Enhanced Energy-Saving Mechanisms in TSCH Networks for the IIoT: the PRIL Approach

Stefano Scanzio ¹, Gianluca Cena ², and Adriano Valenzano ²

¹CNR-IEIIT
²Affiliation not available

October 30, 2023

Abstract

Lifetime of motes in wireless sensor networks can be enlarged by decreasing the energy spent for communication. Approaches like time slotted channel hopping pursue this goal by performing frame exchanges according to a predefined schedule, which helps reducing the duty cycle. Unfortunately, whenever the receiving radio interface is active but nobody in the network is transmitting, idle listening occurs. If the traffic pattern is known in advance, as in the relevant case of periodic sensing, proactive reduction of idle listening (PRIL) noticeably lowers energy waste by disabling receivers when no frames are expected for them. Optimal PRIL operation demands that, at any time, the transmitter and receiver sides of a link have a coherent view of its state (either enabled or disabled). However, this is not ensured in the presence of acknowledgment frame losses.

This paper presents and analyzes some strategies to cope with such events. An extensive experimental campaign has been carried out through discrete event simulation to determine what consequences above errors may have from both a functional and performance viewpoint. Results show that, although no strategy is optimal in all circumstances, different solutions can be profitably adopted depending on the specific operating conditions.
Enhanced Energy-Saving Mechanisms in TSCH Networks for the IIoT: the PRIL Approach

Stefano Scanzio, Senior Member, IEEE, Gianluca Cena, Senior Member, IEEE, and Adriano Valenzano, Senior Member, IEEE

Abstract—Lifetime of motes in wireless sensor networks can be enlarged by decreasing the energy spent for communication. Approaches like time slotted channel hopping pursue this goal by performing frame exchanges according to a predefined schedule, which helps reducing the duty cycle. Unfortunately, whenever the receiving radio interface is active but nobody in the network is transmitting, idle listening occurs. If the traffic pattern is known in advance, as in the relevant case of periodic sensing, proactive reduction of idle listening (PRIL) noticeably lowers energy waste by disabling receivers when no frames are expected for them. Optimal PRIL operation demands that, at any time, the transmitter and receiver sides of a link have a coherent view of its state (either enabled or disabled). However, this is not ensured in the presence of acknowledgment frame losses.

This paper presents and analyzes some strategies to cope with such events. An extensive experimental campaign has been carried out through discrete event simulation to determine what consequences above errors may have from both a functional and performance viewpoint. Results show that, although no strategy is optimal in all circumstances, different solutions can be profitably adopted depending on the specific operating conditions.

Index Terms—Energy-saving mechanisms, green networking, ultra-low power, WSN, WSAN, IIoT, IEEE 802.15.4, TSCH, PRIL

I. INTRODUCTION

In the past decade wireless extensions were progressively integrated into the wired infrastructure of industrial distributed systems. This results in current heterogeneous networks, which rely on different transmission technologies [1]. In particular, wireless sensor (and actuator) networks (WSN/WSAN) are increasingly employed wherever the lack of a suitable communication infrastructure makes wireless meshing convenient. Besides the ability to satisfy the timing constraints imposed by applications (e.g., deadlines), ultra-low power consumption is often demanded. The ability not to waste energy is perhaps the primary requirement of wireless networks whose nodes are powered on batteries and must operate unattended for long time periods, often exceeding ten years.

WSN/WSAN are core technologies for the Industrial Internet of Things (IIoT) [2], [3]. Significant examples of how they can be exploited in such context are the retrofitting of industrial plants for monitoring and diagnostics [4], greenhouse automation [5], heating, ventilation, and air conditioning (HVAC) systems [6], smart and precision agriculture [7], petrochemical industry [8] and many others. Several commercial solutions are based on the well-know IEEE 802.15.4 standard [9], including its Time Slotted Channel Hopping (TSCH) operating mode. TSCH relies on scheduled transmissions and mechanisms to make the network nodes sleep when they are not involved in the communication, thus allowing tangible energy saving [10]. As a side effect, this also reduces the amount of exhausted batteries to be disposed of and improves sustainability. Popular examples of TSCH-based technologies that are customarily used in industrial environments are ANSI/ISA 100.11a and IEC WirelessHART [11], as well as IETF 6TiSCH [12].

In TSCH, scheduled transmissions between a pair of nodes (that is, opportunities for the sender node to use a reserved portion of the available bandwidth) repeat cyclically over time, with a period that is typically 2s or less. Receiver nodes are activated accordingly to read incoming frames. When a receiver is switched on but no transmissions take place, a condition commonly known as idle listening occurs, which causes non-negligible energy waste. However, if the traffic pattern generated by applications can be predicted to some extent, as in the case of periodic sampling performed at the IIoT perception layer, techniques known as Proactive Reduction of Idle Listening (PRIL) permit to limit this phenomenon dramatically, by temporarily disabling the receiver on a link when no frames are expected to be exchanged soon.

A preliminary PRIL scheme, called PRIL-F, was presented in [13], where the transmitter puts the receiver to sleep by including suitable commands along with data. Such mechanism proved to offer significant energy saving in specific network topologies often found in industrial control systems, for instance the single-hop star topology. In that work, the possibility that an acknowledgement (ACK) frame could be lost was not taken into account. However, such events are not unlikely to happen, and may cause the two sides of the link to have an inconsistent view of its state. For example, the receiver could be sleeping, but the transmitter is not aware of this and keeps on sending frames uselessly. Since the effect of sleep commands is temporary, the correct TSCH operations are not impaired. Nevertheless, negative consequences on energy saving capability are definitely expected.

This paper explicitly tackles ACK frame losses by proposing and comparing some strategies to deal with their occurrence. Besides, it also offers guidelines to select the most appropriate solution depending on the network operating conditions. Compared to [13], this work analyzes and properly solves the above problem by:

1) deriving the actual failure probability for ACK frames from experiments performed with real devices;
2) proposing a new, quite effective A-open strategy; and

Stefano Scanzio, Gianluca Cena, and Adriano Valenzano are with the National Research Council of Italy (CNR–IEIIT), Corso Duca degli Abruzzi 24, 1-10129 Torino, Italy (Corresponding author: Stefano Scanzio)
3) analyzing alternative strategies, like 2-open, to show their lack of effectiveness.

The structure of the paper is the following: Section II briefly reviews the literature about energy-saving techniques and summarizes some main characteristics of PRIL. Section III deals with the strategies mentioned above, while Sections IV and V present the setup adopted in the experiments and the obtained results, respectively. Finally, Section VI draws some concluding remarks.

II. ENERGY SAVING IN TSCH

Conserving energy is likely the main goal of WSN since their inception, and surely one of the main reasons behind TSCH. Many works appeared in the literature focus on energy consumption in TSCH. Some of them adopt clever scheduling algorithms to assign time slots and frequencies, so as to extend the global network lifetime [15], [16], [17]. When sporadic traffic is considered, other solutions apply specific scheduling algorithms to shared cells [18], that is, cells which can be used by more than one sender node. An improved node load balancing is used in [19] to prolong network lifetime. Instead, [20] aims at reducing energy consumption during network formation. Works in [21], [10] suggest practical guidelines and analyze how some communication parameter settings impact on performance indicators, including power consumption. The accuracy of synchronization among TSCH nodes is another aspect that affects performance and power consumption. This is why some works propose strategies to reduce [22] or dynamically adapt [23] the guard time (a TSCH feature intended to cope with non-perfect synchronization), hence saving energy. Authors of [24] have shown how a dynamic modification of the enhanced beacon periods can help in improving synchronization and connection time, with a reduction of energy consumption. Transmission channels can be selected on the fly by means of black and white listing techniques. A better quality of the communication support at the medium access control (MAC) layer results in improvements to energy consumption, reliability, and latency at the same time [25], [26].

PRIL techniques, on which this work relies, are typically orthogonal to those described above, and they are another appealing way to pursue the energy goal [13], [14]. As mentioned above, cells reserved to the link between a sender (NTX) and a receiver (NRX) represent transmission opportunities. Scheduled cells that are not actually used by NTX for frame exchanges cause NRX to listen uselessly. PRIL operates by switching NRX off when the scheduled cells are expected to remain unused, based on predictions about the NTX traffic.

TSCH nodes are time-synchronized, and time is divided into slots of fixed duration T_slot. In every slot several concurrent transmissions (up to the number N_slot of available channels) can be contextually performed by different senders. Usually, N_slot = 16 for the 2.4 GHz O-QPSK PHY defined by the IEEE 802.15.4 specification. Mapping between logical channels (channel offsets in the scheduling table) and physical channels (frequencies actually used for transmission) relies on a pseudo-random function. This technique, known as channel hopping, improves robustness against disturbance thanks to frequency diversity, and results in higher reliability.

Every time slot is identified by the Absolute Slot Number (ASN), a counter shared among nodes that is incremented by one on every slot. The protocol defines both shared cells, which can be used by several nodes to transmit data, and dedicated (non-shared) cells, which can be only exploited by a specific sender. This paper explicitly focuses on confirmed (unicast) transmissions in dedicated cells, which are by far the most interesting option for data exchanges (e.g., values cyclically sampled by sensors). Although PRIL could be extended to deal with multcast transmissions and shared cells as well, by permitting the transmitter to selectively switch off its associated receivers, doing so makes the mechanism more complex. For this reason these cases have not been considered here, and are left as future work.

As outlined in the example in the upper part of Fig. 1 a cell is defined in an N_ch × N_slot slotframe matrix that repeats over time every N_slot slots. Thus, every scheduled cell provides node NTX with a cyclic transmission opportunity that is made available with period T_sfr = N_slot · T_slot. The quantity T_sfr is typically known as slotframe interval. Within the same matrix, more than one such opportunities can be defined for the same pair of nodes. This is the case of the link from NTX to N_RX in the figure, which has two. The relative position of the current cell within the matrix (i.e., the column, which corresponds to the slot offset) is obtained by each node as ASN mod N_slot. Channel hopping does not directly impact on the definition and implementation of PRIL techniques, and so a compact representation has been used in the following for the matrix, as shown at the bottom of Fig. 1. The state of each scheduled cell is reported from the point of view of NTX, which is the only node directly affected by PRIL: either a frame is received in the cell (RX) or idle listening (IDLE) is experienced otherwise. The use of PRIL has no effect on the quality of clock synchronization among nodes, because its implementation does not require any modification to the TSCH synchronization protocol.

Let T_app be the mean time elapsing between two consecutive instances of a specific scheduled cell where a data frame is actually sent. In many industrial applications packet generation is periodic, in which case this time coincides with the sampling period T_app. In typical configurations T_app > T_sfr, and the energy wasted for idle listening can be significant because, not considering retries, only one cell every c · T_app/T_sfr is actually exploited for transmission (c is the link capacity, given by the number of cells in the matrix allocated to it).
Fig. 2. Effectiveness of PRIL vs. conventional TSCH (N_RX viewpoint).

Practical PRIL implementations rely on sleep commands embedded by N_TX inside IEEE 802.15.4 Information Elements (IE). On the reception of a sleep command the N_RX receiving circuitry is switched off for a time period encoded in the command itself. When the command effect ceases, the receiver automatically reverts to the conventional TSCH behavior. Computing the duration of the sleep period for nodes located one hop away from the source node is a trivial task when traffic is periodic (it is equal or directly related to T_app). PRIL-F [13] is based on this assumption and, to the best of our knowledge, it is the only PRIL implementation experimentally evaluated so far. Other methods to evaluate the sleeping time in more complex scenarios, e.g., by leveraging traffic inspection and/or machine learning to predict cell usage patterns, are still under development.

In this work we consider cyclic data transmissions between two nodes with period T_app, which is coherent with PRIL-F. We wish to derive generic strategies that apply to any PRIL technique, regardless of the way the sleeping time to be inserted in the sleep command is calculated. Assuming the traffic to be periodic is not a big limitation, as most industrial applications that rely on WSN (and even home automation) work this way. Instead, restricting the analysis to a single hop has to do specifically with PRIL-F: although it can be used also in multi-hop networks, power saving is only achieved on nodes located one hop away from the traffic source. This approach has been proved able to lower power consumption sensibly, without affecting communication latency and reliability [13], and consequently it can be profitably adopted in any context. PRIL strategies capable to reduce power consumption on nodes located more than one hop away from the traffic source are outside the scope of this paper and are left for future work. It is worth remarking that dynamic changes brought by RPL and/or machine learning to predict cell usage patterns, are still more complex scenarios, e.g., by leveraging traffic inspection and/or machine learning to predict cell usage patterns, are still under development.

In this work we consider cyclic data transmissions between two nodes with period T_app, which is coherent with PRIL-F. We wish to derive generic strategies that apply to any PRIL technique, regardless of the way the sleeping time to be inserted in the sleep command is calculated. Assuming the traffic to be periodic is not a big limitation, as most industrial applications that rely on WSN (and even home automation) work this way. Instead, restricting the analysis to a single hop has to do specifically with PRIL-F: although it can be used also in multi-hop networks, power saving is only achieved on nodes located one hop away from the traffic source. This approach has been proved able to lower power consumption sensibly, without affecting communication latency and reliability [13], and consequently it can be profitably adopted in any context. PRIL strategies capable to reduce power consumption on nodes located more than one hop away from the traffic source are outside the scope of this paper and are left for future work. It is worth remarking that dynamic changes brought by RPL to the network topology do not affect the correct operation of the network permanently, and not even of PRIL (when moved to a new position, sleeping links are awakened automatically).

Fig. 2 shows a simple example of PRIL operation. When a frame reaches N_RX (downward blue arrow in the figure), the receiver is deactivated for a suitable time to prevent idle listening. In practice, the content of the sleep command is the number of scheduled cells that have to be “disabled” after command reception. When a data frame gets lost, for instance because of electromagnetic noise, some cells are used for re-transmissions, and the sleeping time encoded in every retry is shortened accordingly. The loss of an ACK frame is not considered in the example in Fig. 2. Unfortunately, the analysis performed in Subsection IV-A on data acquired in real networks shows that such a kind of event is not so rare, and the probability to lose ACK frames is not much smaller than that for data frames. In the next section some techniques are presented that are explicitly aimed to deal with ACK losses.

III. ROBUSTNESS AGAINST ACK LOSSES

Let N_TX \xrightarrow{x} N_RX denote a reserved cell devoted to transmissions between nodes N_TX and N_RX. Notation N_TX \xrightarrow{x} N_RX refers to the x-th occurrence of that cell, where x is a sequence number that identifies the specific cell instance. Thus, N_TX \xrightarrow{x+1} N_RX represents the following occurrence. In the case only one scheduled cell is allocated to the pair of nodes, the next occurrence is characterized by the same slot offset but is found in the next slotframe. As said before, not all the scheduled cells are used for transmission, and the fraction of them which are actually exploited depends on the value of T_app. A reserved cell where the receiving interface of N_RX is switched off by PRIL is denoted N_TX \xrightarrow{x} N_RX.

When a transmission takes place in N_TX \xrightarrow{x} N_RX, N_RX notifies N_TX of the correct reception of the (data) frame by means of the corresponding acknowledgement. In our analysis we associate events F and A to the transmission attempt for the frame and its acknowledgement, respectively. With a little abuse of notation, we will use the same symbols F and A to denote both the specific data/acknowledgement frame and the associated transmission outcome, when no ambiguity occurs. We adopt the following convention: when the frame is successfully received by N_RX then F = 1. In this case, the related acknowledgement is either returned correctly (A = 1) or lost (A = 0). Instead, F = 0 means that the frame has not been received at all by N_RX.

It follows that N_TX is sure that the frame has been correctly received only when A = 1. By contrast, event combinations F = 0 and A = 1 \land A = 0 are equivalent from the N_TX viewpoint, despite they represent very different conditions for N_RX. In particular, in both cases N_TX has to retransmit the same frame in the next cell (F_{x+1}), irrespective of whether it has already been delivered correctly to its destination.

The sleep command is modeled as an attribute .s of the data frame (F_s). For example, F_{s=4} means that the frame embeds a sleep command whose duration is equal to 4. It is intended to disable the next 4 cells on the link (N_TX \xrightarrow{x} N_RX), so that transmission between N_TX and N_RX is re-enabled in N_TX \xrightarrow{x+5} N_RX. Using relative times in sleep commands simplifies the description of our techniques, however all proposed strategies can be easily implemented by using absolute times.

A. Closed strategy

The closed strategy corresponds to the PRIL operation described in [13]. Examples of its behavior are sketched in the upper part of Fig. 3. In particular, in Fig. 3c a frame that conveys a sleep command for deactivating the next t cells is assumed to be delivered correctly in cell x (F_{s=4} = 1), which causes the receiver to switch off cells N_TX \xrightarrow{x+5} N_RX. However, that transmission attempt suffers from an ACK loss (A = 0). Sender N_TX, which misses the confirmation,
retransmits the data frame, and keeps doing so since the $N_{\text{RX}}$ receiving interface is off. This condition, where the sender repeatedly fails to deliver frames because communication is disabled on the receiver, is denoted *useless talking*.

Let $N_{\text{tries}}$ be the maximum number of transmission attempts the MAC is allowed to perform for a frame (retry limit plus one). When $t \geq N_{\text{tries}} - 1$ all retries will unavoidably fail, otherwise the link $N_{\text{TX}} \rightarrow N_{\text{RX}}$ is re-enabled on $N_{\text{RX}}$ after staying “closed” for $t$ reserved cells. In both cases the loss of $A_x$ results in an energy waste on $N_{\text{TX}}$ caused by its useless talking that is proportional to $\min(N_{\text{tries}} - 1, t)$. On the receiver side the energy saved by PRIL is optimal, since the $N_{\text{RX}}$ interface is switched off as soon as $F_x = 0$ is received. Every failed attempt possibly occurring at the beginning of any frame transmission ($F_x = 0$) is re-enabled on $N_{\text{RX}}$ to avoid useless talking, because $N_{\text{RX}}$ energy waste in idle listening. Each failed early failures reduce the amount of useless talking, should they be followed by an ACK loss, because the energy spent in the related $\varphi$ cells is ascribed to transmission errors.

### B. 1-open and n-open strategies

The 1-open strategy lessens the limits of the closed strategy by letting “open” the next reserved cell following the sleep command $F_x = 1$. In other words, when a correctly received frame ($F_x = 1$), $N_{\text{RX}}$ sets cell $N_{\text{TX}}$ to $N_{\text{RX}}$ as usable, thus deferring the shut-off of the receiving interface to cells $N_{\text{TX}}$. By doing so, if $A_x = 0$ and the retry limit has not been exceeded there is a chance for the next frame exchange to succeed ($F_x = 1$), so that $N_{\text{TX}}$ can stop unnecessary retransmissions.

In this case, when $F_x = 1$ the following reserved cell $N_{\text{TX}}$ is usable and can be actually exploited if either $A_x = 0$ or other frames addressed to $N_{\text{RX}}$ are queued in $N_{\text{TX}}$. The only drawback occurs when $A_x = 1$ and the $N_{\text{TX}}$ queue is empty, because cell $N_{\text{TX}}$ remains unused and $N_{\text{RX}}$ wastes energy in idle listening.

As in the closed strategy, when an ACK loss is followed by a frame loss in the next reserved cell ($A_x = 0 \land F_x = 0$), $N_{\text{TX}}$ keeps on transmitting and wastes energy in useless talking ($A_x = 0$, since $N_{\text{RX}}$ is unable to send confirmations and stop automatic retransmission. To mitigate this phenomenon, $n$-open strategies can be devised where, following a successful frame delivery, $n$ reserved cells are left open on $N_{\text{RX}}$. As in the closed strategy, $N_{\text{RX}}$ is unable to send confirmations and deal with possible retransmissions. Such strategies prevent the occurrence of conditions like the one depicted in Fig 3.i, and are able to cope with the loss of up to $n - 1$ data frames following an acknowledgement loss ($A_x = 0 \land F_x = 0$).

The probability to exceed the $n$-open recovery capabilities, which depends on the failure probability for $A$ and $F$ ($\epsilon_a$ and $\epsilon_f$, respectively) is typically small and equal to $\epsilon_a \cdot \epsilon_f^n$. For example, when $\epsilon_a = 8.0$ % and $\epsilon_f = 12.6$ % this probability is $13.0$% for $n = 2$, and drops to $0.016$ % for $n = 3$. The cost for preventing these unlikely events is to accept idle listening in $n$ cells on every successful transmission. From a practical viewpoint and considering the typical loss probabilities, setting $n > 1$ is not convenient because of the increased energy waste for idle listening. This aspect has been confirmed by an experimental evaluation of the $2$-open strategy.

### C. A-open strategy

The active-open ($A$-open) strategy is an enhancement of 1-open that can be implemented only in devices capable to perform clear channel assessment (CCA), for instance by leveraging energy detection. Similarly to 1-open, sleep commands in $A$-open turn off reception on $N_{\text{RX}}$ starting from cell $N_{\text{TX}} \rightarrow N_{\text{RX}}$, while leaving it enabled in cell $N_{\text{TX}}$. However, if the CCA mechanism on $N_{\text{RX}}$ detects activity on air in cell $N_{\text{TX}}$, the next reserved cell ($N_{\text{TX}} \rightarrow N_{\text{RX}}$) remains enabled and the reception shut-off is postponed to $N_{\text{TX}} \rightarrow N_{\text{RX}}$. This action is repeated until CCA detects a clear cell (e.g., a cell without any evidence of ongoing transmission). Typically, the energy spent to perform CCA resembles what is wasted for idle listening. In the worst case (when disturbance is repeatedly heard in subsequent scheduled cells), the same power consumption as conventional TSCH is experienced on $N_{\text{RX}}$.

The example in Fig 3.i shows that, as in the 1-open approach in Fig 3.i, when $F_x = 1$ the cell $x = 1$ is left open on $N_{\text{RX}}$ and can be used by $N_{\text{TX}}$ for transmitting a frame. In the $A$-open case, however, if $A_x = 0$ and $F_x = 0$ the detection of activity on air by CCA prevents $N_{\text{RX}}$ from disabling reception in cell $x = 2$, so that $N_{\text{RX}}$ can hear the
Sender: $N_TX$
1: if NOT is_empty_buf() AND cell_x!=OFF:
2: $F_x$=dequeue_buf(); $m$=$N_{rxs}$-1
3: $t$=get_next_t()
4: send($F_x$)
5: while recv($A_x$)=0 AND $m$>0:
6: \( x++\), $m$-- $t$--
7: if $t$=0: send($F_x$)
8: else: send($F_x$)
9: if $A_x$=1:
10: cell_{x+1,x+t}=OFF
11: if $mode$=closed: cell_{x+1}=ON

Receiver: $N_{RX}$
12: if cell_x=OFF:
13: if recv($F_x$ OR $F_x$=1):
14: send($A_x$)
15: if $F_x$=1: # Only frame with sleep cmd
16: cell_{x+1,x+t}=OFF
17: if $mode$=closed: cell_{x+1}=ON
18: if $mode$=$A$-open AND CCA_x=1: cell_{x+1}=ON

Fig. 4. Pseudo-code for closed, 1-open, and A-open strategies (i.e. modes). The pseudo-code of $N_{RX}$ is repeated for every single scheduled cell $N_{TX}$ $\rightarrow$ $N_{RX}$, while that of $N_{TX}$ also encompasses all the subsequent scheduled cells possibly used for retries upon transmission errors.

second retransmission. The most prominent advantage of this strategy is the total absence of useless talking, though the number of idle listening events increases as in the 1-open technique.

D. PRIL operation

In Fig. 4 the operation of three PRIL strategies is described using pseudo-code, separately for $N_{TX}$ and $N_{RX}$. Variable mode can assume one of the values closed, 1-open, and A-open, and permits to differentiate the behavior of the strategies. For the sender $N_{TX}$, in correspondence to the scheduled cell $N_{TX} \rightarrow N_{RX}$, when there is at least one pending frame and the cell was not switched off by PRIL (line 1) a frame is extracted from the queue and the number of available tries $m$ is initialized. Function get_next_t() in line 3 predicts how many $N_{TX} \rightarrow N_{RX}$ subsequent scheduled cells can be switched off because not needed for transmission, and returns the value $t$ to be included in the sleep command. Its actual implementation is out of the scope of this paper. After the initial transmission of frame $F_x$=1 in line 4, the frame is retransmitted until either the related $A_x$ returns to $N_{TX}$ or the number of allowed retries is over. When $t$ reaches 0 the sleep command is no longer included in the frame. In sophisticated implementations, the value $t$ may change also between subsequent retries. In these cases, re-evaluation of $t$ has to be performed inside the while statement. If $A_x$ notified the correct reception of the frame (line 9), the following $t$ cells related to $N_{TX} \rightarrow N_{RX}$ are switched off. In lines 10 and 16, if $t$<1 the range $x+1, ..., x+t$ identifies the empty set, and the command cell_{x+1,...,x+t}=OFF has no effect. Only for 1-open and A-open the next cell $N_{TX} \rightarrow N_{RX}$ remains usable (line 11).

Regarding the receiver $N_{RX}$, the pseudo-code reported in the figure is repeated for every scheduled cell $N_{TX} \rightarrow N_{RX}$ that was not switched off (line 12). After the correct reception of a frame, both those which include a sleep command and those which do not (line 13), the receiver sends an ACK frame back to $N_{TX}$. If the received frame contains a sleep command (test $F_x$=1 in line 15) PRIL switches off the following $t$ cells (line 16). In the 1-open and A-open case the next $N_{TX} \rightarrow N_{RX}$ cell is left usable. Only in the A-open case, if the channel is sensed busy (CCA_x=1) the next cell $N_{TX} \rightarrow N_{RX}$ is left open (line 18). Extending the algorithm to $n$-open is simple: all it is needed is to change cell_{x+1}=ON with cell_{x+1,...,x+n}=ON (lines 11 and 17).

From the complexity viewpoint, strategies are comparable: in fact, their algorithmic implementations are quite similar. It is important to remark again that $A$-open can be used only in devices provided with CCA support.

IV. EXPERIMENTAL SETUP

A discrete event simulator has been used to evaluate and compare the energy-saving techniques from the performance point of view. The simulator, called TSCH-predictor, is based on the SimPy framework. With respect to other tools like TSCH-Sim [27] and 6TiSCH [28], it benefits from a much simpler implementation that enables a quick and easy behavioral assessment of new algorithms proposed for TSCH. Regarding the features of TSCH-predictor that are of interest for this work, packet losses are modelled as Bernoulli trials with probabilities $\epsilon_f$ and $\epsilon_s$. Different probabilities can be defined for uplinks and downlinks and, if needed, for every single channel. Statistics it provides about latency were cross-validated with measures obtained on real devices in four experimental conditions. The discrepancy between real and simulated behavior, in terms of the average latency, was always below 6%. Instead, statistics about power consumption were derived by counting the number of specific events (idle listening, frame transmission/reception) and by multiplying them by the related energy consumption derived from measurements obtained on real devices. More information about features and limitations of TSCH-predictor can be found in [13].

For the simulation campaigns performed in this paper, the simulator was configured accordingly to data collected on a real setup. The values assigned to the main parameters are summarized in Table I. The power consumption model derives from experiments carried out with Open-MoteSTM devices equipped with an Atmel AT86RF231 radiochip and an STM32F103RB 32-bit microcontroller [29]. In particular, the energy required by $N_{TX}$ to transmit a 127 B data frame and receive the corresponding acknowledgement is about 485.7 $\mu$J, while the energy needed by $N_{RX}$ to receive such a frame and transmit the acknowledgement is about 651.0 $\mu$J. Finally, the energy spent for idle listening is about 303.3 $\mu$J.

A. Estimation of ACK loss probability

Properly estimating the loss probability $\epsilon_a$ for ACK frames plays an important role in this work. Hence, its evaluation relied on measurements performed on a real setup (the same used in [10]). The testbed consists of OpenMote B devices running the OpenWSN operating system and the 6TiSCH protocol stack. A Wi-Fi traffic pattern that mimics the variability of real traffic was purposely injected into the system to model environments suffering from non-negligible interference. This
simple expedient increased the frame loss probability and helped obtaining more realistic figures. Generation of long traffic bursts was purposely avoided in interferers to make TSCH transmission attempts (for both data and ACK frames) as independent as possible. The loss probabilities for ACK and data frames ($\epsilon_a$ and $\epsilon_f$) were obtained from the Default condition of [10], whose log is the longest one among those available in that report (15 days). In particular, the ping utility was used to repeatedly issue ICMP echo commands with period 120 s. ICMP echo request packets were sent from the root node ($N_{RX}$) to the leaf node ($N_{TX}$), while ICMP echo replies flowed in the opposite direction ($N_{RX} \rightarrow N_{TX}$).

In the following we assume that the failure probability $\epsilon_f$ is the same in the two directions $N_{TX} \rightarrow N_{RX}$ and $N_{RX} \rightarrow N_{TX}$, and the same holds for $\epsilon_a$. However, $\epsilon_f$ and $\epsilon_a$ may in general differ. In the considered conditions the loss probability for data frames, computed using the method in [10], is $\epsilon_f = 12.6\%$. Instead, the evaluation of $\epsilon_a$ is not so straightforward. The probability $P_{ACK}$ that, at the end of a frame transmission including up to $(N_{tries} - 1)$ retries, the ACK frame is correctly received by $N_{TX}$, under the assumption that spectrum conditions in the related cells are independent, is

$$P_{ACK} = 1 - \epsilon_f + (1 - \epsilon_f)\epsilon_a^{N_{tries}}. \tag{1}$$

Let us consider a single data frame exchange performed by means of confirmed transmission services. The expected number $N_{txas}$ of ACK frames successfully received by $N_{TX}$ is equal to $P_{ACK}$. In fact, whatever the sequence of attempts, the transmission procedure is terminated as soon as an ACK frame is correctly delivered to its destination. Moreover, when the retry limit is exceeded, no ACK frame is returned to $N_{TX}$ and the overall probability of this event is $1 - P_{ACK}$. Thus, $N_{txas} = N_{txa} \cdot P_{ACK} + 0 \cdot (1 - P_{ACK}) = P_{ACK}$.

The average number $N_{txas}$ of ACK frames that are lost by $N_{TX}$ can be computed as $N_{txas} = N_{txa} - N_{txas}$, where $N_{txa}$ is the average number of ACK frames sent by $N_{RX}$. $N_{txas}$ is equal to the average number $N_{rfx}$ of data frames received correctly by $N_{TX}$. It follows that the loss probability for ACK frames $\epsilon_a$, computed as the ratio $N_{txas} / N_{txa}$, is

$$\epsilon_a = \frac{N_{txas}}{N_{txa}} = 1 - \frac{N_{txas}}{N_{txa}} = 1 - \frac{P_{ACK}}{N_{rfx}}. \tag{2}$$

In the current OpenWSN implementation, IEEE 802.15.4 MAC layer sequence numbers are not exploited to discard duplicated frames. For this reason, every time anACK frame is lost the retransmitted frame is interpreted by the receiver as a new packet (in our case, an ICMP request or reply). However, thanks to the sequence number field included in the ICMP message header, the ICMP requestor (placed above the MAC layer in $N_{TX}$) is able to detect duplicate ICMP replies. They are labelled with the string “DUP!” in the ping log.

Every ICMP echo request sent by $N_{TX}$ is received by $N_{RX}$ on average, for every original ICMP echo request (accounting for possible losses and duplicates). Replies, in turn, may be lost or duplicated. By assuming that transmissions concerning requests and replies are statistically independent, the average number of replies received by $N_{TX}$ for every ping request it issues is equal to $N_{rfx}$.

The fraction of duplicate ICMP echo replies with respect to all the ICMP echo requests is $\alpha_{dup} = N_{rfx}^2 - 1$. This value can be estimated from the experimental logs collected for $N_{TX}$ as $\tilde{\alpha}_{dup} = N_{DUP} / N_{ping}$, where $N_{DUP}$ is the number of log lines that are marked “DUP!”, while $N_{ping}$ is the overall number of issued ping requests. Then $N_{rfx}$ can be computed as

$$N_{rfx} \approx \sqrt{\tilde{\alpha}_{dup} + 1}. \tag{3}$$

$\epsilon_a$ can be evaluated recursively by substituting (1) and (3) in (2). The starting value of $\epsilon_a$ can be set equal to $\epsilon_f$. We have observed that less than 10 interactions are enough to provide adequate accuracy. Using the values $\epsilon_f = 12.6\%$, $N_{DUP} = 10800$, obtained from the experimental logs, the ACK loss probability, adopted in the remaining part of this work, is $\epsilon_a = 8.0255\%$, which can be safely rounded to $\epsilon_a = 8.0\%$.

V. RESULTS

Reducing power consumption is the main goal of PRIL, and the experimental campaign in Subsection VA compares the different strategies focusing on this aspect. Instead, the impact of the closed, 1-open, 2-open, and A-open strategies on latency is taken into account in Subsection VB. Although responsiveness is not the primary goal of PRIL, the closer the latency is to conventional TSCH, the better the strategy. This is particularly relevant when WSNs are deployed in industrial scenarios, where timings are often important. Finally, Subsection VC provides a short recap of differences and similarities among strategies from several points of view. To make comparison more valuable, we also included results about conventional TSCH (that is, without the adoption of any PRIL technique), which can be considered a sort of baseline. Clearly, PRIL should be able to reduce power consumption of TSCH significantly in order to be attractive.

All experiments described in the following sections were carried out with two nodes ($N_{TX}$ and $N_{RX}$) and a cyclic packet flow with period $T_{app}$. The simulated duration was set to 10 years for all campaigns, and every experiment included at least 5256000 packets (when $T_{app} = 60$ s).

A. Power consumption

The $T_{app} = 60$ s period, corresponding to 3000 slots, has been selected to prevent queuing phenomena in $N_{TX}$. We have

| Quantity | Description | Typical value |
|----------|-------------|---------------|
| $T_{slot}$ | Duration of a slot | 20 ms |
| $N_{slot}$ | Number of slots in a slotframe | 101 |
| $N_{tries}$ | Maximum number of allowed tries | 16 |
| $\epsilon_f$ | Frame loss probability | 12.6 % |
| $\epsilon_a$ | ACK loss probability | 8.0 % |

1Experimental data used to compute the loss probability values are included in the file default-101-16-15days.dat, which can be downloaded from https://dx.doi.org/10.21227/fg62-bp39.
also carried out additional experiments aimed at analyzing the effect of queuing as shown in the following subsections. Results obtained in experiments focusing on power consumption are summarized in Table II. In particular, $P$ is the total power consumption, $P^\text{NTX}$ and $P^\text{NRX}$ represent the power consumption on $N_{\text{TX}}$ and $N_{\text{RX}}$, respectively, while $P^\text{listen}$ and $P^\text{rx}$ refer to the specific contributions of idle listening. For clarity, they are also plotted in Fig. 5 where $P$ is expressed as $P^\text{NTX} + P^\text{NRX} + P^\text{listen} + P^\text{rx}$, and concerns frame receptions.

The operating condition labelled default in the table refers to the “typical” parameter settings we considered in this paper, where the values of $\epsilon_f$, $\epsilon_a$, and $N_{\text{tries}}$ are fixed. For example, $I$-open outperformed 2-open in all considered conditions. In fact, the penalty paid for leaving two cells “open” is higher than the benefit achieved by a better ability to prevent useless talking. An interesting outcome is that $A$-open always outperformed $I$-open. This result is not surprising, as $A$-open takes advantage from the ability of the wireless adapter to perform CCA. The other side of the coin is that its implementation is more complex than $I$-open on some devices.

In the default condition $A$-open outperforms all other strategies, and the power saved with respect to $I$-open and closed is 0.90 and 2.67 $\mu$W, respectively, corresponding to energy consumption reductions equal to 3.2% and 9.3%. A significant difference between X-open (n-open and A-open) and closed strategies is that the former are optimal for the transmitter whereas the latter is the best for the receiver. This occurs because in the closed strategy $N_{\text{TX}}$ keeps transmitting frames (on $A_x$ = 0 events) that cannot reach $N_{\text{RX}}$ because the relevant cell is disabled, thus consuming a non-negligible amount of energy. On the contrary, in X-open strategies a redundant number of cells is left open, which increases the energy wasted by $N_{\text{RX}}$.

In the high error condition reported in Table II, both $\epsilon_f$ and $\epsilon_a$ were doubled, whereas they were halved in the low error case. It is worth noting that the closed strategy achieves best results in the low error condition ($\epsilon_f = 6.3\%$ and $\epsilon_a = 4.0\%$): this is because the useless talking phenomenon is unlikely to occur when transmission attempts fail seldom. In the two operating conditions at the bottom of Table II, $N_{\text{tries}}$ was increased to 32 (high tries) and reduced to 8 (low tries), respectively. For small $N_{\text{tries}}$ values the closed strategy is advantageous, because the amount of useless talking is limited by $N_{\text{tries}}$. By contrast, for large values of $N_{\text{tries}}$ the lowest energy consumption is achieved by the A-open strategy.

When either the disturbance level (high error) or the retry limit (high tries) is increased, the gap between closed and A-open, in terms of power consumption, also grows. Thus, two additional sets of experiments were performed to investigate in more details the impact of the environment (disturbance) and network configuration (retry limit) on power consumption. Results, summarized in Fig. 5, concern the first set of experiments, where the total power consumption $P$ was studied by varying the failure probability.

In the leftmost plot (Fig. 5a), $\epsilon_a$ is kept constant and equal to 8.0% while $\epsilon_f$ varies from 0 to 50%. Curves obtained for the four strategies show similar trends, and the consumption of closed progressively approaches A-open.

Differences in the curve shapes are evident in Fig. 5b, where $\epsilon_f = 12.6\%$ while $\epsilon_a$ varies from 0 to 33%. In this case,
the total power consumption for the closed strategy grows almost linearly with respect to $\epsilon_a$ because of the larger amount of useless talking, whereas it is less steep for X-open strategies. Finally, in Fig. 6 c $\epsilon_f$ and $\epsilon_a$ are varied jointly according to a proportional law. When the disturbance level is high A-open implies lower energy consumption compared to the other strategies. In particular, A-open is optimal when $\epsilon_f > 9\%$ and $\epsilon_a > 5.7\%$, otherwise closed is the best choice.

Fig. 7 shows the results obtained with the second set of experiments, where power consumption was analyzed by varying $N_{\text{tries}}$ from 1 to 36 while failure probabilities were left fixed ($\epsilon_f = 12.6\%$ and $\epsilon_a = 8.0\%$). The three plots show the overall power consumption $P$, as well as the contributions on the sender ($P_{\text{TX}}^{\text{TX}}$) and receiver ($P_{\text{RX}}^{\text{RX}}$) side. Once again, A-open outperforms 1-open with respect to every metric ($P$, $P_{\text{TX}}^{\text{TX}}$, and $P_{\text{RX}}^{\text{RX}}$), while 1-open is better than 2-open. Fig. 7 a shows that, as long as $N_{\text{tries}} \leq 12$, the closed strategy is significantly less demanding than A-open. When $N_{\text{tries}} \geq 3$ power consumption is almost independent from $N_{\text{tries}}$ for both A-open and 2-open, while for the other two strategies it increases linearly with different slopes (closed is the steepest).

It is worth observing that all strategies exhibit a peak when $N_{\text{tries}} < 3$ because of the $P_{\text{RX}}^{\text{RX}}$ contribution (see Fig. 7 c). This behavior can be easily explained, since reducing the number of available retries increases the probability that all attempts for the same packet fail, thus preventing the sleep command from reaching $N_{\text{RX}}$ and turning off the corresponding cells. Thus, the energy wasted for idle listening grows higher. Fig. 7 c shows that the best strategy for $N_{\text{RX}}$ is closed, because it switches all the relevant cells off as soon as the sleep command is received. The other strategies leave one or more cells open after the reception of the command, and this contributes to energy waste.

Power consumption on $N_{\text{TX}}$ is reported in Fig. 7 b. With the exception of A-open, it increases linearly with $N_{\text{tries}}$ for all strategies because of the useless talking phenomenon (when $N_{\text{TX}}$ and $N_{\text{RX}}$ have different views of the link state). Power consumption is independent from the number of available tries when $N_{\text{tries}} \geq 30$. This is due to the value we selected for $T_{\text{app}}$ in the experiments, which is roughly equivalent to 30 cells. In fact, the sleep command is configured to disable cells for a time interval equal to $T_{\text{app}}$, thus preventing useless talking from increasing further.

B. Latency

Latency is a key performance indicator for many industrial applications. For this reason a second experimental campaign was performed with the twofold goal of evaluating latency as well as the effect of a reduction of $T_{\text{app}}$ on both latency and power consumption. Four operating conditions have been considered, where $T_{\text{app}}$ was set to 60 s, 30 s, 10 s, and 5 s, respectively, while the remaining parameters were the same as in the default case described above. Latency is defined as the time, as seen by the application, taken by a packet to travel from $N_{\text{TX}}$ to $N_{\text{RX}}$. In our case it includes the time spent by the packet in the queue of the sending node, the time to manage retransmissions, and a uniformly distributed waiting delay. The waiting delay is due to the lack of synchronization between packet generation at the application level and the slotframe boundaries defined by the network. In long experimental runs, variations of this delay are also affected by the clock skew between the synchronized network time and the local clock.
(derived from a free-running counter driven by the quartz oscillator of the node). Other delays, such as propagation times over the air, are negligible when compared to the above contributions.

Besides power consumption, results reported in Table III and in the plot on the right in Fig. 8 show the main statistical indices related to latency, namely mean ($\mu_d$), standard deviation ($\sigma_d$), 99-$\mu$ (d$_{p99}$), 99.9-$\mu$ (d$_{p99.9}$) and 99.99-$\mu$ (d$_{p99.99}$), and maximum (d$_{max}$). Results labeled 60s in the upper rows of the table highlight that latency is mostly the same for all the strategies analyzed in this paper. When the generation period shrinks, as in the 30s case, latency for the closed and n-open strategies starts growing. In particular, the time between the generation of two subsequent packets is enough to prevent queuing phenomena in N$_{TX}$ only when $T_{app} \geq 60$s.

In the other experimental conditions (10s and 5s) latency increases for all strategies, including conventional TSCH. This is due to the larger average queuing time in N$_{TX}$. In the 5s case queuing effects are evident, and also TSCH and A-open suffer from a latency increase. In general, closed appears to be the slowest strategy, while A-open is optimal because its latency is the same as conventional TSCH. It is worth observing that, when the generation period is small enough, almost all scheduled cells are used to transmit/retransmit data frames, and the energy saved by PRIL is consequently lower.

### C. Optimal strategy

Simulations show that no strategy is optimal in every application context. For instance, A-open outperforms other

---

**Table III**

| Strategy | $T_{app}$ [s] | $p_{RX}$ | $p_{TX}$ | $|P|_{app}$ | $|P|_{n}$ | $|P|_{n}$ | $\mu_d$ | $\sigma_d$ | $d_{p99}$ | $d_{p99.9}$ | $d_{p99.99}$ | $d_{max}$ |
|----------|---------------|----------|----------|-------------|-------------|-------------|---------|-----------|----------|----------|----------|----------|
| TSCH     | 60            | 0.00     | 143.862  | 10.07      | 157.36      | 167.42      | 1.311   | 1.007    | 4.900    | 7.200    | 9.480    | 15.340   |
| closed   | 0.00          | 0.00     | 18.88    | 12.41      | 31.29       | 1.311       | 1.007   | 4.900    | 7.200    | 9.480    | 15.340   |
| 1-open   | 0.00          | 5.000    | 11.17    | 18.35      | 29.52       | 1.311       | 1.007   | 4.900    | 7.200    | 9.480    | 15.340   |
| 2-open   | 0.00          | 10.052   | 12.07    | 23.57      | 28.62       | 1.311       | 1.007   | 4.900    | 7.200    | 9.480    | 15.340   |
| A-open   | 0.00          | 5.055    | 10.07    | 18.55      | 28.62       | 1.311       | 1.007   | 4.900    | 7.200    | 9.480    | 15.340   |
| TSCH     | 30            | 0.00     | 137.576  | 20.13      | 164.56      | 184.70      | 1.311   | 1.007    | 4.900    | 7.200    | 9.460    | 15.520   |
| closed   | 0.00          | 0.000    | 37.52    | 26.64      | 64.16       | 1.480       | 1.160   | 5.500    | 7.680    | 9.460    | 15.520   |
| 1-open   | 0.00          | 10.000   | 22.33    | 36.94      | 59.26       | 1.334       | 1.030   | 5.020    | 7.300    | 9.500    | 15.980   |
| 2-open   | 0.00          | 20.192   | 20.39    | 47.17      | 67.56       | 1.314       | 1.010   | 4.940    | 7.220    | 9.420    | 15.940   |
| A-open   | 0.00          | 10.110   | 20.13    | 37.10      | 57.23       | 1.311       | 1.007   | 4.920    | 7.200    | 9.460    | 15.520   |
| TSCH     | 10            | 0.00     | 112.43   | 60.40      | 193.39      | 253.79      | 1.312   | 1.008    | 4.920    | 7.220    | 9.500    | 15.720   |
| closed   | 0.00          | 4.000    | 74.83    | 80.97      | 155.80      | 1.513       | 1.257   | 5.920    | 8.900    | 11.760   | 18.900   |
| 1-open   | 0.00          | 29.972   | 61.84    | 110.93     | 172.78      | 1.339       | 1.046   | 5.160    | 7.580    | 9.900    | 17.680   |
| 2-open   | 0.00          | 60.299   | 60.52    | 141.26     | 201.78      | 1.315       | 1.012   | 4.940    | 7.300    | 9.540    | 17.760   |
| A-open   | 0.00          | 30.277   | 60.40    | 111.24     | 171.64      | 1.312       | 1.008   | 4.920    | 7.220    | 9.500    | 15.720   |
| TSCH     | 5             | 0.00     | 74.711   | 120.80     | 236.63      | 357.44      | 1.384   | 1.091    | 5.360    | 7.880    | 10.680   | 18.360   |
| closed   | 0.00          | 0.000    | 130.37   | 161.92     | 292.29      | 1.584       | 1.323   | 6.120    | 9.360    | 12.220   | 20.820   |
| 1-open   | 0.00          | 52.345   | 121.19   | 214.26     | 335.46      | 1.395       | 1.110   | 5.460    | 7.960    | 10.860   | 19.120   |
| 2-open   | 0.00          | 74.711   | 120.81   | 236.63     | 357.44      | 1.384       | 1.091    | 5.360    | 7.880    | 10.680   | 18.360   |
| A-open   | 0.00          | 52.539   | 120.80   | 214.46     | 353.26      | 1.384       | 1.091    | 5.360    | 7.880    | 10.680   | 18.360   |

Fig. 8. Power consumption (dissected, on the left) and latency ($\mu_d \pm \sigma_d$ and $d_{p99}$, on the right) of PRIL strategies for different values of $T_{app}$. 

Simulations show that no strategy is optimal in every application context. For instance, A-open outperforms other
TABLE IV
SYNOPSIS ABOUT STRATEGIES (■ GOOD, □ ACCEPTABLE, □ SUBPAR).

| Context                        | closed | 1-open | A-open |
|--------------------------------|--------|--------|--------|
| Consumption on sender          | □      | ■      | ■      |
| Consumption on receiver        | ■      | □      | □      |
| High disturbance               | □      | ⊠      | ■      |
| Typical conditions             | □      | ⊠      | ■      |
| Low disturbance                | ■      | ⊠      | ⊠      |
| High retry limit               | □      | ⊠      | ■      |
| Low retry limit                | ■      | ⊠      | ⊠      |
| Transmission latency           | □      | ⊠      | ■      |

strategies in many cases, but in some circumstances the closed behavior is better. Moreover, as expected, n-open solutions do not provide any benefits in typical operating conditions when \( n > 1 \). Table [V] recaps how each strategy ranks in terms of power consumption in any given operating condition. The first two lines refer to the two sides of the link in typical conditions. The last line concerns instead latency. A strategy was identified as “good”, “acceptable”, or “subpar”, depending on how it compares to the other strategies.

The table highlights that A-open achieves the shortest latency and, most important, the lowest power consumption in critical operating conditions, e.g., when the disturbance level is particularly high or the retry limit is increased to improve reliability. If CCA detects activity in an otherwise unused scheduled cell because of external interference, the A-open strategy delays cell switching off to the next transmission opportunity, slightly increasing power consumption. This phenomenon was not modelled in the simulator because CCA is highly hardware-dependent (e.g., energy-based vs. carrier-based detection). On the other hand, closed ensures the best energy saving capability on the receiver side, and even for the whole link if the disturbance level and retry limit are low enough. Finally, 1-open can be profitably applied in the same conditions as A-open, when the node is unable to perform CCA.

VI. CONCLUSIONS

When dealing with proactive reduction of idle listening techniques in TSCH, the basic strategy, termed closed, mitigates the problem and enables a consistent amount of energy to be conserved on the receiving side of a link. However, it suffers from performance degradation when ACK frames get lost and useless talking is experienced on the transmitting side. The 1-open and A-open strategies presented in this paper have been conceived to deal with this drawback, and have been checked against the closed solution. Evaluation has been carried out by means of a discrete-event simulator developed ad hoc, and configured with parameters obtained from an experimental setup that includes real devices communicating over the air.

Results show that there is not a single winning strategy, as the behavior depends on specific operating conditions of the wireless network. Thus, we have identified the conditions under which every strategy offers the best performance, in terms of both power consumption and latency. This characterization shows that, in many realistic cases relevant for industrial contexts, A-open outperforms the other strategies.

We deem that our analysis satisfactorily addresses and solves the problem caused by ACK losses in PRIL. Therefore, future work will focus on new and more effective implementations of PRIL techniques, aimed at increasing the amount of saved energy by extending the currently available implementation (PRIL-F) to the multi-hop case.
