Supporting Ultra-Low-Power Nodes in 6TiSCH Industrial Wireless Sensor Networks

Dries Van Leemput, Jeroen Hoebeke, Eli De Poorter

***Abstract*—Industrial wireless sensor networks offer a viable alternative to wired solutions where there is a lack of suitable communication infrastructure. Among these networks, Time Slotted Channel Hopping (TSCH) emerges as a noteworthy choice due to their capacity for deterministic latency, heightened reliability, and low power consumption. Nonetheless, challenges arise from the energy-intensive joining procedure and inherent idle listening associated with TSCH, impeding the integration of battery-powered end devices. Therefore, this paper introduces a novel 6TiSCH Low-Power Node (6LPN) that supports ultra-low-power operations. The proposed solution optimizes the energy-intensive joining procedure through reduced advertisement channels, optimal scanning time, and a delayed join, achieving a 90% reduction in energy consumption. Furthermore, idle listening is eliminated by queueing downlink traffic in a 6TiSCH Friend Node (6FN), ensuring an 87-94% reduction in power consumption during operational mode while maintaining an average latency of queued frames of 7.62 seconds. By comparing the impact of the optimizations on three IoT hardware platforms, we demonstrate that optimal results are obtained for devices with low RX and idle transceiver current. Finally, our solution maintains full backward compatibility with 6TiSCH and does not introduce additional control traffic.**

***Index Terms*—Internet of Things (IoT), Industrial Wireless Sensor Networks (IWNSs), Energy Efficiency, 6TiSCH, Time Slotted Channel Hopping (TSCH).**

NOMENCLATURE

6LPN	6TiSCH Low-Power Node
6FN	6TiSCH Friend Node
6LN	6LoWPAN Node
6LR	6LoWPAN Router
6LBR	6LoWPAN Border Router
KA	Keep Alive
EB	Enhanced Beacon
EACK	Enhanced Acknowledgement
DIS	DODAG Information Solicitation
DIO	DODAG Information Object
DAO	Destination Advertisement Object
TSN	Time Source Neighbor
LPM	Low-Power Mode

I. INTRODUCTION

Industrial Wireless Sensor Networks (IWSNs) have emerged as a viable alternative to wired solutions in industrial environments. Compared to their wired equivalents, Wireless Sensor Networks (WSNs) offer better flexibility, scalability, and cost-effectiveness. These characteristics make IWSNs especially attractive for retrofitting industrial plants, for deployments in harsh conditions where cable failures are common, or in dynamic manufacturing facilities where machinery and equipment frequently change locations or are often reconfigured. In all of these situations, IWSNs offer cost benefits by removing the time-consuming and expensive need for frequent rewiring.

Among IWSN technologies designed for industrial contexts, Time Slotted Channel Hopping (TSCH) has gained popularity for its deterministic latency, high reliability, and low power consumption through reduced duty cycles. The Internet Engineering Task Force (IETF) IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) protocol further enhances TSCH-based networks by offering IPv6 connectivity on top of TSCH. This includes the IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) adaptation layer, in addition to the Routing Protocol for Low-power and Lossy Networks (RPL) and Constrained Application Protocol (CoAP). Alternatives for TSCH-based IWSNs include ANSI/ISA 100.11a and IEC WirelessHART. The low-power nature of TSCH-based devices allows for battery-powered operation, increasing flexibility and scalability.

Despite these advantages, battery-powered devices within TSCH-based networks often need to maintain an operational lifespan of more than ten years to mitigate economic and environmental implications associated with frequent battery replacements. Moreover, the challenge of replacing batteries in remote or inaccessible locations presents dangers, high costs, or even impossibilities. Additionally, the hazardous nature of batteries restricts their deployment in strictly regulated areas like potentially explosive environments, requiring using costly ATEX-certified batteries. Introducing battery-less devices powered by energy harvesters could resolve these issues, but it requires even greater energy efficiency from IWSNs. As such, despite the energy efficiency of TSCH-based networks, challenges related to their energy-intensive joining procedure and idle listening must be addressed to provide ultra-low-power nodes and ensure either an acceptable battery lifetime or battery-less operation. Simplifying the energy-intensive joining procedure holds particular importance for battery-less devices, which typically rely on supercapacitors with lower energy density compared to batteries. In addition to the variability of energy harvesters, this leads to an intermittent behavior requiring nodes to join the network multiple times [1]. Furthermore, eliminating idle listening is beneficial not only for battery-less devices but also for extending the battery life of their counterparts powered by traditional batteries.

D. Van Leemput, J. Hoebeke, and E. De Poorter are with IDLab, Department of Information Technology, Ghent University - imec, Belgium e-mail: firstname.lastname@ugent.be.

Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX.

Several previous works already studied the joining procedure of 6TiSCH and proposed optimizations [2]–[6], while others aimed to reduce idle listening during operational mode [7]–[10]. However, none of these works both optimize the joining procedure and eliminate idle listening, without compromising the energy efficiency of routers. To achieve this, we introduce two 6TiSCH device types: the 6TiSCH Low-Power Node (6LPN) and the 6TiSCH Friend Node (6FN). The 6LPN enables the integration of ultra-low-power or battery-less devices into 6TiSCH networks, while the 6FN serves as a specialized variant of a 6LoWPAN Router (6LR), queuing downlink traffic for 6LPNs to eliminate idle listening. Furthermore, 6LPNs and 6FNs can be integrated into existing 6TiSCH networks without the need for modifications to other devices. 6LPNs only communicate through 6FNs, which adhere to the 6TiSCH standard specification when communicating with other devices. As such, our solution is fully backward compatible with the 6TiSCH standard while introducing no additional control traffic.

To summarize, our main contributions are:

- We introduce two 6TiSCH device types: the 6LPN and the 6FN. The 6LPN is an ultra-low-power device and connects to a 6FN, which queues downlink traffic for 6LPNs. Both device types are compatible with 6TiSCH devices, acting as 6LoWPAN Node (6LN), 6LR, or 6LoWPAN Border Router (6LBR).
- We present an optimized joining and friendship establishment procedure for 6LPNs without introducing additional control traffic. This procedure includes the use of limited advertisement channels, an optimal scanning duration, and a delayed join.
- We define an ultra-low-power operational mode for 6LPNs, which involves the complete elimination of idle listening. In this mode, 6LPNs activate their radio only when necessary, without the requirement of an always-on 6FN, ensuring maximum energy efficiency for both the 6LPN and the 6FN.

In what follows, Section II discusses related works, technical background on 6TiSCH is provided in Section III, and the operation and evaluation of 6LPN and 6FN are detailed in Sections IV and V, respectively. Finally, Section Section VI concludes this paper.

II. Related Work

Table I shows an overview of existing literature covering energy consumption optimization in TSCH-based networks, such as IEEE 802.15.4e and ISA100.11a. These optimizations focus on the joining procedure and/or the operational mode of the network. In addition, several works integrate low-power devices into TSCH-based networks and other Internet of Things (IoT) standards. This related work section first outlines TSCH-based works and then addresses low-power device types in other standards.

A. Supporting Low-Power Nodes in TSCH-based Networks

Optimizing the joining procedure in 6TiSCH networks to reduce the network joining time and energy consumption has been the topic of several research works. Kalita et al. [2] provide a comprehensive review of both TSCH and 6TiSCH network formation optimizations, including theoretical analysis and testbed experiments. Tanaka et al. [3] introduce SF-Fastboot, a scheduling function designed to speed up overall 6TiSCH network formation. By reserving a specific channel offset for beacon transmission, the initial synchronization time is reduced. In addition, congestion arising from multiple join requests from various nodes is mitigated by assigning dedicated slots for this channel offset, which further reduces the total network formation time. Karalis et al. [4] explore the impact of the scan duration on the initial synchronization time and energy consumption. The authors used mathematical analysis, an algorithmic approach, and empirical results to determine the optimal scan period. They found this to be C slotframes, where C represents the number of available channels. This resulted in an improvement of $48.37\,\%$ and $47.10\,\%$ in terms of average initial synchronization time, compared to the default scan periods of Contiki-NG and OpenWSN, respectively. Chew et al. [5] address energy-efficient network joining strategies for nodes powered by energy harvesting. A duty-cycled network joining is proposed, where nodes join the network during gap times of other joining nodes to reduce energy wastage of the nodes in waiting. To reduce the initial scanning time, limited advertisement channels are used. This approach is especially beneficial when multiple nodes try to join the network simultaneously. Das et al. [6] present a hierarchical network management approach. A system manager manages the initial resources, and the remaining resources are allocated to local routers for their management. Routers use their resources to handle resource-constrained end devices to ensure rapid joining within the network. In addition, the scanning time is minimized by reducing the number of advertisement channels. As shown in Table I, these works optimize the joining procedure of TSCH-based networks in terms of scanning and/or network joining. Our work combines the most effective existing techniques to reduce scanning time, alongside introducing an additional technique, which involves delaying network joining of ultra-low-power nodes to further reduce scanning time. Furthermore, network joining (i.e., after TSCH synchronization) is also optimized and adapted to integrate a friendship establishment procedure while reusing existing control frames, thereby remaining compliant with the minimal 6TiSCH configuration [13]. Moreover, the operational mode of TSCH was left unexplored in the above works, meaning idle listening during empty RX slots still exists, increasing the energy consumption during the operational mode.

To that end, other works try to limit idle listening in TSCH to conserve energy in the operational mode of TSCH. Fafoutis et al. [7] propose an Adaptive Static Scheduling for TSCH networks, which builds on top of existing static schedules by adaptively activating a subset of allocated timeslots. This approach reduces the overhead of idle listening in unused slots, particularly beneficial for networks with dynamic traffic patterns. Scanzio et al. [8] propose Proactive Reduction of Idle Listening (PRIL), which allows skipping RX slots where idle listening is anticipated. To this end, the authors introduce a

TABLE I: Comparison of our work and related works covering (i) energy consumption optimization in TSCH-based networks and (ii) low-power device types in these networks and other standards. The comparison includes TSCH joining optimizations, synchronization, elimination of idle listening, downlink queueing, and constrained routers.

MAC standard	Work	Joining optimization		Synchronization	Idle listening elimination	Downlink queueing	Constrained routers
		Scanning	Network joining				
IEEE 802.15.4e	6TiSCH formation review [2]	✓	✓	✓	✗	✗	✓
	SF-Fastboot [3]	✓	✓	✓	✗	✗	✓
	Optimal scanning time [4]	✓	✗	✓	✗	✗	✓
	Duty-cycled joining [5]	✓	✓	✓	✗	✗	✓
	Adaptive Static Scheduling [7]	✗	✗	✓	Adaptive	✗	✓
	PRIL [8]–[10]	✗	✗	✓	Reduced	✓	✓
ISA100.11a	Hierarchical management [6]	✓	✓	✓	✗	✗	✓
	Asynchronous ISA100.11a [6]	/		✗	✓	✗	✗
IEEE 802.15.1	BLE LPN [11]	/		✗	✗	✓	✗
IEEE 802.15.4	Thread SED [12]	/		✗	✗	✓	✗
	Thread SSED [12]	/		✓	✗	✓	✗
IEEE 802.15.4e	Our work - 6LPN	✓	✓	✓	✓	✓	✓

new Medium Access Control (MAC) sleep command, which temporarily suspends listening in specified cells. The authors extend this work by exploring various approaches for the sleep command [9] and by evaluating several PRIL strategies, considering ACK frame losses [10]. While these works either reduce idle listening or adapt it to dynamic network requirements, our work significantly outperforms prior work by using packet buffering, allowing our approach to completely eliminate idle listening for ultra-low-power devices. This in addition to optimizing the joining procedure, as shown in Table I.

Another approach to completely remove joining, synchronization, and idle listening was proposed by the authors in [6] using an asynchronous communication scheme. Routers use separate sections of the superframe for scheduled communications and asynchronous data publications, using slow hopping. As such, nearby routers need to coordinate closely to ensure at least one of them is listening all the time or one router needs to be always on. However, in contrast to their approach, our work is fully TSCH compatible and we can still use the existing TSCH schedule because low-power end devices synchronize periodically. In addition, in our approach also downlink traffic is supported by queueing in the 6FN.

B. Low-Power Device Types in Other Standards

A number of low-power wireless communication technologies already support low-power device types, which communicate with a parent or friend node. For instance, the Bluetooth Low Energy (BLE) specification [11] offers a Low Power Node (LPN) and a Friend Node (FN) device type. The FN queues downlink traffic for the LPN, allowing the LPN to remain in a low-power state, periodically polling for downlink messages. It can either go back to sleep or wake up after a chosen time interval to retrieve the message. Similarly, the Thread specification [12] defines a Sleepy End Device (SED) that periodically polls for downlink traffic from a parent router. Furthermore, the Thread specification defines a Synchronized SED (SSED) that employs Coordinated Sample Listening (CSL) [14] to periodically wake up and listen for downlink traffic without the need to transmit a polling message. The main difference of our work is that routers (6FNs) do not require to be always on. As such, their power consumption remains unaffected, enabling constrained routers.

To summarize, our work presents, to the best of our knowledge, the first implementation of an ultra-low-power device type for a low-power wireless communication technology that uses synchronized communication without idle listening and can connect to an energy-constrained router.

III. Technical Background

The IETF 6TiSCH working group proposed the 6TiSCH protocol stack to provide reliable and deterministic behavior for Industrial IoT (IIoT) networks. It is based on the TSCH MAC layer and supports IPv6 through the 6LoWPAN adaptation layer. TSCH was introduced in the IEEE 802.15.4 standard [14] as a MAC protocol targeted at low-rate wireless networks featuring high reliability, low power consumption, and deterministic latency. Time is divided into timeslots, which can either be used exclusively by two nodes (dedicated slots) or shared between multiple nodes (shared slots). In the latter case, nodes perform Clear Channel Assessment (CCA) before accessing the channel. A timeslot allows for the transmission of a frame and, in the case of unicast transmission, an Enhanced Acknowledgement (EACK) from the receiver. Both TSCH frames as well as EACKs carry Information Elements (IEs) that are used to distribute information between communicating nodes. Timeslots are grouped into one or multiple repeating slotframes, where the Absolute Slot Number (ASN) keeps track of the total number of timeslots that have passed since the start of the network. Furthermore, TSCH employs channel hopping, resulting in a different frequency channel being used for each timeslot in consecutive slotframes. Therefore, a TSCH link is defined as the pairwise assignment of a directed communication between nodes in a given timeslot on a given channel offset [14]. To determine on which channel to communicate for an ASN and channel offset CO, (1) is used, where CH is the frequency channel to communicate on and HS the hopping sequence containing the available channels.

$$CH = HS[(ASN + CO)\%|HS|] \tag{1}$$

To assign TSCH links to different (pairs of) nodes, a Scheduling Function (SF) is required that assigns both a slot offset and channel offset. IEEE 802.15.4 does not dictate a SF and leaves it up to the implementer to choose or develop a suitable SF. The most widely used SFs are the 6TiSCH Minimal Scheduling Function (MSF) [15] and Orchestra [16]. TSCH nodes synchronize to a Time Source Neighbor (TSN) by acknowledgement-based or frame-based synchronization. In the former method, the TSN includes the synchronization error in the Sync IE of EACKs, which are sent in response to unicast frames. When there are no unicast frames scheduled to the TSN, zero-payload Keep Alive (KA) messages are sent to trigger an EACK. When using frame-based synchronization, TSCH nodes synchronize based on the expected and actual reception time of frames sent by their TSN. As such, TSCH nodes periodically transmit Enhanced Beacons (EBs) to synchronize, but also to advertise the network and distribute the SF details. These EBs are transmitted during timeslots that are marked as 'advertising' timeslots, which can be dedicated (each node has a dedicated timeslot to transmit EBs) or shared. The minimal 6TiSCH configuration [13] describes a minimal set of rules to operate a TSCH network and mandates a shared advertising timeslot with time offset and channel offset 0 as a fall-back mode or in case there is no SF available.

RPL [17] is the default routing protocol in the 6TiSCH network stack and is a distance vector routing protocol that forms a Destination Oriented Directed Acyclic Graph (DODAG) converging at the RPL root. Communication is possible by routing via the root (non-storing mode) or by maintaining local routing tables (storing mode). It uses DODAG Information Object (DIO) messages to form the DODAG and enable uplink routing. DIOs are either sent multicast by each router using a Trickle timer [18], or unicast in response to a multicast DODAG Information Solicitation (DIS) message. The latter is sent by a node wanting to join the network and solicit DIOs to speed up the joining procedure. After receiving a DIO, the node can join the RPL instance and DODAG by selecting a preferred parent. Subsequently, the node transmits a unicast Destination Advertisement Object (DAO) for downlink routing to its parent(s) (storing mode) or the RPL root (non-storing mode), which may optionally reply with a DAO-ACK.

IV. Low-Power Node for 6TiSCH

This section describes the details of the 6LPN and 6FN device types and how they can enable the integration of extremely low-power devices into 6TiSCH networks. Despite TSCH being a low-power MAC protocol, nodes frequently wake up to receive incoming frames at predefined timeslots. This periodic wake-up introduces energy loss due to "idle listening", when nodes activate their radio to listen to incoming traffic but no frames are received during those slots. Choosing an appropriate slotframe size becomes a crucial trade-off between network performance (latency, throughput, and synchronization accuracy) and power consumption. Increasing the slotframe size to reduce idle listening for some low-power nodes can negatively impact the performance of other nodes with sufficient energy resources, such as mains-powered nodes. Moreover, idle listening occurs in idle RX slots, while no energy is lost during idle TX slots, as the node does not need to listen. As such, increasing the slotframe size to reduce idle listening (i.e., the number of RX slots) also reduces the number of TX slots, which degrades performance without enhancing energy consumption. Negotiating RX slots dynamically based on traffic demand, such as negotiated slots in 6TiSCH MSF, solves this issue partially. However, this negotiating these slots introduces additional control traffic, which may not be feasible for low-power nodes.

Therefore, we suggest queuing downlink traffic for a 6LPN in a 6FN, to which the 6LPN remains synchronized. The 6LPN can still transmit uplink data using TX slots but skips all scheduled RX slots unless actual frames are queued at the 6FN, at which point it listens to the incoming downlink traffic. The 6LPN acts as a RPL leaf node only, which allows the friendship feature to work. In addition, this reduces the required control traffic since the transmission of EBs and DIOs is not mandatory for 6TiSCH leaf nodes [14], [17]. The remainder of this section first describes several assumptions for the 6TiSCH network, before detailing the operational mode, and the joining and friendship establishment.

A. Assumptions

For the Low Power Node option to connect to a 6TiSCH network, several assumptions need to be fulfilled.

A 1 (Known advertisement channels). A 6LPN knows which channels are contained in the hopping sequence used for advertisement (to transmit EBs). Therefore, it only needs to scan on those channels contained in the hopping sequence.

A 2 (Dedicated RX slot). Each node can be reached on its dedicated unicast RX slot, during which it listens for incoming traffic. The timeslot handle for this timeslot is calculated using a hash function of the link-layer address of the receiving node. As a result, all nodes can be reached if their link-layer address is known. This functionality covers the most common TSCH schedulers, such as 6TiSCH MSF [15] (autonomous cells) and Orchestra [16] (receiver-based unicast slotframe).

A 3 (RPL joined). We assume a node has fully joined the network when transmitting a DAO and completing the RPL join process, after which the operational mode starts. Nonetheless, other mechanisms such as 6LoWPAN Neighbor Discovery (ND), Duplicate Address Detection (DAD), and Constrained Join Protocol (CoJP) can still be executed during the operational mode without loss of generality. Nonetheless, these mechanisms are optional and implementation-specific as they are not mandatory in the minimal 6TiSCH configuration [13].

B. Operational Mode

In the operational mode, the network is in stable operation and the 6LPN is connected to its 6FN, which serves both as TSN and preferred parent. The main sources of energy loss during operational mode are idle listening [7]–[10] and periodic synchronization [1]. We will discuss both sources separately.

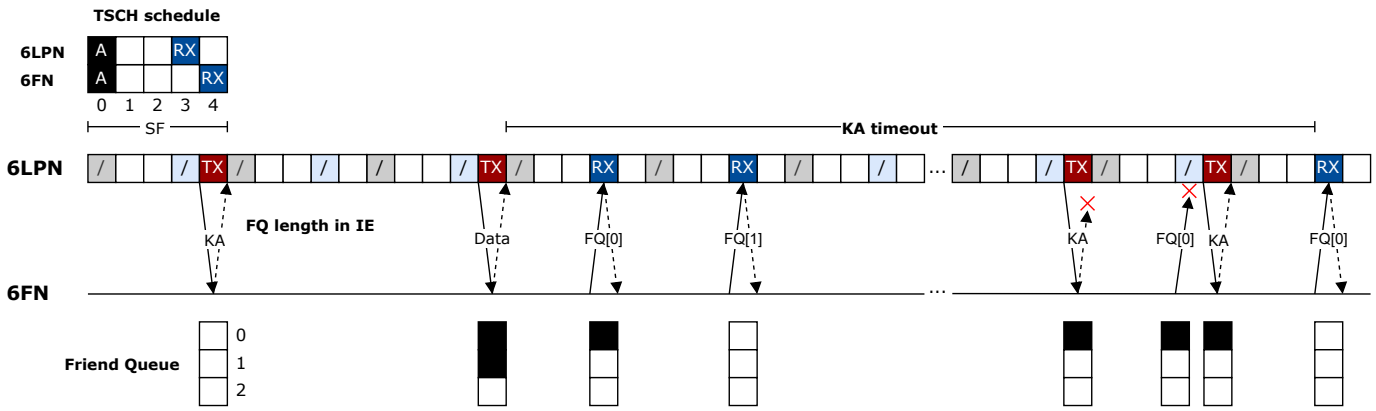


Fig. 1: Time schedule of a 6LPN-6FN interaction. The TSCH schedule (top left) contains a single advertisement slot and one dedicated RX slot for each node. Blue and red dots indicate RX and TX mode, respectively, while tinted slots indicate skipped slots. The 6LPN synchronizes periodically using KAs and the 6FN returns the Friend Queue length in EACKs. If non-empty, the 6LPN wakes up during the next RX slot to receive queued frames.

1) Idle listening: To eliminate idle listening entirely, (i) shared and advertisement slots must be removed and (ii) the dedicated RX slot only activated when there is an actual pending frame. When **shared and advertisement slots are removed**, an alternative method must be available to receive and transmit EBs and multicast DIOs. Given that the 6LPN is configured as leaf mode only, it is not required to transmit DIOs and EB. Furthermore, the 6LPN uses acknowledgment-based synchronization instead of listening for EBs in advertisement slots, which eliminates idle listening related to synchronization. KAs are transmitted before the KA timeout expires in the dedicated TX slot to the 6FN, corresponding to the 6FN's dedicated RX slot. For conciseness, we will call this the TX slot. Periodic multicast DIOs are duplicated by the 6FN and transmitted as unicast DIOs to the 6LPN, allowing the removal of shared and advertisement slots while maintaining periodic synchronization and DODAG updates. To **remove idle listening during dedicated RX slots**, the 6FN maintains a Friend Queue for downlink traffic directed to the 6LPN. Consequently, the 6LPN does not need to wake up for each scheduled RX slot. The 6FN conveys the current Friend Queue length to the 6LPN using an additional IE in the EACKs sent in response to a KA or a data frame. Therefore, KAs are not only used to periodically synchronize but also to poll the 6FN for a Friend Queue status update. Following this update, the 6LPN wakes up for the next $|FQ|$ scheduled RX slots, where FQ indicates the Friend Queue length. As a result, idle listening can be completely removed for the 6LPN, significantly reducing power consumption. When the 6LPN needs to transmit uplink data, it uses the next available TX slot to the 6FN. Since the 6FN replies with an EACK, the data frame serves as both a polling and synchronization message, resetting the KA timeout accordingly. Interactions with the 6FN and neighboring 6TiSCH devices follow the regularly employed SF without any modifications, resulting in a seamless integration with a standard-compliant 6TiSCH network.

Figure 1 shows an example time schedule of a 6LPN and the interaction with a 6FN. The top left corner shows a simplified TSCH schedule, containing a single advertisement slot at slot offset 0 and one dedicated RX slot for each node (*A2*). The shared advertisement slot, shown in black, is used for both advertisement (EBs) and multicast control traffic (e.g., DIOs). The frequency space is not shown as it does not impact the operation. In the 6LPN time schedule, blue and red slots indicate the radio is activated in RX or TX mode, respectively. Tinted slots indicate skipped slots while white slots represent idle slots. As can be seen, the 6LPN synchronizes periodically with the 6FN using KAs and the 6FN passes the Friend Queue length through an IE encapsulated in returning EACKs. At a certain point, the 6LPN sends uplink data to the 6FN and the Friend Queue contains two frames. Consequently, the 6LPN wakes up in the two next occurring RX slots and skips all consecutive timeslots. Before the KA timeout expires, the 6LPN attempts synchronization with the 6FN but the EACK fails, leading to unsuccessful synchronization and polling. However, in the event of an EACK failure, TSCH frames are retransmitted up to a predefined limit. Therefore, the 6LPN retransmits the KA during the next available TX slot. In the example of Fig. 1, the EACK is successfully delivered, causing the 6LPN to wake up in the next RX slot. As a result, a failed EACK does not automatically result in the queued downlink messages being deferred to the next KA timeout. Furthermore, as retransmissions are retained, reliability is not compromised compared to a regular TSCH link.

2) Periodic Synchronization: To minimize power consumption, the KA timeout should be chosen as large as possible. However, the KA timeout depends on clock drift and $GuardTime$ window, during which a receiver listens for incoming frames. For instance, a $32.768\,\mathrm{kHz}$ crystal with a $30\,\mathrm{ppm}$ drifts at most $30\,\mu\mathrm{s}$ per second, resulting in a maximum drift of $60\,\mu\mathrm{s}$ between two nodes employing such a crystal. Assuming a $GuardTime$ of $1\,\mathrm{ms}$, nodes should synchronize every $16\,\mathrm{s}$. Increasing the KA timeout, therefore, requires either a higher precision crystal or a longer $GuardTime$ window. Higher precision crystals are not always readily available and generally consume more power. As such, it is more favorable to increase the KA timeout by extending the $GuardTime$. Although an increased $GuardTime$ also affects energy consumption in received slots, this impact is

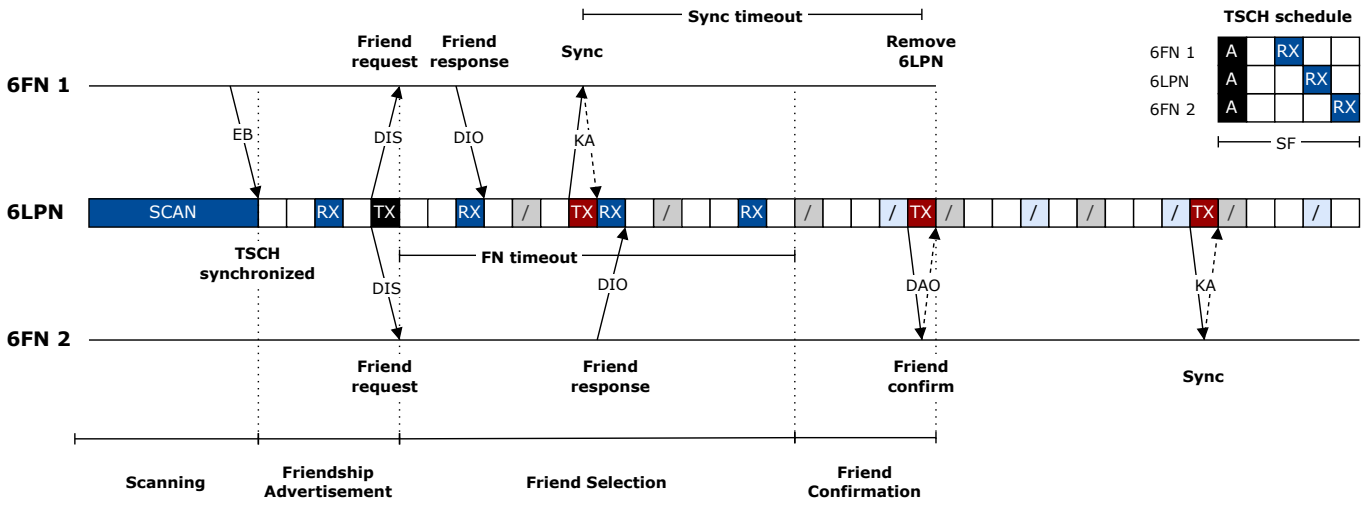


Fig. 2: 6LPN time schedule and exchanged messages during joining and friendship establishment, comprising four phases: (i) Scanning, (ii), Friendship Advertisement, (iii) Friend Selection, and (iv) Friend Confirmation. Existing RPL messages are reused as Friend request, response, and confirmation. After initially synchronizing upon receiving an EB from a 6FN, the 6LPN advertises through a multicast Friend request. As nearby 6FNs respond with a Friend response, the 6LPN chooses the best suitable 6FN and confirms this friendship, after which the operational mode begins.

negligible for a 6LPN because RX slots are rare due to the removal of idle listening. However, increasing the KA timeout comes at the cost of an increased downlink latency since downlink traffic is queued at the 6FN. As such, downlink latency is upper-bounded by the KA timeout. Therefore, choosing the KA timeout results in a trade-off between downlink latency and power consumption.

C. Joining and Friendship Establishment

During joining, 6LPNs synchronize to the TSCH schedule, join the RPL instance, and choose a suitable 6FN to pair with. As per *A3*, we assume nodes to be fully joined to the 6TiSCH network after transmitting a DAO. Friendship establishment occurs simultaneously with joining, using existing RPL control traffic. This avoids introducing additional control traffic, thereby reducing energy consumption. Figure 2 depicts an example of joining and friendship establishment, showing the TSCH time schedule of a 6LPN and the exchanged messages between two 6FNs. The TSCH schedule of Figure 1 is reused and extended with a dedicated RX slot for 6FN1 at slot offset 2. As shown in Figure 2, joining and friendship establishment is divided into four phases: (i) Scanning, (ii) Friendship Advertisement, (iii) Friend Selection, and (iv) Friend Confirmation, which we will discuss chronologically.

1) Scanning: Scanning is by far the most energy-intensive aspect of the joining process [1], [4], [5]. During this phase, a pledge[1] systematically hops across all channels in the hopping sequence until it receives an EB from a nearby node operating on the same channel it is currently listening on. Due to the channel-hopping feature, the pledge consumes the most energy during this scanning phase as it is constantly in receiving mode. Therefore, we limit the scanning phase using three techniques. First, the pledge scans a channel for the **optimal scanning duration**, as defined in Karalis et al. [4], before moving on to a different channel. This optimal scanning duration is calculated as $T_{sf} * N_{ch}$, where T_{sf} is the (advertising) slotframe length and N_{ch} is the number of different advertisement channels in the hopping sequence. This requires a pledge to have prior knowledge of both T_{sf} and N_{ch}, hence they must be predetermined before network deployment. If these parameters are unknown to pledges, a slightly modified scanning duration can be employed, as detailed in [19]. However, in this work, we assume both T_{sf} and N_{ch} to be predetermined before network deployment. Second, the **number of advertisement channels is reduced** while retaining the same number of channels for other communication [1], [5], [6], [20], which has been shown to positively impact scanning time. This is achieved by defining an advertisement hopping sequence next to the original hopping sequence. The third involves **delaying 6LPNs to join the network** until all other nodes have joined the network and are advertising EBs. After all, 6LPNs operate as leaf nodes without contributing to the mesh as router. Moreover, they would spend a portion of their scanning time scanning for EBs, which are not yet being transmitted. This requires a manual setup procedure in network deployment, as 6LPNs must only be activated once all devices in the network are broadcasting EBs. Employing these techniques significantly reduces the scanning time before a valid EB is received. Moreover, these techniques can be employed while meeting the requirements of the minimal 6TiSCH configuration [13]. This moment concludes the scanning phase and the 6LPN selects the router from which it received an EB as TSN (6FN1 in Figure 2).

2) Friendship Advertisement: Upon synchronizing to the TSCH schedule, the 6LPN transmits a multicast Friend Request in the shared slot, announcing its presence to other nearby 6FNs. To avoid introducing extra control traffic, a DIS message is modified by a single unused bit in the reserved

[1]A pledge is a node that attempts to join a network, but has not yet completed the join process.

field to indicate its dual purpose as Friend Request. As such, nearby routers that do not implement 6FN functionality will ignore this DIS. Following the DIS transmission, the 6LPN removes the shared slot from its schedule to start reducing idle listening. After all, nearby routers reply with a unicast DIO in response to the multicast DIS, for which they can use the dedicated RX slot of the 6LPN (*A2*). After transmitting the DIS, the 6LPN starts a FN timeout, which starts the Friend Selection phase.

3) Friend Selection: During Friend Selection, all nearby 6FNs who receive the Friendship Advertisement (multicast DIS) reply with a unicast DIO as response. To indicate its dual usage as a Friend Response message, the DIO is modified in the same manner as the DIS/Friend Request message, using a single unused bit in the Reserved field. During Friend Selection, the 6LPN performs parent selection each time it receives a new DIO/Friend Response, changing its preferred parent and TSN based on the link metric associated with that 6FN. This parent selection procedure deviates from the traditional parent selection in RPL, where the path cost to each candidate parent is evaluated. This path cost includes both the rank of the candidate parent, reflecting its distance to the root, and the link metric to that candidate parent. The reason we deviate from this default behavior is that the reliability of the 6LPN-6FN link is more important than the distance to the RPL root. After all, the 6LPN will be tied to the 6FN for the remainder of the operational mode and communicate exclusively with the 6FN. Furthermore, it will not receive DIOs from other suitable 6FNs and can, therefore, not switch preferred parent, unlike the traditional RPL behavior. The Friend Selection phase ends when the FN timeout has elapsed.

4) Friend Confirmation: The 6LPN confirms its friendship with the best-suited 6FN by sending a unicast DAO. Other 6FNs that were previously selected as preferred parent and TSN, remove the 6LPN after the KA timeout. At this point, the 6LPN joined the RPL DODAG and can send and receive traffic to and from the RPL root through its preferred parent, being the 6FN. As joining is complete and there is no need to receive frames other than from the 6FN, the 6LPN removes all unicast RX slots to eliminate idle listening completely.

V. Evaluation

We introduced the novel 6LPN and 6FN device types in Contiki-NG and released the corresponding open-source code online. Unfortunately, the Contiki-NG software does not support the use of low-power Sleep Timers (STs) but mandates a General Purpose Timer (GPT) when running TSCH for any of its platforms, even though several supported platforms include the hardware capabilities for this. As such, when evaluating the code on hardware platforms, a GPT is used even in Low-Power Mode (LPM), which causes the LPM current consumption to increase significantly (mA instead of $\mu\mathrm{A}$). Therefore, to evaluate the performance of a 6LPN, we used the Cooja simulator within Contiki-NG instead. This still requires a full implementation of the introduced device types and allows for the emulation of real hardware platforms. Energy consumption was analyzed using the Energest module of Contiki-NG, which accurately tracks the time spent in each Central Processing Unit (CPU) and radio state for the chosen emulated hardware platform. This allows us to use current consumption values corresponding to LPMs using a ST and gain insights into the CPU and radio state timings for 6LPNs. Table II lists the employed simulation parameters. To ensure a realistic environment, the Logistic loss radio medium was used, employing a non-linear relationship between the signal level and Packet Reception Ratio (PRR), adding a random noise level (Additive White Gaussian Noise (AWGN)) to the signal level of each frame. The main parameters for the configuration of the logistic loss function are given in Table II. The Orchestra TSCH-RB-397-31-17 SF was used, which features an EB slotframe of length 397, a broadcast slotframe of length 31, and a receiver-based unicast slotframe of length 17, fulfilling A 2. For more information on the detailed of the SF, the reader is referred to [16]. An EB interval of $16\,\mathrm{s}$ was chosen as it is the default value in Contiki-NG and provides a good balance between network joining time and energy consumption of joined nodes. The Friend Queue size was chosen as eight, identical to the neighbor queue length in Contiki-NG.

TABLE II: Cooja (top) and Contiki-NG (bottom) simulation parameters.

Parameter	**Value**
Cooja radio medium	Logistic loss
TX range	$120\,\mathrm{m}$
RX sensitivity	$-100\,\mathrm{dBm}$
$RSSI_{50\,\%}$	$-92\,\mathrm{dBm}$
Path loss exp.	3
σ_{AWGN}	3
Contiki-NG version	Release-4.8
TSCH Scheduling Function	Orchestra TSCH-RB-397-31-17
EB interval	$16\,\mathrm{s}$
Friend Queue size	8
Max retransmissions	8

Energest defines two CPU modes for TSCH[2]. Firstly, LPM requires a $32\,\mathrm{kHz}$ ST to allow the device to wake up at the next scheduled active slot with sufficient accuracy. Secondly, an Active mode requires a $16\,\mathrm{MHz}$ GPT for a more precise timeslot timing. As for the radio, Receiving (RX) and Transmitting (TX) states are defined. Table III lists the current consumption at $3\,\mathrm{V}$ of the Energest states for three available platforms in Contiki-NG, according to their data sheets. For the TX state, we assumed a TX power of $0\,\mathrm{dBm}$. The three platforms represent an evolution in the available hardware for TSCH and Contiki-NG, with the Zolertia Z1 [21] introduced in 2009, the Texas Instruments (TI) cc2538 [22] in 2013, and the Nordic Semiconductor nRF52840 [23] in 2018.

In our simulations, we used the Z1 platform for all 6TiSCH device types to simulate Energest state durations, as it is the only real-life platform supported in Cooja. The same Energest state durations were applied for the cc2538 and nRF52840 platforms, allowing to analyze the impact of variations in CPU and radio state values between platforms on 6LPN performance.

[2]A third, LPM2 mode is also defined. However, it is never used in the available platforms that can be simulated in Cooja.

TABLE III: Current consumption of the Energest states for the Z1, cc2538, and nRF52840 platforms, according to their data-sheets [21]–[23]. Contiki-NG requires the LPM to have a ≥32 kHz ST and the active mode a ≥16 MHz GPT.

Energest state		Z1	cc2538	nRF52840
CPU	LPM (≥32 kHz ST)	20.45 µA	1.3 µA	3.16 µA
	Active (≥16 MHz GPT)	10 mA	13 mA	6.3 mA
Radio	RX	18.8 mA	24 mA	6.53 mA
	TX (0 dBm)	17.4 mA	24 mA	6.4 mA

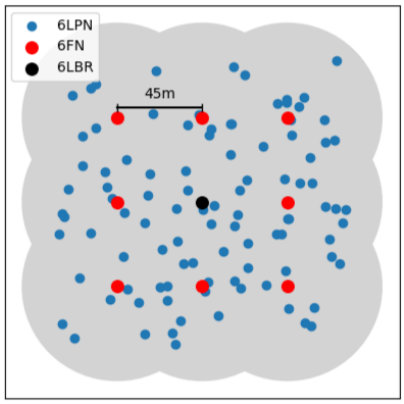


Fig. 3: Network topology to simulate joining and friendship establishment. The network consists of one 6LBR (black) and eight 6FNs (red) with their communication range (grey cloud). Blue dots represent all 100 locations where a 6LPN joined the network.

We will first analyze the impact of the techniques to reduce the scanning time on the energy consumption during joining and friendship establishment. Next, we evaluate the power consumption versus latency trade-off during the operational mode.

A. Joining and Friendship Establishment

To evaluate the energy consumption during joining and friendship establishment, we simulated a 6LPN joining a network up to three hops from the 6LBR at 100 different locations, selected at random. Fig. 3 depicts the network topology, including one central 6LBR (black) and eight 6FNs (red) placed in a grid with 45 m between each device. The grey cloud shows the combined communication range of all 6FN, which is 50 m per 6FN. Blue dots represent all 100 locations where a 6LPN joined the network, at one hop (directly to the 6LBR), two hops, or three hops (corner-placed 6FNs) from the 6LBR. In all simulations, the EB interval was fixed at 16 s.

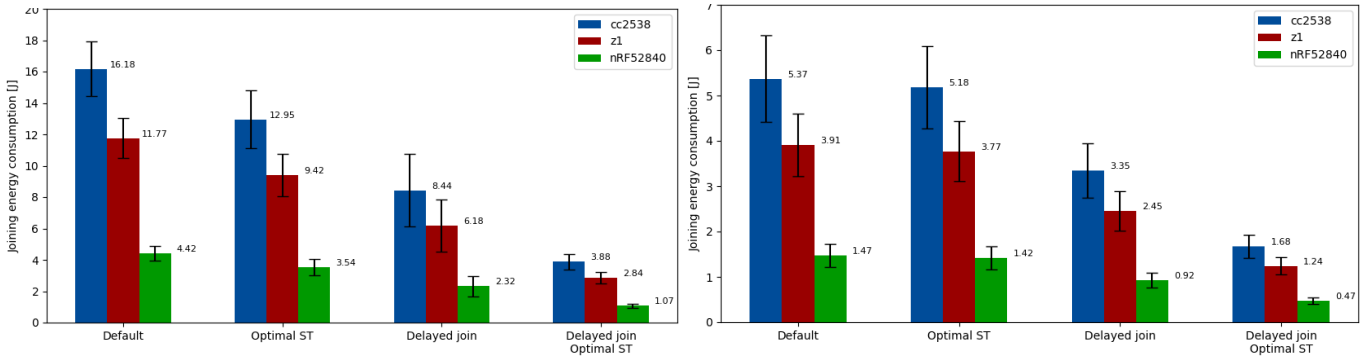
Fig. 4 shows the impact of the employed techniques to reduce energy consumption during joining and friendship establishment, presented in Section IV-C: (i) reduced number of advertisement channels, (ii) optimal scanning duration, and (iii) delayed joining. While decreasing the EB interval also has a significant impact on the joining time, we kept the interval at 16 s to not affect the energy consumption of the 6FNs and other nodes in the network. After all, decreasing the EB interval would decrease the energy consumption of joining 6LPNs at the expense of routers, which is in contrast to our goal of supporting constrained routers. Fig. 4a and 4b show the results for 16 and 2 advertisement channels, respectively. Each subplot covers the same four scenarios. The first scenario is a baseline Contiki-NG configuration with a default 1 s scanning duration and where each device is powered on at the same time. The second scenario uses the optimal scanning duration for the number of advertisement channels. In the third scenario, the 6LPN joins a converged network, where all 6FNs periodically transmit EBs. Finally, scenario four combines an optimal scanning duration with a delayed join. For each scenario, the energy consumption during joining and friendship establishment (up until the transmission of a DAO, see *A 3*) was determined using Energest for each of the 100 locations, using a different random seed for every simulation. Each bar in Fig. 4 shows the average joining energy consumption, in addition to the 95 % CI.

As shown in Fig. 4, the most significant impact on energy consumption arises when reducing the number of advertisement channels from 16 to 2, resulting in a 56-67 % reduction across all scenarios. When using delayed joining, energy consumption is decreased by 37-48 % compared to the default scenario, where all devices are powered on simultaneously. However, using an optimal scanning duration instead of delayed joining only diminishes energy consumption by 5-20 %. This can be explained by the fact that at some locations, the 6LPN is multiple hops away from the 6LBR. Therefore, 6FNs in between the 6LBR and 6LPN need to join the network before advertising it using EBs. In contrast, when using delayed joining, an optimal scanning duration reduces energy consumption by 45-50 %. As a result, combining a delayed joining procedure with an optimal scanning duration can reduce energy consumption up to 68-76 % compared to the default scenario. However, the biggest improvement is obtained when combining all three techniques, with a reduction of 90 % compared to the default scenario with 16 advertisement channels.

Finally, the radio RX state current consumption significantly influences joining energy consumption, as the device continuously listens for EBs. For instance, the Z1 consumes 73 %, and the nRF52840 27 % of the cc2538 energy consumption due to their different RX current consumption values (18.8 mA, 6.53 mA, and 24 mA, respectively).

B. Operational Mode

As pointed out in Section IV-B2, choosing the KA timeout results in a compromise between power consumption and downlink latency. This is because downlink traffic can only be received in response to a KA polling frame. To illustrate the impact of this trade-off, power consumption and latency were measured in Cooja simulations, comparing a 6LPN to a conventional 6LN. In each simulation, two devices were considered: a 6LPN and a 6FN or a 6LN and a 6LR. The 6FN or 6LR transmitted periodic downlink frames of 20 B, with the transmission interval varying from one to five minutes, including a reference scenario without downlink traffic, equaling six transmission intervals. To ensure an operational mode, the power consumption was measured after two hours and lasted for an hour of simulation time. In addition, the downlink latency was measured for each transmitted frame, with the KA timeout ranging from 15 s to 240 s, totaling seven



(a) 16 advertisement channels

(b) 2 advertisement channels

Fig. 4: Impact of number of advertisement channels, optimal scanning duration, and delayed joining on joining energy consumption for the cc2538, Z1, and nRF52840 platforms. Each bar shows the average joining energy consumption with a 95% Confidence Interval (CI). Reducing the number of advertisement channels has the biggest impact on energy consumption (56-67 % reduction), after delayed joining (37-48 % reduction), and optimal scanning time (5-20 % reduction). However, combining all techniques yields a 90 % reduction.

different KA timeouts. Each scenario was repeated 10 times with a different random seed, resulting in 420 simulations of 1 hour.

Table IV lists the difference in average power consumption and latency between a 6LN and a 6LPN, and between a 6LR and a 6FN. To ensure a fair comparison, power consumption values assume no data traffic, and both power consumption and latency values assume a KA timeout of 15 s, similar to the 16 s EB interval used for the 6LN and 6LR. Additional simulations were performed to measure the uplink latency from the 6LPN and 6LN to the 6FN and 6LR, and the downlink latency from the 6LR to the 6LN. These three additional scenarios account for 30 simulations in total, each with a different random seed. Each simulation used a transmission interval of 1 min during 2 hours of stable operation. In the absence of downlink data, there is a 88 % reduction in power consumption during operational mode for the Z1 and nRF52840 platforms. In contrast, the cc2538 shows a 94 % reduction due to its higher current consumption during the RX state, which is limited on the 6LPN compared to a 6LN. Furthermore, the 6FN has a negligible 0.4 % increase in power consumption compared to a 6LR, ensuring the possibility of constrained routers. However, this comes at the cost of increased average downlink latency from 193 ms to 7.62 s and maximum latency from 320 ms to 15.62 s. This is due to the downlink queueing at the 6FN. In contrast, uplink latency is only slightly affected. As a result, industrial control applications with critical up- and downlink latency requirements cannot be fulfilled. However, for industrial monitoring applications, only guaranteed uplink latency is necessary [24]. Therefore, 6LPNs can be suitable for integration in these kinds of networks as they offer a guaranteed uplink latency, albeit with a reduced downlink latency performance.

Fig. 5 shows the power consumption latency trade-off for the cc2538 (Fig. 5a), Z1 (Fig. 5b), and nRF52840 (Fig. 5c), using dotted lines to denote average and maximum downlink latency for the selected KA timeout. Colored full lines represent power consumption with and without downlink data, with a period ranging from 1 to 5 min. Power consumption without data traffic can be further reduced to around 90 %

TABLE IV: Simulated power consumption and latency values comparing 6LN with 6LPN and 6LR with 6FN, assuming no data traffic for power consumption values and a KA timeout of 15 s for all values. Power consumption is reduced significantly (87-94 %) for 6LPNs without affecting 6FNs. Due to queued downlink traffic, average latency is increased to 7.62 s.

	Power consumption (no data)			Latency (KA 15 s)	
	Z1	cc2538	nRF52840	Avg.	Max.
6LN	841 µW	1076.8 µW	309.9 µW	193 ms	247 ms
6LPN	108.1 µW	65.1 µW	36.6 µW	208 ms	289 ms
	-87.15 %	-93.95 %	-88.19 %	+7.77 %	+17 %
6LR	837.6 µW	1072.4 µW	307.7 µW	193 ms	320 ms
6FN	840.6 µW	1076.1 µW	309 µW	7.62 s	15.62 s
	+0.36 %	+0.35 %	+0.42 %	+3848 %	+4781 %

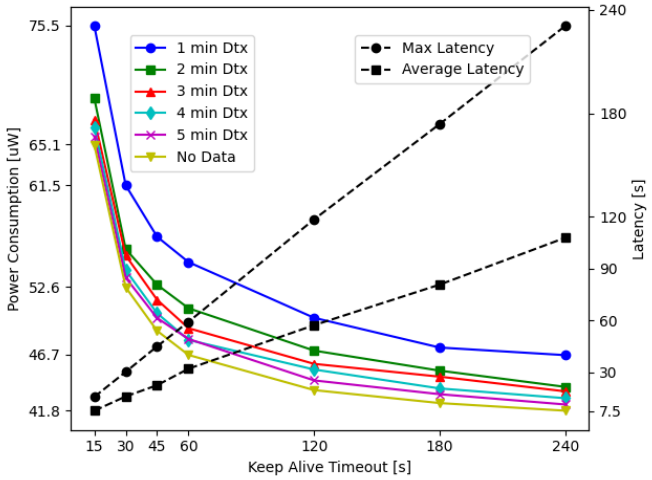
(Z1 and nRF52840) and 96 % (cc2538) by extending the KA timeout to 4 min. However, this inevitably increases latency, averaging around 2 min and peaking at 4 min. As shown in Fig. 5, increasing the KA timeout above 60 s only marginally decreases power consumption while further increasing downlink latency. Furthermore, for all platforms and KA timeouts, receiving downlink traffic causes only a marginal increase in power consumption by a few micro-watts.

Unlike the energy consumption during joining, which was dominated by the RX state, power consumption during operational mode is primarily influenced by the LPM. Therefore, the nRF52840 still performs best, consuming 36.6 µW for a 15 s KA timeout and no downlink traffic. However, the cc2538 now outperforms the Z1 at 65.1 µW compared to 108.1 µW.

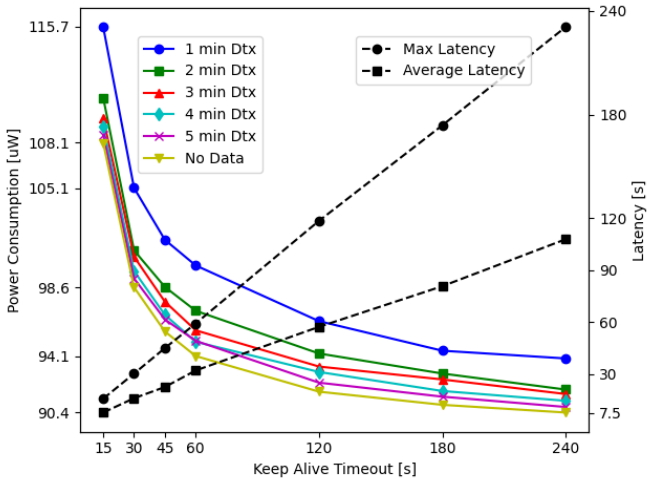
VI. Conclusion

6TiSCH networks offer deterministic latency, high reliability, and low power consumption in IWSNs, supporting IPv6 connectivity. However, despite the low power consumption of 6TiSCH devices, an energy-intensive joining procedure and idle listening hamper an acceptable battery lifetime or the integration of battery-less devices. Our two novel 6TiSCH device types (the 6LPN and 6FN) solve these challenges by providing an optimized joining procedure and eliminating idle listening to provide ultra-low-power 6TiSCH devices.

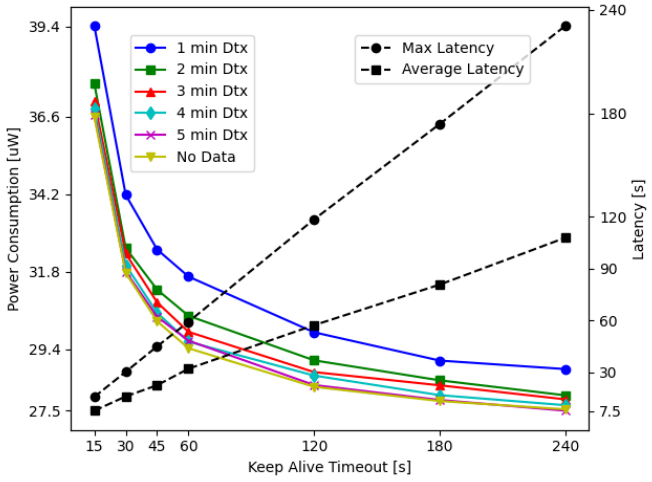
Our analysis shows that reducing the number of advertisement channels has the largest impact on energy consumption



(a) cc2538



(b) z1



(c) nRF52840

Fig. 5: Latency vs power consumption trade-off for 6LPNs. Dotted lines show the average and maximum latency versus the KA timeout, while colored lines show the power consumption assuming periodic downlink traffic with a period of 1 to 5 min, in addition to a no data reference. Results show a clear trade-off in the choice of KA timeout, where an increasing timeout results in decreased power consumption and increased latency.

during the joining procedure, reducing the energy consumption by 56-67 %. However, combining several techniques, such as reducing advertisement channels, optimizing scanning time, and implementing a delayed join, yields a 90 % reduction in energy consumption. Therefore, combining multiple techniques is crucial to achieving optimal energy reduction during the joining procedure, which remains the highest energy-consuming period in the device's lifetime.

During operational mode, a 6LPN shows an 87-94 % power consumption reduction compared to a conventional 6LN, despite an increased average downlink latency of 7.62 s. While this does not satisfy the requirements of industrial control applications with strict downlink latency needs, it proves suitable for industrial monitoring applications as it maintains uplink latency guarantees. A trade-off exists between power consumption and downlink latency by choosing the KA timeout. However, increasing the KA timeout further than 1 min only results in negligible power consumption gains with an increased downlink latency. The idle current consumption emerges as the dominant factor during operational mode, with platforms featuring high RX current consumption experiencing more substantial power consumption reductions for 6LPNs due to the elimination of idle listening.

As a result, platforms with low RX current and efficient low-power modes are ideal for implementing 6LPNs, ensuring optimal energy limitation during both joining and operational modes. Future advancements in radio chips and power management circuits will result in even more energy-efficient platforms, further reducing the power consumption of 6LPNs. This not only extends the lifetime of battery-powered devices but also paves the way for the integration of battery-less devices into 6TiSCH networks, improving the sustainability of IWSNs.

ACKNOWLEDGMENT

Part of this research was funded by the Flemish FWO SBO S001521N IoBaLeT (Sustainable Internet of Batteryless Things) project.

REFERENCES

[1] D. Van Leemput, J. Hoebeke, and E. De Poorter, "Integrating Battery-Less Energy Harvesting Devices in Multi-Hop Industrial Wireless Sensor Networks," *IEEE Communications Magazine*, vol. 62, no. 7, pp. 66–73, July 2024.

[2] A. Kalita and M. Khatua, "6TiSCH – IPv6 Enabled Open Stack IoT Network Formation: A Review," *ACM Trans. Internet Things*, vol. 3, no. 3, July 2022. [Online]. Available: https://doi.org/10.1145/3536166

[3] Y. Tanaka, P. Minet, T. Watteyne, and F. Teraoka, "Accelerating 6tisch network formation," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, July 2021, pp. 18–24.

[4] A. Karalis, D. Zorbas, and C. Douligeris, "Optimal initial synchronization time in the minimal 6tisch configuration," *IEEE Access*, vol. 9, pp. 69 316–69 334, 2021.

[5] Z. J. Chew, T. Ruan, and M. Zhu, "Energy Savvy Network Joining Strategies for Energy Harvesting Powered TSCH Nodes," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1505–1514, February 2021.

[6] K. Das, P. Zand, and P. Havinga, "Industrial Wireless Monitoring with Energy-Harvesting Devices," *IEEE Internet Computing*, vol. 21, no. 01, pp. 12–20, January 2017.

[7] X. Fafoutis, A. Elsts, G. Oikonomou, R. Piechocki, and I. Craddock, "Adaptive static scheduling in ieee 802.15.4 tsch networks," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb 2018, pp. 263–268.

[8] S. Scanzio, G. Cena, A. Valenzano, and C. Zunino, "Energy Saving in TSCH Networks by Means of Proactive Reduction of Idle Listening," in *Ad-Hoc, Mobile, and Wireless Networks*, L. A. Grieco, G. Boggia, G. Piro, Y. Jararweh, and C. Campolo, Eds. Cham: Springer International Publishing, 2020, pp. 131–144.
[9] G. Cena, S. Scanzio, and A. Valenzano, "Enabling listening suspension in the time slotted channel hopping protocol," in *2021 17th IEEE International Conference on Factory Communication Systems (WFCS)*, June 2021, pp. 19–26.
[10] S. Scanzio, G. Cena, and A. Valenzano, "Enhanced energy-saving mechanisms in tsch networks for the iiot: The pril approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 7445–7455, June 2023.
[11] "Mesh Profile Bluetooth Specifications," Technical Report. [Online]. Available: https://www.bluetooth.com/specifications/specs/
[12] "Thread Network Fundamentals," Thread Group Inc, White Paper, September 2022. [Online]. Available: https://www.threadgroup.org/support
[13] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," RFC 8180, May 2017. [Online]. Available: https://www.rfc-editor.org/info/rfc8180
[14] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.
[15] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)," RFC 9033, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9033
[16] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, November 2015, p. 337–350.
[17] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, March 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6550
[18] P. Levis, T. H. Clausen, O. Gnawali, J. Hui, and J. Ko, "The Trickle Algorithm," RFC 6206, March 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6206
[19] A. Karalis, D. Zorbas, and C. Douligeris, "Reducing the Synchronization Time in the Minimal 6TiSCH Configuration Under Channel Uncertainty," in *2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2023, pp. 63–69.
[20] X. Vilajosana, P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Standardized Low-Power Wireless Communication Technologies for Distributed Sensing Applications," *Sensors*, vol. 14, no. 2, pp. 2663–2682, 2014.
[21] "Z1 Datasheet," Zolertia, March 2010, Rev. C. [Online]. Available: https://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf
[22] "CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee® Applications," Texas Instruments, April 2015, Rev. D. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc2538.pdf
[23] "nRF52840 Product Specification," Nordic Semiconductor, December 2023, v1.8. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.8.pdf
[24] 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, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/2/488