

# Wireless Sensor Networks based on TSCH/TDMA with Power Consumption and Latency Constraints

Stefano Scanzio\*, Gabriele Formis\*<sup>†</sup>, Tullio Facchinetti<sup>‡</sup>, Giacomo Paolini<sup>§</sup>, Gianluca Cena\*

\*National Research Council of Italy (CNR-IEIIT), Italy. <sup>†</sup>Politecnico di Torino, Italy.

<sup>‡</sup>University of Pavia, Italy. <sup>§</sup>University of Bologna, Italy.

Emails: stefano.scanzio@cnr.it, gabriele.formis@polito.it, tullio.facchinetti@unipv.it,

giacomo.paolini4@unibo.it, gianluca.cena@cnr.it

**Abstract**—One of the main goals of wireless sensor networks is to permit the involved nodes to communicate with low energy budgets, as they are typically battery-powered. When such networks are employed in industrial scenarios, constraints about latency may have a significant role, too. The TSCH mechanism, and more in general TDMA schemes, rely on traffic scheduling, and consequently they can feature low power consumption and more predictable latency. Some recent proposals like PRIL-M enable further consistent energy savings, but unfortunately they cause at the same time a dramatic increase in latency.

This work presents an extension of PRIL-M, we named PRIL-ML, that achieves a significantly shorter latency in exchange for a slight increase in power consumption. Its operating principles are first illustrated, then some approximate equations are provided for assessing analytically the improvements it achieves, starting from simulation results obtained for both standard TSCH and the original PRIL-M technique.

## I. INTRODUCTION

Wireless sensor networks (WSNs) are gaining ground in the industrial network landscape, which is characterized by a high degree of heterogeneity in terms of the employed communication protocols and the related characteristics [1]. One of the main objectives of WSNs is to achieve extremely low power consumption. Nonetheless, other objectives may exist as well. For example, the ability to ensure predictable and bounded latency (although orders of magnitude larger than typical industrial protocols) may sometimes be of non-negligible importance. The time slotted channel hopping (TSCH) [2] protocol defined in the IEEE 802.15.4 standard [3] relies on traffic scheduling [4] and provides improvements concerning both power consumption and determinism [5], [6]. The basic TSCH access technique splits time into fixed-length slots and resembles time division multiple access (TDMA). Additionally, channel hopping makes TSCH more robust against narrowband disturbance [7], [8].

Several techniques have been developed to enhance energy efficiency in TSCH-based WSNs. The approach in [9] involves non-orthogonal multiple access (NOMA) use, which improves spectral efficiency, reduces latency, and increases energy efficiency by allowing sensor nodes to transmit data to their cluster heads using NOMA transmissions. Another technique

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”).

is the 6TiSCH Low Latency Autonomous Scheduling [10], which reduces latency and radio duty cycle, thus enhancing energy efficiency and reliability in Industrial Internet of Things (IIoT) applications. Additionally, the MSU-TSCH algorithm [11] supports mobility and adapts to topology changes, significantly reducing energy consumption and end-to-end latency. The DualBlock scheme [12] introduces an adaptive intra-slot CSMA/CA mechanism, reducing collisions and energy consumption within congested slots.

Besides above techniques, Proactive Reduction of Idle Listening (PRIL) has been also proposed recently. Its initial version (PRIL-F) [13], [14] permits to lower energy consumption on the first-hop of every path. Subsequently, it has been extended to properly manage multi-hop networks as well (PRIL-M) [15]. In practice, PRIL-F decreases power consumption of nodes located one hop away from the source node, whereas PRIL-M deals with all the following hops to the destination. From the point of view of energy reduction, PRIL-M performance is optimal. Unfortunately, it leads to a dramatic increase of communication latency, which makes it unsuitable in several specific industrial contexts.

In this paper, PRIL multi-hop with latency constraints (PRIL-ML) is proposed, which is significantly more responsive than PRIL-M by slightly increasing energy consumption. Due to the similarity between TSCH and TDMA, all the ideas described here can be easily applied to the latter too.

The paper organization is the following: in Section II, the TSCH protocol and the former PRIL-M technique are described, while the new PRIL-ML technique is introduced in Section III. Results about TSCH, PRIL-M, and some analytical estimates about PRIL-ML are reported in Section IV, before Section V that concludes the work.

## II. TSCH AND PRIL-M

As depicted in Fig. 1, TSCH partitions the wireless transmission medium on both the time and frequency domains by means of a slotframe matrix. Time partitioning is a sort of TDMA, where time is split into a number of fixed-size slots (often there are 101, whose duration is typically either 10 or 20 ms). Frequency division enables instead concurrent transmissions (on up to 16 channels in the 2.4 GHz band), which increase the overall network throughput. Every cell in the matrix, identified by its slot and channel offsets, is allocated

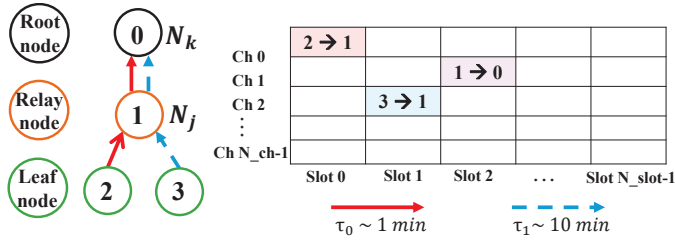


Fig. 1. Example of TSCH operation and slotframe matrix.

to one (or more) specific links. Besides preventing collisions, traffic scheduling also leads to a consistent reduction of power consumption, since nodes can stay in a deep-sleep state and wake up only in those slots where they must either transmit or receive a frame. Additionally, channel hopping provides more stable links by decreasing variability of the frame loss ratio [8]. In fact, the slotframe repeats over time (period is 2.02 s in the case of 101 slots of width 20 ms) and, at every repetition, the slot scheduled to the link is mapped to a different physical channel, providing both time and frequency diversity.

Fig. 1 depicts a multi-hop network of four nodes organized in two layers (the tree routing topology is constructed, e.g., by the RPL protocol), which is used as a running example throughout this work. The global schedule is reported in the matrix on the right side of the figure. Relevant portions of it are stored inside every node, which manage its possible states. For instance,  $N_1$  switches to the *receive* state in slots 0 and 1, and in the *transmission* state in slot 2 (only if its queue contains at least one frame addressed to node  $N_0$ ). As the figure shows, two periodic flows are defined,  $\tau_0$  and  $\tau_1$ , which originate from leaf nodes and are directed to the root. The nodes traversed by the packets of  $\tau_0$  are  $N_2$ ,  $N_1$ , and  $N_0$ , while for  $\tau_1$  they are  $N_3$ ,  $N_1$ , and  $N_0$ . Due to tolerances and random drifts of the nodes' quartz oscillators, the generation processes of flows (defined in terms of their nominal rate and initial phase) are typically asynchronous. Generation periods in our sample network were set to  $T_{\tau_0} \simeq 1$  min and  $T_{\tau_1} \simeq 10$  min, but not necessarily they have to be multiples of each other. These aspects have been taken into account in this work.

The presence of flows with quite long (and different) periods is commonplace in WSNs. Since (at least) one transmission opportunity (i.e., cell) is configured in the slotframe for any given link every 2.02 s, while generations periods are much larger (1 and 10 min), most of the reserved slots remain unused. This makes receiving nodes waste a non-negligible amount of energy waiting for frames that will never arrive, a phenomenon known as *idle listening*. The simulator was configured in such a way that the energy spent for sending a maximal-size frame (127 B), for receiving it, and wasted for idle listening is  $E_{\text{send}} = 485.7 \mu\text{J}$ ,  $E_{\text{rec}} = 651.0 \mu\text{J}$ , and  $E_{\text{listen}} = 303.3 \mu\text{J}$ , respectively. These values are derived from the experimental results reported in [16], related to an OpenMoteSTM device equipped with an Atmel AT86RF231 radiochip and an STM32F103RB 32-bit microcontroller.

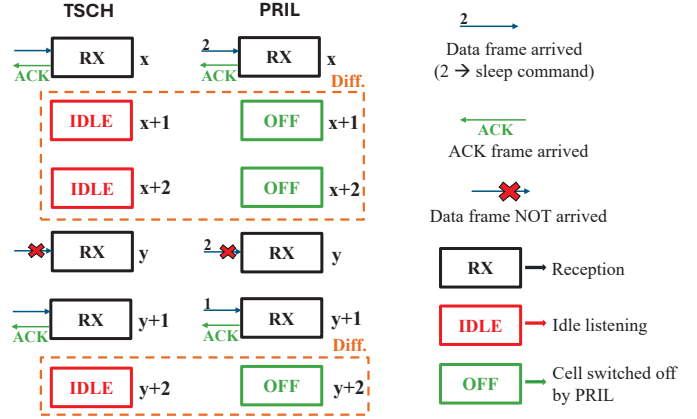


Fig. 2. Example of PRIL-F operation compared with default TSCH.

PRIL techniques rely on specific sleep commands, which are embedded in the frames and permit the sender to turn off the receiver for a given amount of time, in such a way to prevent (or just lower) the impact of idle listening. If only periodic traffic is considered, the receiving side of the link that corresponds to the first hop of the path followed by the packets of a given flow can be disabled precisely, without any impact on latency but with a significant decrease in energy consumption. This technique is termed PRIL-F and is described in [13].

The simple example in Fig. 2 compares the behavior of PRIL-F and standard TSCH. Rectangular shapes represent the instances of the slot allocated to the link in subsequent slotframes. The packet generation period at the application level is set equal to 3 slotframes, i.e., 6.06 s. In PRIL-F, the sleep command (with value 2) added to the frame (at time  $x$ ) disables the following two slots on the receiver (i.e., it sets their state to OFF), preventing idle listening completely. In the second exchange (at time  $y$ ) the initial frame is assumed to be lost, and hence only one slot can be disabled by PRIL-F.

PRIL-M has been conceived to complement PRIL-F (which only deals with the first hop), and operates on the following hops (e.g., from  $N_1$  to  $N_0$  in Fig. 1). Since the involved links can be traversed by multiple flows with different periods, relay nodes set the duration of sleep commands based on the fastest flow they see, i.e., the one with the minimum period (in the example of Fig. 1,  $\tau_0$ , whose period is  $T_{\tau_0} = 1$  min). This permits to prioritize this flow, which is expected to have more demanding timing requirements. On the one hand this leads to a significant reduction in energy consumption, but on the other it causes a substantial latency increase for the other flows. Detailed explanations about PRIL-M can be found in [15].

### III. PRIL-ML

The basic idea behind the PRIL-ML technique, first presented in this paper, is to decrease the time for which the receiving side of a link is turned off by PRIL-M, to limit its impact on latency. A preliminary analysis carried out in this study highlights a noticeable decrease in latency, with a

negligible increase in energy consumption. As for PRIL-M, PRIL-ML only acts starting from the second hop of a path.

Referring to Fig. 1 and focusing on the link  $j \rightarrow k$  between node  $N_j$  and its neighbor  $N_k$ , PRIL-M permits  $N_j$  to disable those cells of  $N_k$  that are affected by idle listening. Among all the flows  $\tau_i$  traversing it, node  $N_j$  considers those directed to  $N_k$  (a separate instance of PRIL-M is created for every outgoing link). Let  $\mathcal{T}_{j \rightarrow k} = \{\tau_i | \tau_i \text{ crosses } N_j \text{ and then } N_k\}$  be the set of all the flows that traverse both  $N_j$  and  $N_k$ , in that order, and  $T_{\min}^{j \rightarrow k} = \min(T_{\tau_i} | \tau_i \in \mathcal{T}_{j \rightarrow k})$  the minimum period between flows belonging to  $\mathcal{T}_{j \rightarrow k}$ . The related (fastest) flow is identified with the symbol  $\tau_*^{j \rightarrow k}$ . The PRIL-M algorithm foresees that  $N_j$  periodically turns off part of the slots scheduled to link  $j \rightarrow k$  in such a way to prevent idle listening on  $N_k$ . In particular, this procedure is triggered by the reception of packets from  $\tau_*^{j \rightarrow k}$ , which occurs with period  $T_{\min}^{j \rightarrow k}$ .

Since packet generation for the different flows is asynchronous, we can assume that a packet  $p_i$  that belongs to  $\tau_i$  ( $p_i \in \tau_i$ ), with  $\tau_i \neq \tau_*^{j \rightarrow k}$ , arrives to node  $N_j$  (where it is enqueued for relay) at a time uniformly distributed in the interval  $[0, T_{\min}^{j \rightarrow k}]$  between two subsequent packets belonging to  $\tau_*^{j \rightarrow k}$ . This means that, due to the way relay is performed by  $N_j$ , the average latency of packet  $p_i$  increases by  $\Delta_{\mu}^{\text{PRIL-M}} = T_{\min}^{j \rightarrow k}/2$ , while its worst-case latency grows by  $\Delta_{\max}^{\text{PRIL-M}} = T_{\min}^{j \rightarrow k}$  (this happens when  $p_i$  arrives to  $N_j$  just after the packet belonging to  $\tau_*^{j \rightarrow k}$  has been sent, which temporarily deactivates all the following cells that cause idle listening on receiver  $N_k$ ). Above simplified formulas do not take into account the effect of retransmissions on latency and power consumption (which are instead modeled by the simulator properly). Nevertheless, experimental data of Section IV show that they provide a good approximation of reality, and are adequate to evaluate the effect of the proposed PRIL-ML technique.

The sleep command is included by  $N_j$  (encoded as a TSCH *information element*) every time it relays the last packet in the  $j \rightarrow k$  buffer. Hence, upon the arrival of a packet from  $\tau_*^{j \rightarrow k}$ , all enqueued packets are always forwarded according to TSCH rules, before the link is switched off by PRIL. The sleep duration in PRIL-M is derived from the value  $T_{\min}^{j \rightarrow k}$ , by subtracting the actual time taken to empty the buffer by sending all the queued frames. Since transmission takes place in First-In First-Out order, the packet from  $\tau_*^{j \rightarrow k}$  can be preceded and followed in the queue by packets from other flows. The main idea behind PRIL-ML is to set the nominal sleep period  $T_{\text{act}}^{j \rightarrow k}$  of outgoing link  $j \rightarrow k$  to a sub-multiple of  $T_{\min}^{j \rightarrow k}$ , that is,  $T_{\text{act}}^{j \rightarrow k} = \lceil T_{\min}^{j \rightarrow k} / r \rceil$ . In theory, any duration  $T_{\text{act}}^{j \rightarrow k} \leq T_{\min}^{j \rightarrow k}$  could be chosen, but we believe this does not add much from the applications' perspective. If, for instance, we set  $r = 4$ , the link between  $N_j$  and  $N_k$  is reopened after a time equal to about  $T_{\min}^{j \rightarrow k} / 4$ , allowing node  $N_j$  to flush any queued packets four times as fast.

The increase in power consumption over PRIL-M due to PRIL-ML adoption is equal, at worst, to  $\Delta P = \frac{(r-1) \cdot E_{\text{listen}}}{T_{\min}^{j \rightarrow k}}$ , where  $E_{\text{listen}}$  is the energy wasted by the receiver when idle

TABLE I  
ENERGY CONSUMPTION FOR THE DIFFERENT TECHNIQUES AND NODES

Technique	Power ( $\mu W$ )	$N_0$	$N_1$	$N_2$	$N_3$	All
TSCH	$P_{\text{send}}$	0.0	11.1	10.1	1.0	22.2
	$P_{\text{rec}}$	14.8	14.9	0.0	0.0	29.7
	$P_{\text{listen}}$	143.2	293.4	0.0	0.0	436.6
	$P$	158.0	319.4	10.1	1.0	488.5
PRIL-M	$P_{\text{send}}$	0.0	19.9	18.9	1.9	40.7
	$P_{\text{rec}}$	13.8	13.7	0.0	0.0	27.5
	$P_{\text{listen}}$	0.4	0.0	0.0	0.0	0.4
	$P$	14.2	33.6	18.9	1.9	68.6

TABLE II  
END-TO-END LATENCY FOR THE DIFFERENT TECHNIQUES AND FLOWS

Tech.	Latency (s)	$\mu_d$	$\sigma_d$	$d_{\min}$	$d_{p99}$	$d_{p99.9}$	$d_{p99.99}$	$d_{\max}$
TSCH	$\tau_0$	1.644	1.300	0.060	5.960	8.360	11.12	18.12
	$\tau_1$	1.731	1.413	0.040	6.440	9.500	11.90	17.96
	$\tau_0 + \tau_1$	1.652	1.310	0.040	5.980	8.560	11.28	18.12
PRIL-M	$\tau_0$	2.658	1.587	0.060	7.940	10.64	13.72	22.00
	$\tau_1$	<b>30.58</b>	<b>17.20</b>	<b>0.040</b>	<b>60.38</b>	<b>63.20</b>	<b>65.94</b>	<b>69.86</b>
	$\tau_0 + \tau_1$	5.196	9.676	0.040	53.86	60.30	63.04	69.86

listening takes place in one slot.  $\Delta P$  represents an upper bound because packets queued in  $N_j$  would be sent anyway. Regarding latency, the increase in the relay time of  $N_j$  over TSCH due to PRIL-ML adoption is equal, on average, to  $\Delta_{\mu}^{\text{PRIL-ML}} = T_{\text{act}}^{j \rightarrow k}/2$ , while in the worst case it grows by  $\Delta_{\max}^{\text{PRIL-ML}} = T_{\text{act}}^{j \rightarrow k}$ . In fact, every time the outgoing link  $j \rightarrow k$  is re-opened, packets enqueued in  $N_j$  can be sent without further delays. This means that latency is  $r$  times shorter than PRIL-M.

Regarding the implementation of PRIL-ML, when a frame delivering an extended sleep message (bearing both  $T_{\text{act}}$  and  $r$ ) arrives on the receiving side of link  $j \rightarrow k$ , it triggers a sequence of actions in node  $N_k$ . In particular, for  $r$  times the receiver is put to sleep for  $T_{\text{act}}^{j \rightarrow k}$  (the final sleeping time may be slightly shorter), awakening it every time for (at least) one slot, as if flow  $\tau_*^{j \rightarrow k}$  were  $r$  time faster than it actually is. PRIL-ML complexity is reasonably low [15], which enables inclusion in inexpensive motes.

#### IV. RESULTS

The TSCHmodeler discrete-event simulator was used to simulate the network in Fig. 1 with both standard TSCH and PRIL-M (it has not been extended to PRIL-ML yet). Every simulation has a duration of 10 years, and the slotframe matrix was configured so that the first three slots are allocated to transmissions from  $N_2$  to  $N_1$ , from  $N_3$  to  $N_1$ , and from  $N_1$  to  $N_0$ , respectively. An experimental campaign was preliminary carried out (not reported here for space reasons) that shows that the results provided by the simulator for a TSCH network are very similar to those obtained on real OpenMote B devices running OpenWSN (6TiSCH).

In the case of periodic traffic with a known generation period, the effect of PRIL-M on energy consumption is extremely positive, as shown by the results reported in Table I. As can be seen, the idle listening component of energy consumption is practically zero for all nodes in the network. Unfortunately,

PRIL-M impacts negatively on end-to-end latency, as shown in Table II, which reports the main statistics about flows  $\tau_0$  and  $\tau_1$ , namely average value ( $\mu_d$ ), standard deviation ( $\sigma_d$ ), minimum ( $d_{\min}$ ), 99-percentile ( $d_{p99}$ ), 99.9-percentile ( $d_{p99.9}$ ), 99.99-percentile ( $d_{p99.99}$ ), and maximum ( $d_{\max}$ ), for both standard TSCH and PRIL-M.

Regarding PRIL-M, latency increase for the fastest flow  $\tau_0$  over TSCH is rather small compared to  $\tau_1$ . This can be explained by looking at the example of Fig. 1. Any retransmissions of a packet belonging to  $\tau_0$  on the link  $2 \rightarrow 1$  cause a delay on its arrival on  $N_1$ . If the queue of  $N_1$  is empty, the packet will be forwarded with a sleep command equal to  $T_{\min}^{1 \rightarrow 0} = T_{\tau_0} \simeq 1$  min, otherwise a more complex algorithm is executed as described in [15]. If the next packet belonging to  $\tau_0$ , which is generated by  $N_2$  after a time  $T_{\tau_0}$ , arrives to node  $N_1$  without retransmissions, it finds the link  $1 \rightarrow 0$  in the OFF state and suffers from an additional delay that increases latency beyond what the packet would experience without PRIL-M.

The main problem concerns, however, the steep latency increase for flow  $\tau_1$ , whose statistics in Table II are set in boldface. The simulated average latency for PRIL-M is  $\mu_d^{\text{PRIL-M}} = 30.58$  s, which resembles the value obtained with our approximate model by summing  $\Delta_{\mu}^{\text{PRIL-M}} = T_{\min}^{1 \rightarrow 0} / 2$  to the average latency of TSCH, that is,  $\hat{\mu}_d^{\text{PRIL-M}} = \mu_d^{\text{TSCH}} + \Delta_{\mu}^{\text{PRIL-M}} = 1.731 + 30 = 31.731$  s. A similar reasoning can be done to estimate the maximum latency as  $\hat{d}_{\max}^{\text{PRIL-M}} = d_{\max}^{\text{TSCH}} + \Delta_{\max}^{\text{PRIL-M}} = 17.96 + 60 = 77.96$  s. In this case, the theoretical value differs in a more pronounced way from the simulated one  $d_{\max}^{\text{PRIL-M}} = 69.86$  s, as the maximum is more sensitive to the actual occurrence of rare events (for example, having a high number of retransmissions for the same packet on both links  $3 \rightarrow 1$  and  $1 \rightarrow 0$ ).

At present, we are not able to simulate PRIL-ML behavior yet. Nevertheless, we can apply our simple theoretical model to find satisfactory estimates. By setting  $r = 4$ , we obtain  $\Delta_{\mu}^{\text{PRIL-ML}} = 7.5$  s and  $\Delta_{\max}^{\text{PRIL-ML}} = 15$  s. Consequently, approximated values for the average and worst-case latency can be evaluated as  $\hat{\mu}_d^{\text{PRIL-ML}} = 9.231$  s and  $\hat{d}_{\max}^{\text{PRIL-ML}} = 32.96$  s, respectively. The price to be paid for latency decrease is the rise of power consumption. Luckily, this increase is fairly limited compared to the total power consumption of TSCH ( $P^{\text{TSCH}} = 488.5 \mu\text{W}$ ), and can be evaluated as  $\Delta P = (4 - 1) \cdot 303.3 \mu\text{J} / 60 \text{ s} = 15.2 \mu\text{W}$ .

This implies that the overall power consumption of the entire network when PRIL-ML is employed is approximately  $\hat{P}^{\text{PRIL-ML}} = P^{\text{PRIL-M}} + \Delta P = 68.6 + 15.2 = 83.8 \mu\text{W}$ . Such increase only affects the idle listening component of node  $N_0$ . By using our theoretical model for PRIL-ML, this specific contribution grows from  $P_{\text{listen}}^{\text{PRIL-M}} = 0.4 \mu\text{W}$  up to  $\hat{P}_{\text{listen}}^{\text{PRIL-ML}} = 0.4 + 15.2 = 15.6 \mu\text{W}$ , which remains quite small compared to TSCH, where  $P_{\text{listen}}^{\text{TSCH}} = 143.2 \mu\text{W}$ .

## V. CONCLUSION

A new improvement for TSCH networks, which was named PRIL-ML, has been proposed in this paper to address the main drawback of the former PRIL-M technique, that is,

a significant increase in communication latency. Although PRIL-M succeeds in minimizing energy consumption as much as possible, its considerably high delays may prevent its adoption in time-sensitive applications.

PRIL-ML permits to adjust the trade-off between energy consumption and end-to-end latency, and consequently it makes it possible to customize the network parameters based on the application requirements. For instance, the average latency can be shrunk to less than one-third of PRIL-M with an increase in the total power consumption of just 22%. As future work we plan to extend the simulator to tackle PRIL-ML, so that it can be studied in detail.

## REFERENCES

- [1] S. Scanzio, L. Wisniewski, and P. Gaj, "Heterogeneous and dependable networks in industry – A survey," *Computers in Industry*, vol. 125, p. 103388, 2021.
- [2] X. Vilajosana, T. Watteyne, T. Chang, M. Vućinić, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A Tutorial," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 595–615, 2020.
- [3] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.
- [4] A. Tabouche, B. Djamaa, and M. R. Senouci, "Traffic-Aware Reliable Scheduling in TSCH Networks for Industry 4.0: A Systematic Mapping Review," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2834–2861, 2023.
- [5] 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. 167 042–167 058, 2020.
- [6] G. Cena, C. G. Demartini, M. Ghazi Vakili, S. Scanzio, A. Valenzano, and C. Zunino, "Evaluating and Modeling IEEE 802.15.4 TSCH Resilience against Wi-Fi Interference in New-Generation Highly-Dependable Wireless Sensor Networks," *Ad Hoc Networks*, vol. 106, p. 102199, 2020.
- [7] J. Muñoz, P. Muhlethaler, X. Vilajosana, and T. Watteyne, "Why Channel Hopping Makes Sense, even with IEEE802.15.4 OFDM at 2.4 GHz," in *2018 Global Internet of Things Summit (GloTS)*, 2018, pp. 1–7.
- [8] G. Cena, S. Scanzio, M. G. Vakili, C. G. Demartini, and A. Valenzano, "Assessing the Effectiveness of Channel Hopping in IEEE 802.15.4 TSCH Networks," *IEEE Open Journal of the Industrial Electronics Society*, vol. 4, pp. 214–229, 2023.
- [9] N. Gholami, N. Moghim, M. Ghazvini, and S. Haghani, "Utilizing Non-Orthogonal Multiple Access for Both Latency and Energy Efficiency Improvement in TSCH-Based WSNs," *IEEE Access*, vol. 10, pp. 28 922–28 937, 2022.
- [10] N. M. Pradhan, B. S. Chaudhari, and M. Zennaro, "6TiSCH Low Latency Autonomous Scheduling for Industrial Internet of Things," *IEEE Access*, vol. 10, pp. 71 566–71 575, 2022.
- [11] W. Jerbi, O. Cheikhrouhou, A. Guermazi, and H. Trabelsi, "MSU-TSCH: A Mobile Scheduling Updated Algorithm for TSCH in the Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 7, pp. 7978–7985, 2023.
- [12] J. Park, H. Kim, H.-S. Kim, and S. Bahk, "DualBlock: Adaptive Intra-Slot CSMA/CA for TSCH," *IEEE Access*, vol. 10, pp. 68 819–68 833, 2022.
- [13] 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*, pp. 1–11, 2022.
- [14] 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.
- [15] S. Scanzio, F. Quarta, G. Paolini, G. Formis, and G. Cena, "Ultra-Low Power and Green TSCH-Based WSNs With Proactive Reduction of Idle Listening," *IEEE Internet of Things Journal*, pp. 1–12, 2024.
- [16] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, "A Realistic Energy Consumption Model for TSCH Networks," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482–489, Feb 2014.