This information is then included in a DAO using INT (see Table 2). If the voltage over the supercapacitor reaches V_{off}, a *shutdown* command is sent to the node in Cooja, which will cause it to leave the network. Subsequently, the battery-less simulator calculates the duration required to recharge the capacitor to V_{on}, assuming the given harvesting power, and transmits a *power* command to the simulated node, causing it to rejoin the 6TiSCH network.

B. TOPOLOGY WITH TWO BATTERY-LESS 6TiSCH ROUTERS

Employing the Cooja extension described in the previous Section, the adaptive scheduling algorithm is evaluated using a topology featuring two battery-less routers, as depicted in Fig. 7. Node #4 periodically transmits a Confirmable CoAP POST message to the 6LBR, which replies with a

TABLE 4. Network, harvesting, and simulation configuration parameters used in the simulations.

Parameter	Small-scale	Large-scale
CoAP POST payload	7 B	
CoAP ACK payload	0 B	/
CoAP interval	1 min	5 min
RPL OF	MRHOF	
Link metric	ETX + divisor - 1	
Parent switch threshold	1.5	
TSCH Scheduling Function	TSCH-RB-397-62-17	TSCH-RB-397-62-21
Advertisement channels	2	
Communication channels	16	
Platform	nRF52840	
Capacitance	200 mF	
Operating range	[1.8 V, 5 V]	
Capacitor leakage	2 µA	
PMU efficiency	80 %	
V_{on}	2.5 V	
V_{off}	3.5 V	
P_{harv}	[200, 1200 µW]	[200, 800 µW]
P_{harv} variation	/	[−100, 100 µW]
P_{harv} period	20 min	5 min
Radio medium	Logistic loss	
TX range	120 m	
RX sensitivity	−100 dBm	
$RSSI_{50\%}$	−92 dBm	
Path loss exp.	3	
σ_{AWGN}	3	
Simulation duration	3 h40 min	9 h
Simulation runs	1	20

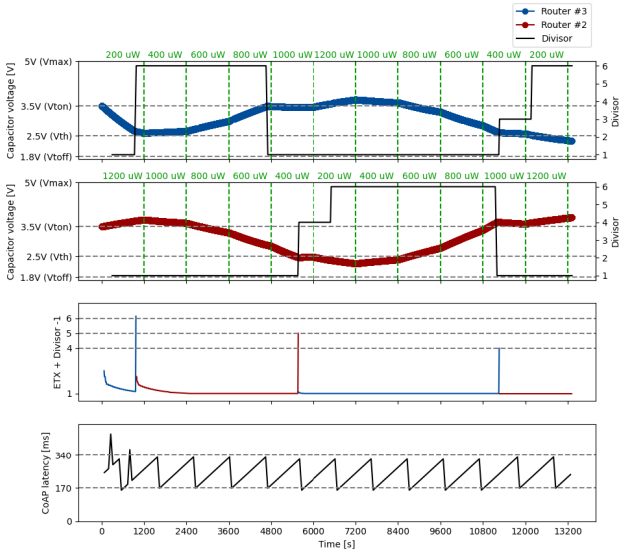


FIGURE 8. Capacitor voltage for two battery-less routers employing the adaptive scheduling algorithm, including the ETX and CoAP latency of a child node. Black lines indicate the divisor value, whereas green vertical lines indicate a change in harvesting power, accompanied by their corresponding values. ETX is displayed in blue if Router #3 is preferred parent and in red if Router #2 is preferred parent.

CoAP ACK. Both Router #2 and #3 are situated within communication range of Node #4 and the 6LBR, forwarding frames toward their destination. The routers employ the battery-less simulator to simulate an energy harvester, PMU, and supercapacitor. The 6LBR is connected to the central manager responsible for executing the adaptive scheduling algorithm.

As shown in Table 4, MRHOF [27] is used as OF with a parent switch threshold of 1.5. The ETX link metric is adjusted to account for the divisor value, as discussed in Section VI. As SF, the Orchestra RBS TSCH-RB-397-62-17 is used [29]. The number of advertisement channels is reduced to 2, which reduces joining time and energy consumption [6], [24]. A capacitance of 200 mF was chosen, which is sufficient storage to join a 6TiSCH network and provides resilience to periods of low harvesting power. The leakage current is configured at 2 µA at 5 V and the PMU efficiency at 80 % for both the conversion from harvester to the supercapacitor and from supercapacitor to the load. V_{on} and V_{th} are determined using (11) and (12), respectively. E_{join} is set at 0.47 J, which emerged as the average energy needed to join a 6TiSCH network in our previous work, using the same hardware platform [24]. E_p is determined through a preliminary simulation run with a divisor of 1, resulting in an estimated 0.45 J during a prediction interval of 15 min for both routers. This leads to $V_{th} = 2.5$ V and $V_{on} = 3.5$ V. The simulations employed a logistic loss radio medium, which models a non-linear relationship between Received Signal Strength Indicator (RSSI) and PRP, and introduces a random level of Additive White Gaussian Noise (AWGN) to the signal level of each packet. The parameters in Table 4 reflect a typical indoor scenario [35].

A simulation is performed to evaluate the impact of the adaptive scheduling algorithm on both routers. Spanning 3 h and 40 min, the simulation includes changes in harvesting power every 20 min, ranging from 200 µW and 1200 µW, resembling the typical power generated by an indoor solar panel in environments like warehouses or offices with light levels of 200 to 600 lx [2]. Each router experiences a different harvesting power pattern, as illustrated in Fig. 8. The top plots show the capacitor voltage for Router #3 in blue and Router #2 in red, where green vertical lines indicate a change in harvesting power, accompanied by their corresponding values. The divisor for each router is represented by the black lines. The lower plots show the ETX of Node #4 toward its preferred parent, taking the divisor penalty into account, and the CoAP latency from Node #4 to the 6LBR. Blue ETX values indicate Router #3 as preferred parent, while red values represent Router #2. CoAP latency includes the 2-hop latency of the CoAP POST message to the 6LBR, considering not only the MAC latency but also latency due to higher-layer protocols (up to CoAP).

Initially, both routers join the network with minimal energy consumption due to the small network size. Router #3 is chosen as preferred parent by Node #4, as shown on the ETX plot. However, limited harvesting power of 200 µW leads to a steep voltage drop. The algorithm reacts by increasing the divisor to its maximum value of 6, stabilizing the voltage. However, this action increases the ETX to 6 following the reception of a DIO by Node #4. Consequently, Node #4 switches to Router #2, which has a divisor of 1 due to

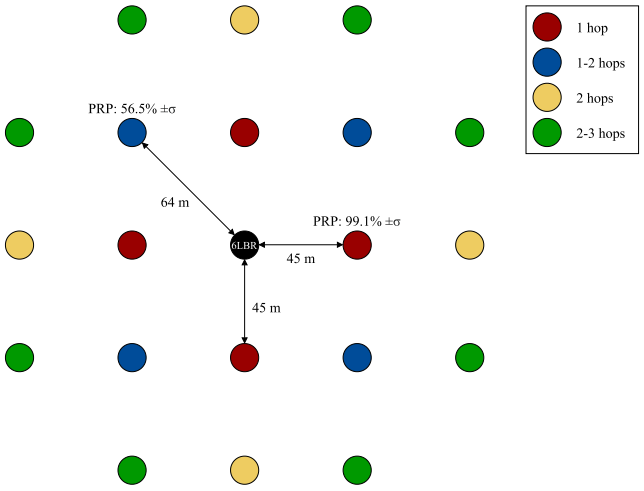


FIGURE 9. Large-scale network topology of battery-less network. Nodes are colored according to their hop distance to the 6LBR, which depends on the PRP.

high harvesting power. Subsequently, the harvesting power of Router #3 increases while decreasing for Router #2, resulting Router #2's divisor to revert to 1 while Router #3's divisor initially increases to 4 before maxing out at 6. Another parent switch is needed to maintain latency below 340 ms. At the end of the simulation, Node #4 switches parent again due to the same phenomenon. This simulation shows the algorithm's ability to adapt the local SF of a battery-less 6TiSCH based on the harvesting power, thereby preventing network disconnection. Additionally, the ETX increase ensures a parent switch to the most favorable router, enhancing multi-hop latency.

C. LARGE-SCALE BATTERY-LESS 6TiSCH NETWORK

To assess the algorithm's performance in a large-scale battery-less network, simulations were conducted using a network of 21 nodes, as illustrated in Fig. 9. As shown, the nodes are arranged in a grid with a distance of 45 m between them, resulting in a PRP of 99.1 % $\pm\,\sigma$. This configuration creates a single hop from the red nodes to the 6LBR. For blue nodes, with a PRP of 56.5 % $\pm\,\sigma$ over a diagonal distance of 64 m, one or two hops are required to reach the 6LBR, depending on the choice of preferred parent. Similarly, yellow nodes require two hops, and green nodes require two or three hops. This setup enables the evaluation of the impact of hop count on the latency to the 6LBR.

As shown in Table 4, each node transmits a non-confirmable CoAP POST message to the 6LBR at 5 min intervals. The harvesting power P_{harv} for each node was randomly selected within the range of [200, 800 µW] and fluctuated randomly by [−100, 100 µW] every 5 min. This aligns with light levels of 200 to 400 lx [2].

Ten simulations, each lasting 9 h, were conducted using 20 battery-less nodes, with and without the algorithm, resulting in a total of 20 simulations, each with a different random seed. For each node, metrics such as successful reporting (the percentage of CoAP POST messages successfully received compared to the expected number), uptime (the percentage of time the node was powered and actively participating in the network), update rate, and latency were calculated. Table 5 presents the average values for these metrics across 20 nodes and 10 simulations, both with and without the algorithm, along with the differences between the two scenarios.

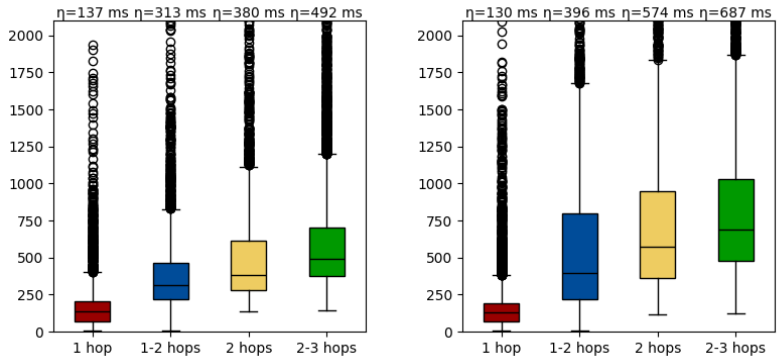
TABLE 5. Results of the simulations with and without the algorithm across 20 nodes and 10 simulations, along with the differences between the two scenarios.

	Without algorithm	With algorithm	Difference
Successful reporting	77.49 %	90.19 %	+12.70 %
Node uptime	78.64 %	91.10 %	+12.46 %
Update rate	430 s	356 s	−17.21 %
Latency	474 ms	627 ms	+32.28 %
Overhead per node	/	87.94 nW	/

The results indicate that the algorithm significantly improves successful reporting and node uptime, as it allows energy consumption to be adjusted based on the current harvesting power. These metrics showed a 12 % increase compared to the scenario without the algorithm, resulting in a successful reporting rate of 90.19 % and a node uptime of 91.10 %. Additionally, the update rate decreased to 356 s, representing a reduction of 17.21 %. The overhead associated with the algorithm, in terms of power consumption due to INT, UDP messages, and additional DIOs (as discussed in Section VII), was minimal, at just 87.94 nW. This, along with the adaptive slotframe size, explains the positive impact of the algorithm. However, a downside of using the algorithm is the increased latency, which rose by 32.28 %, resulting in an average latency of 627 ms, due to the adjusted slotframe size. Nonetheless, this is still good enough for multiple applications, such as building automation [36].

To better understand the impact of hop count on latency, Fig. 10 presents boxplots showing the latency per hop both without (Fig. 10a) and with (Fig. 10b) the algorithm. For clarity, the plots are truncated at 2 s, though outliers up to 5 s exist in both cases. As illustrated, the algorithm has minimal impact on 1-hop latency since the RX slots of the 6LBR remain unchanged. However, for paths requiring more than one hop, latency increases significantly when the algorithm is applied, with the 100th percentile reaching 2 s and the third quartile at 1 s. This increased latency, compared to a smaller four-node network, is attributed to the TSCH back-off mechanism [25], which introduces a random delay after packet collisions. The larger network size leads to more collisions, thereby increasing latency.

Consequently, while the algorithm substantially improves reporting success in battery-less networks, it also results in higher latency. Therefore, the algorithm may not be suitable for critical industrial applications, but it is effective for non-critical applications like building automation.



(a) Without algorithm (b) With algorithm

FIGURE 10. Boxplots of the latency per hop without (10a) and with (10b) algorithm. For visual purposes, the plots are cut off at 2 s, but outliers up to 5 s are present with and without the algorithm.

IX. CONCLUSION

Synchronized multi-hop IWSNs present a promising solution for establishing perpetual battery-less networks by eliminating the need for continuous router listening. However, dynamic adaptation of communication schedules and network topology is essential to accommodate the variability of energy harvesting sources. This paper introduces an adaptive scheduling algorithm for 6TiSCH networks, aiming to optimize communication schedules of battery-less routers dynamically. The algorithm relies on an analytical traffic model and energy model for 6TiSCH devices, allowing the prediction of the energy storage of these devices, which use supercapacitors as storage elements. Since performing energy consumption predictions on constrained nodes is infeasible, a central network manager performs these calculations based on telemetry data provided by all nodes. This approach enables real-time adjustments to the TSCH SF to account for changes in network topology, energy harvesting, and storage levels, ensuring optimal performance in resource-constrained environments.

The evaluation of the adaptive scheduling algorithm reveals its capability to dynamically adapt the local schedule of 6TiSCH routers to their energy constraints when powered by a varying harvesting power resembling an indoor solar panel. The algorithm increases the unicast slotframe size during low harvesting power periods without impacting control traffic. This urges child nodes to choose another router with higher harvesting power or energy availability. In a small-scale topology with four nodes, 2-hop CoAP latency below 340 ms can be achieved. In a large-scale topology with 21 nodes, reporting success and node uptime are increased significantly compared to a battery-less network that does not employ the algorithm. However, this comes at the cost of increased latency. As such, the algorithm paves the way for a sustainable and completely battery-less 6TiSCH network, suitable for non-critical building automation applications.

REFERENCES

[1] X. Vilajosana, T. Watteyne, T. Chang, M. Vucinic, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 595–615, 1st Quart., 2020.

[2] D. Van Leemput, A. Sabovic, K. Hammoud, J. Famaey, S. Pollin, and E. De Poorter, "Energy harvesting for wireless IoT use cases: A generic feasibility model and tradeoff study," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15025–15043, 2023.

[3] A. Sabovic, C. Delgado, D. Subotic, B. Jooris, E. De Poorter, and J. Famaey, "Energy-aware sensing on battery-less LoRaWAN devices with energy harvesting," *Electronics*, vol. 9, no. 6, p. 904, May 2020.

[4] C. Delgado, J. M. Sanz, C. Blondia, and J. Famaey, "Batteryless LoRaWAN communications using energy harvesting: Modeling and characterization," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2694–2711, Feb. 2021.

[5] D. Van Leemput, "Battery-less energy harvesting devices in multi-hop industrial wireless sensor networks," Ph.D. dissertation, Dept. Inf. Technol., Ghent Univ., Ghent, Belgium, 2024.

[6] D. Van Leemput, J. Hoebeke, and E. De Poorter, "Integrating battery-less energy harvesting devices in multi-hop industrial wireless sensor networks," *IEEE Commun. Mag.*, vol. 62, no. 7, pp. 66–73, Jul. 2024.

[7] Z. J. Chew, T. Ruan, and M. Zhu, "Energy savvy network joining strategies for energy harvesting powered TSCH nodes," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1505–1514, Feb. 2021.

[8] K. Das, P. Zand, and P. Havinga, "Industrial wireless monitoring with energy-harvesting devices," *IEEE Internet Comput.*, vol. 21, no. 1, pp. 12–20, Jan. 2017.

[9] D. Van Leemput, J. Hoebeke, and E. De Poorter, "Analytical traffic model of 6TiSCH using real-time in-band telemetry," *Internet Things*, vol. 23, Oct. 2023, Art. no. 100847.

[10] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, Jun. 2022, Art. no. 101089.

[11] A. Karaagac, E. De Poorter, and J. Hoebeke, "Alternate marking-based network telemetry for industrial WSNs," in *Proc. 16th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Apr. 2020, pp. 1–8.

[12] A. Karaagac, E. De Poorter, and J. Hoebeke, "In-band network telemetry in industrial wireless sensor networks," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 517–531, Mar. 2020.

[13] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois, "Monitoring KPIs in synchronized FTDMA multi-hop wireless networks," in *Proc. Wireless Days (WD)*, Mar. 2016, pp. 1–6.

[14] S. Gajica, "Monitoring of 6TiSCH infrastructure with MQTT and zabbix NMS software," in *Proc. Int. Symp. Ind. Electron. Appl. (INDEL)*, Nov. 2020, pp. 1–6.

[15] F. Graf, T. Watteyne, and M. Villnow, "Monitoring performance metrics in low-power wireless systems," *ICT Exp.*, vol. 10, no. 5, pp. 989–1018, Oct. 2024.

[16] H. Hajizadeh, M. Nabi, R. Tavakoli, and K. Goossens, "A scalable and fast model for performance analysis of IEEE 802.15.4 TSCH networks," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–7.
[17] G. Daneels, E. Municio, B. Van De Velde, G. Ergeerts, M. Weyn, S. Latré, and J. Famaey, "Accurate energy consumption modeling of IEEE 802.15.4e TSCH using dual-band OpenMote hardware," *Sensors*, vol. 18, no. 2, p. 437, Feb. 2018.
[18] M. Kubaszek, J. Macheta, Ł. Krzak, and C. Worek, "The analysis of energy consumption in 6TiSCH network nodes working in sub-GHz band," *Int. J. Electron. Telecommun.*, vol. 66, no. 1, pp. 201–210, Jan. 2020.
[19] C. Ouanteur, L. Bouallouche-Medjkoune, and D. Aïssani, "An enhanced analytical model and performance evaluation of the IEEE 802.15.4e TSCH CA," *Wireless Pers. Commun.*, vol. 96, no. 1, pp. 1355–1376, Sep. 2017.
[20] S. Scanzio, M. G. Vakili, G. Cena, C. G. Demartini, B. Montrucchio, A. Valenzano, and C. Zunino, "Wireless sensor networks and TSCH: A compromise between reliability, power consumption, and latency," *IEEE Access*, vol. 8, pp. 167042–167058, 2020.
[21] X. Fafoutis, A. Elsts, G. Oikonomou, R. Piechocki, and I. Craddock, "Adaptive static scheduling in IEEE 802.15.4 TSCH networks," *IEEE 4th World Forum Internet Things (WF-IoT)*, pp. 263–268, Feb. 2018.
[22] S. Scanzio, G. Cena, and A. Valenzano, "Enhanced energy-saving mechanisms in TSCH networks for the IIoT: The PRIL approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7445–7455, Jun. 2022.
[23] A. Hannachi, W. Jaafar, S. Bitam, and N. Ouazene, "A novel energy-efficient cross-layer design for scheduling and routing in 6TiSCH networks," 2024, *arXiv:2403.12949*.
[24] D. Van Leemput, J. Hoebeke, and E. De Poorter, "Supporting ultralow-power nodes in 6TiSCH industrial wireless sensor networks," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38337–38347, Dec. 2024.
[25] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4-2020, Jul. 2020.
[26] P. Thubert, *Objective Function Zero for the Routing Protocol for Low-power and Lossy Networks (RPL)*, document RFC 6552, Internet requests for comments, RFC editor, Mar. 2012.
[27] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document RFC 6719, Internet requests for comments, RFC editor, Sep. 2012.
[28] T. Chang, M. Vucinic, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, *6TiSCH Minimal Scheduling Function (MSF)*, document RFC 9033, May 2021.
[29] S. Duquennoy, 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.*, Nov. 2015, pp. 337–350.
[30] P. Levis, T. H. Clausen, O. Gnawali, J. Hui, and J. Ko, *The Trickle Algorithm*, document RFC 6206, Mar. 2011.
[31] C. Delgado, J. M. Sanz, and J. Famaey, "On the feasibility of battery-less LoRaWAN communications using energy harvesting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.
[32] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 195–213, 1st Quart., 2014.
[33] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, Internet Requests for Comments, RFC Editor, Mar. 2012.
[34] *NRF52840 Product Specification*, document v1.8, Nordic semiconductor, Dec. 2023. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52840PSv1.8.pdf
[35] R. Elsas, E. De Poorter, and J. Hoebeke, "DRiPLOF: An RPL extension for multi-interface wireless sensor networks in interference-prone environments," *Sensors*, vol. 22, no. 10, p. 3906, May 2022.
[36] A. Seferagić, J. Famaey, E. De Poorter, and J. Hoebeke, "Survey on wireless technology trade-offs for the industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 488, Jan. 2020.

DRIES VAN LEEMPUT received the B.Sc. and M.Sc. degrees in electronics and ICT engineering technology and the M.Sc. degree in electrical engineering from Ghent University, Belgium, in 2017, 2018, and 2020, respectively. He is currently pursuing the Ph.D. degree with Ghent University-imec. He joined the Internet Technology and Data Science Laboratory (IDLab) Research Group, Ghent University-imec. His research interests include multi-hop wireless sensor networks in industrial environments, the Internet of Things (IoT), energy harvesting, network and energy modeling, and MAC protocol design for critical wireless systems.

JEROEN FAMAEY (Senior Member, IEEE) received the M.Sc. degree in computer science and the Ph.D. degree in computer science engineering from Ghent University, Belgium, in 2007 and 2012, respectively. He is currently an Associate Professor with the University of Antwerp-imec, Belgium. He is the co-author of over 190 papers published in international peer-reviewed journals and conference proceedings and ten patent applications. His research interests include performance modeling and optimization of wireless connected systems, with a specific interest in the sustainable IoT, and beyond-5G networks.

JEROEN HOEBEKE received the master's degree in engineering computer science from Ghent University, in 2002, and the Ph.D. degree in engineering computer science, in 2007, with his research on adaptive ad hoc routing and virtual private ad hoc networks. He is currently an Associate Professor with the Internet Technology and Data Science Laboratory, Ghent University-imec. He is the author or co-author of more than 200 publications in international journals or conference proceedings. He is conducting and coordinating research on wireless (IoT) connectivity, embedded communication stacks, deterministic wireless communication, and wireless network management.

ELI DE POORTER is currently a Professor with the IDLab Research Group, Ghent University-imec. His team performs research on wireless communication technologies, such as (indoor) localization solutions, the wireless IoT solutions, and machine learning for wireless systems. He has over 200 publications in international journals or in the proceedings of international conferences. He performs both fundamental and applied research. For his fundamental research, he is currently the coordinator of several research projects (SBO, FWO, and GOA). For his applied research, he collaborates with industry partners to transfer research results to industrial applications, and to solve challenging industrial research problems.