DataJoin: An Energy-Efficient Joining Scheme for 802.15.4e TSCH Networks

Conference Paper · October 2018

DOI: 10.1109/WiMOB.2018.8589149

4 authors:

Elodie Morin
University of Grenoble

Mickael Maman
Atomic Energy and Alternative Energies Commission

Roberto Guizzetti
STMicroelectronics

A. Duda
Grenoble Institute of Technology

DataJoin: An Energy-Efficient Joining Scheme for 802.15.4e TSCH Networks

Elodie Morin*§¶, Mickael Maman*, Roberto Guizzetti§, and Andrzej Duda¶
*CEA-LETI, MINATEC, Grenoble, France
§STMicroelectronics, Crolles, France
¶Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

***Abstract*—This paper considers the problem of a node joining an existing 802.15.4e TSCH network. We propose DataJoin, an energy-efficient scheme based on two ideas: i) it schedules the transmission of enhanced beacon frames in a smart way by taking advantage of the information on the common TSCH schedule and ii) it piggybacks the next beacon transmission times onto some acknowledgment frames to allow a joining node to overhear this information.**

We compare DataJoin with the Baseline scheme that corresponds to the state-of-the-art protocol for joining TSCH networks. Simulation comparisons show lower energy consumption by the joining node under DataJoin along with similar joining delays and average network energy consumption for both schemes.

I. INTRODUCTION

The main research effort in the development of protocols for the Internet of Things (IoT) or Wireless Sensor Networks (WSN) has concentrated on optimizing the operational phase of the network deployment: Medium Access Control (MAC) methods, routing (RPL), IP adaptation (6LoWPAN), synchronous schedule construction, or time synchronization. The bootstrapping phase in which a node wants to join an existing network and starts to operate according to the established configuration of the network, was often neglected and only becomes important when people start to deploy networks in real-world use cases. This phase is nevertheless crucial for the joining node and difficult to accomplish for several reasons:

- A node placed near an existing network needs to join the network in an autonomous way (plug&play operation).
- Nodes in IoT or WSN networks are usually synchronized for maximal energy efficiency (e.g., beacon-enabled 802.15.4 or Time Slotted Channel Hopping (TSCH) mode [1]) and operate at low duty cycles so they only wake up at some instants and communicate in an intermittent way.
- The joining node operates in a completely asynchronous way: it does not know the channel nor the instant at which other nodes will wake up and send beacons (frames allowing to learn the parameters of the networks and effectively join the network).
- The joining node may need to scan chosen channels during extensive periods of time to discover its neighbors. Hence, it aims at minimizing the scan duration to consume the minimal energy during this phase, and join the network as fast as possible.
- In the case of energy harvesting nodes, they cannot stay awaken during long periods scanning for beacons—they need to distribute the scanning periods over time and interleave them with the periods of energy harvesting.

There is extensive literature with theoretical work on neighbor discovery in the context of ad hoc, mobile, and sensor networks assuming a *time-slotted discovery model*[1]: time is divided into fixed-width slots and the roles of the nodes are symmetrical. When they become sufficiently close to each other, nodes need to discover each other in bounded time without prior synchronization information. The usual assumptions include a *duty cycle operation* to save energy: nodes sleep during most of the slots and wake up during a single active slot. In each slot, a node can either sleep or actively scan for a neighbor by transmitting a beacon at the beginning and end of the slot, and listening to (scanning for) beacons sent by a neighbor between its beacon transmissions [2]. In this model, the nodes successfully discover each other when two active slots in the wake up/scan schedule temporally overlap.

Kindt et al. [3] recently proposed a *continuous-time periodic-interval* discovery model and a joining scheme in which nodes have different roles (scanners or advertisers). The authors showed that with optimized periods and scan windows, the scheme outperforms all known time-slotted protocols such as DISCO [2], U-Connect [4], Searchlight [5], or Difference Codes [6]. In this paper, we adapt the continuous-time discovery model to 802.15.4e TSCH networks [1] and design a TSCH energy-efficient joining scheme.

In TSCH networks, nodes are synchronized and operate according to a *common shared schedule* that determines which device may transmit frames on which channel and during which timeslot. A node that wants to join the network needs to capture an Enhanced Beacon (EB) to get necessary network information and start to operate in a synchronized way with other nodes.

When nodes operate with small duty cycles, it may be beneficial for a node to loose association with the network and then rejoin the network when needed for a data exchange due to significant energy consumed in guard times and synchronization maintenance during the network operation [7]. In

[1] note that the model is proper to the discovery phase and is independent of TSCH despite some obvious similarities.

this case, rejoining may happen frequently, which required a lightweight and energy-efficient rejoining scheme.

In this paper, we propose DataJoin, an energy-efficient scheme based on two ideas: i) we schedule the transmission of Enhanced Beacon frames in a smart way by taking advantage of the information on the common TSCH schedule and ii) we piggyback the next beacon transmission times onto some acknowledgment frames to allow a joining node to overhear this information.

The proposed scheme brings two major contributions. First, we can reduce the energy consumed during the joining phase by piggybacking the time of the next EB onto ACK frames. When a joining node scans for a beacon and overhears such a frame, it can go to sleep and wake up just before a beacon thus reducing energy consumption. Note that our scheme does not require additional information in all ACK frames even though when nodes send the time information more often, the benefit is greater.

Second, since the sending instants of acknowledgment frames depend on the allocation of timeslots to data transmissions, we propose to smartly couple the beacon scheduling policy with a data scheduling protocol. In this way, nodes send data (hence acknowledgments) and beacons without collisions over multiple channels with sufficient periodicity to optimize the joining latency. We evaluate the proposed scheme with an enhanced 6TiSCH simulator to show good joining performance and lower energy consumption.

II. Background on Neighbor Discovery Schemes

In this section, we present the principles of neighbor discovery schemes. The problem of neighbor discovery was the subject of extensive research that mostly assumed the time-slotted discovery model with time divided into fixed-width slots. The goal was to find the best scheme in which two nodes successfully discover each other with two overlapping active slots in the shortest time.

Prior time-slotted neighbor discovery protocols fall into probabilistic and deterministic categories. In probabilistic protocols, nodes listen, transmit, or sleep with different probabilities, which leads to unpredictable rendezvous latencies and long tails of the discovery probability [8]. Deterministic protocols provide bounds on latency based on the Chinese Remainder Theorem (DISCO, U-Connect) [2], [4], quorum-based overlaps (Quorum, Searchlight, Hello) [9], [10], [11], [5], [12], or combinatorial techniques with Difference Codes [13], [6]. The comparisons of different time-slotted schemes by Kindt et al. [3] showed that the best scheme is Difference Codes [6] closely followed by G-Nihao [14]. However, the optimal difference codes resulting in the best performance [6] can only be found for a few values of target duty cycles so we proposed to take advantage of Singer Cyclic Difference Sets (S-CDS) for a joining scheme that results in low duty cycles and applied the proposed scheme to the joining problem in 802.15.4e TSCH networks [15].

Nihao [14] adopted a different model from time-slotted discovery by pointing out that efficient discovery needs to rely on the transmission of short beacons and the reduction of wakeup slots by sending more beacons, which in turn reduces the number of scan slots. In this way, Nihao differentiates the roles of slots: in a *transmit* slot, a node sends a short beacon at the beginning of the slot and in a *receive* slot, a listening node can receive the beacon. In the S-Nihao variant, a node transmits beacons in each slot with only one slot used by the joining node for listening. The worst case delay in this variant is just the whole slot duration. In the G-Nihao variant, nodes

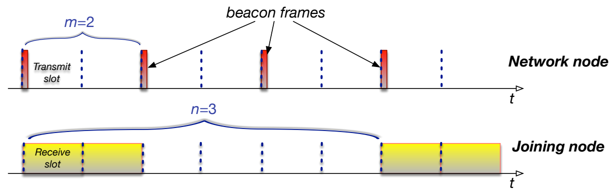


Fig. 1. Principle of the G-Nihao variant with parameters m and n.

send a beacon after m slots and listen during m slots repeating the period of m slots n times (see Fig. 1).

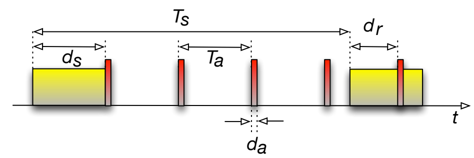


Fig. 2. Continuous-time discovery model.

Kindt et al. [3] presented the *continuous-time periodic-interval* discovery model in which nodes have different roles. Fig. 2 shows the main parameters of the continuous-time discovery model: an advertiser node sends beacons each T_a interval (advertising interval) and the joining node scans the channel each T_s interval (scan interval). The scan takes d_s time and beacons are d_a time long (we adopt the notation of Kindt et al. [3]). The successful discovery scan takes d_r time.

Kindt et al. proposed several variants of discovery protocols. The basic scheme called PI-0M considers the case of $T_a = d_s - d_a$, which corresponds to a straightforward strategy of the scanning period equal to the inter-beacon interval. Scanning interval T_s is chosen to obtain a given duty cycle. A bounded T_s value is needed to cover the possibility of a node not hearing network advertisers.

Note that the time-slotted model of Nihao is close to the continuous-time discovery one because it assumes short beacons at the beginning of the transmit slots, so we already adopt the representation of the continuous-time discovery model in Fig. 1 as introduced below.

III. Overview of IEEE 802.15.4e TSCH

In this section, we overview the operation and construction of TSCH networks: TSCH defines a *slotframe*, a collection of *timeslots* repeating in time. ASN (Absolute Slot Number) is the total number of timeslots elapsed since the start of the network or an arbitrary start time determined by the PAN coordinator. Nodes use a *hopping sequence*, an ordered list of physical channels, to periodically switch the communicating channel. The following relation gives the physical channel (CH) for a given timeslot:

$$CH = HS[(ASN + CO) \bmod seq\_length],$$

where HS is a hopping sequence list, CO is a channel offset, and seq_length, the length of the hopping sequence HS. In this paper, we assume $seq\_length = n_c$, where n_c is the number of available physical channels. The pair (t_s, CO), where t_s is a timeslot, defines a *cell*, the unit of allocation in TSCH. Each node follows a *schedule* that determines in which cell the node can transmit a frame, listen for a transmission from another node, or go to sleep (Section IV presents an example of a TSCH schedule). TSCH nodes enable discovery of the network by other nodes through Enhanced Beacons (EB) advertisement. EBs provide the necessary information to actually join the network (ASN, timeslot template, hopping sequence, slotframe, etc.).

The 802.15.4e standard neither specify the way data transmission slots are allocated, nor the strategy for sending beacons. Vogli et al. [16] proposed two advertisement policies in which nodes transmit EB in a randomly chosen cell. In Random Vertical filling (RV), nodes send EB in the first advertisement slot of a logical multislotframe structure while in the Random Horizontal filling (RH), nodes always use channel offset 0 and distribute EBs on randomly chosen available advertisement slots of the multislotframe structure. Both policies lead to possible beacon collisions when two nodes choose the same slot for sending beacons.

The IETF 6TiSCH group defined a minimal mode of operation for IPv6 connectivity over TSCH networks [17]. In particular, it defines parameters and procedures for a minimal mode of operation to build this kind of networks: in this mode, beacon scheduling is based on the G-Nihao approach, which implies a long scanning interval for joining nodes.

IV. DataJoin: Energy-Efficient Joining Scheme

We assume that a joining node knows the basic parameters of the TSCH network, in particular the longest value of advertisement interval T_a to derive d_s. The idea of the proposed scheme is to:

- adopt the Nihao/PI-0M strategy of scanning a given channel during at least advertisement interval T_a,
- choose T_s, the interval between scans, according to the maximum duty cycle tolerated by the joining node,
- reduce energy consumption of the joining node by overhearing regular data traffic during the scan to obtain the information about the expected time of the next EB on the channel,
- construct TSCH schedules in a way that assigns exclusive cells for EB frames and distributes the transmission of the EB expected time over some ACKs in a uniform manner in a slotframe.

A. Enhanced-Ack frames

To take advantage of data traffic for joining, we propose to add the information about the expected time of the next EB to Enh-Ack frames. IEEE 802.15.4e defines two types of Acks: Imm-Ack and Enh-Ack frames. Enh-Ack frames can contain several Information Elements (IE). In particular, IEEE 802.15.4e specifies an IE used in the Receiver Initiated Transmission (RIT) access method to notify the expected time of the rendezvous for the transmission of a payload frame. We use the RIT Information Element and its Rendezvous Time field (4 bytes) in the Enh-Ack frames to advertise the expected time of the next EB on the same physical channel (the field gives the expected time in units of 10 symbols). The advertised EB time is selected among the beacon frames sent by the neighbors of the Enh-Ack sender or the sender itself. When a joining node scans for a beacon and overhears such an Enh-Ack frame, it can go to sleep and wake up just before the beacon thus reducing the energy consumption. Fig. 3 presents the temporal principles of the proposed scheme.

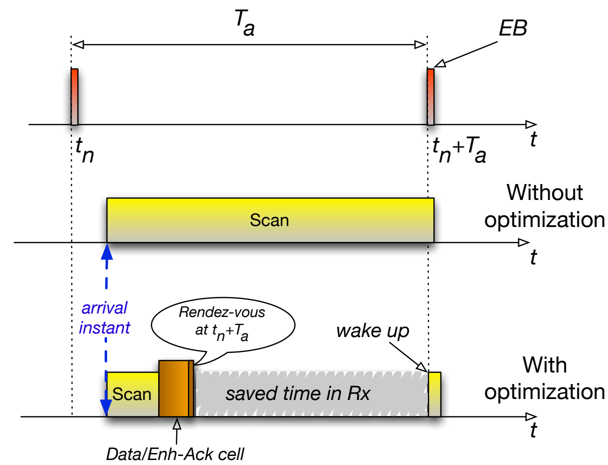


Fig. 3. Energy-efficient joining scheme: a joining node starts scanning and goes to sleep when it overhears the expected time of the next EB.

B. Enhanced Beacons Scheduling

Let $t_a \in [t_n; t_n + T_a]$ be the instant of the Enh-Ack transmission fixed for a given periodic TSCH schedule. The energy efficiency of our scheme come from the shorter radio reception (Rx) time between t_a and the next beacon frame: the shorter Rx time results in lower energy consumption. The reduction of the Rx time, denoted by Δt_{Rx}, corresponds to the probability of reducing the Rx time multiplied by the duration of the time reduction. Since the joining node wakes up randomly, its wake up instant is uniformly distributed between two beacon transmissions. Hence,

$$\Delta t_{Rx} = t_a(T_a - t_a).$$

Δt_{Rx} is maximized for $t_a = \frac{T_a}{2}$, which corresponds to a strategy for scheduling beacons that uniformly distributes EBs and Enh-Acks over timeslots.

Moreover, to obtain the best performance, the scheme must take advantage of channel diversity of TSCH, i.e., spread EBs over different frequency channels. The network needs to provide the joining node in a given neighborhood with the best temporal and channel arrangement of beacons and Enh-Acks. Below, we give the principles of the scheduling scheme and illustrate its operation with an example.

1) Temporal Aspects: A joining node wakes up and starts scanning. When it overhears the Enh-Ack frame with the IE element containing the expected time of the next EB, it goes to sleep and wakes up just before the beginning of EB transmission. To cope with clock drift, the node can wake up a bit earlier than the announced EB time. The duration of the guard interval directly depends on the time spent in sleep state and the hardware clock precision [7].

As Enh-Acks introduce a slight overhead compared to regular Acks, we can fix their percentage to limit the overhead. In the rest of the paper, we assume Enh-Acks for each data transmission, assuming the overhead in transmission is negligible compared to the gain in the time spent in reception (Rx) from the joining node point of view.

2) Channel Aspects: TSCH networks use frequency hopping to mitigate the effects of interference and multipath fading. Since the joining node does not know the channels actually used by the network, it randomly selects one channel and starts scanning it for duration d_s. If it does not receive EBs nor Enh-Acks after T_a, it considers the channel as unused and switches to another channel until it hears a transmission. To improve the performance, network nodes need to uniformly spread beacon transmissions on all channels.

As in the RV/RH filling of Vogli et al. [16], we adopt the notion of a *multislotframe* to guarantee at least one beacon on each channel within a given interval. A *multislotframe* contains n_s repetitions of a slotframe over time. We set the following rule for the multislotframe structure: for each repetition of the multislotframe, channel offsets need to map onto the same physical channels. We can satisfy this condition by selecting n_s as the least common multiple between the slotframe size and seq_length, the hopping sequence length (also equal to n_c in this paper).

To take into consideration the case of a joining node hearing only one advertising node, each advertising node sends beacons on every channel of the hopping sequence within a given multislotframe size.

3) Spatial Aspects: Since the joining node can receive frames only from a part of network nodes in a given neighborhood, an advertising node can schedule its beacons in a non-conflicting way with respect to its neighbors either by listening to their beacon slots, or by learning them from the common TSCH schedule. Nodes select their cells with beacons following their tree level to keep consistency in the network. Since the beaconing decisions only depend on the node schedules and what it locally hears, we do not use any negotiation between nodes—they locally make their decisions, which ensures that the scheme is extendable to large networks. The absence of collisions is guaranteed by using the schedule information along with the tree level from the routing protocol allocation.

C. DataJoin Allocation of Beacons and Enh-Acks

The rules for selecting a cell for beacons in DataJoin are the following:

1) We define n_s *EB-areas* in the multislotframe for placing EBs equidistant in time.
2) An advertising node transmits one EB per physical channel during a multislotframe.
3) Deterministic Phase: If the advertising node transmits Enh-Ack on a given physical channel, it selects the optimal *EB-area*: the chosen *EB-area* should respect the uniform distribution between Enh-Ack and EB frames on this physical channel.
4) Random Phase: If no Enh-Acks are transmitted on a physical channel, advertising nodes randomly select an *EB-area* for this physical channel.
5) When preparing the EB schedule, all advertising nodes first select their optimal slots for Enh-Ack transmissions during the deterministic phase (rule number 3). Then, they randomly place their remaining beacons slots (rule number 4).
6) Whenever the selected EB slots is already occupied by any transmission in the neighborhood, the advertising node will select the ASN that follows the ASN of the *EB-area* on the same physical channel.
7) A node can only allocate one activity (beacon or data exchange) to a given timeslot t_s. It cannot operate simultaneously on two different frequency channels.

With the information obtained from the shared TSCH schedule along with their neighbor lists, nodes can announce the time of the next EB transmitted by one of their neighbors on the same physical channel.

Example allocation. Fig. 4 presents the resulting schedule for an example topology of 60 nodes, the focus of the example is in the neighborhood of Nodes 29, 30, and 33 presented on the right part of the figure and colored in a way to see their allocations in the schedule. The parameters are the following: 5 physical channels and a multislotframe of 30 slots. Fig. 4 shows the cells with their allocations (the physical channel numbers are given in the upper left corner of cells). Note that to see which cells are allocated on a given physical channel, you need to follow a sequence of diagonal cells (e.g., all cells marked with 5 for channel 5).

During the deterministic phase, advertising nodes select the optimal *EB-area* to be close to uniformly distributed active cells on every physical channel in the multislotframe. The beacon cells surrounded by red lines in Fig. 4 represent the allocation in this phase. We can observe that orange Node 30 allocates beacon cells on physical channels 3 and 2 (resp. cell (17,0) and (29,2)). Moreover, the schedule of orange Node 30 on physical channel 3 meets the requirement of Rule 3: Ack in cell (1, 1) and EB in cell (17, 0) are equidistant in the multislotframe. The nodes in the neighborhood select their beacon cells according to their schedule on this channel, for instance, green Node 29 chooses its beacon in cell (5, 2) on channel 3.

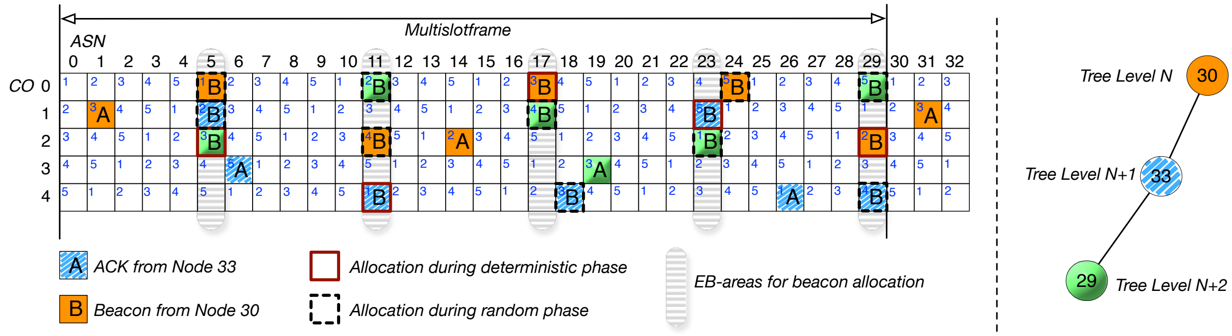


Fig. 4. Example of a TSCH schedule: 5 channel offsets (CO), 5 physical channels (given in cells), and a multislotframe of 30 slots.

After the deterministic phase, nodes randomly select their remaining beacon cells on a given channel within the remaining available *EB-areas*. Fig. 4 shows these allocations with cells surrounded by yellow lines. During this random phase, orange Node 30 randomly selects cells (5, 0) and (11, 2) respectively, to send EB on physical channel 1 and 4. The only remaining possibility to send a beacon on physical channel 5 is the *EB-area* at ASN 23. However, there is already an EB in this cell (from blue Node 33). In this case, the advertising node selects the next available cell on the same physical channel, which corresponds in this case to cell (24, 0).

In our scheme, nodes may announce the time of the next EB transmitted by one of their neighbors on the same physical channel. Even if the advertising node can be within the range of the joining node, the neighbor of the advertising node may be outside the range. In this case, the joining node will wake up for a beacon, but may not receive it. If this happens, the joining node will continue to scan until it receives either an EB or another Enh-Ack, which leads to the situation in which there is no gain in energy consumption.

To illustrate this case, let us assume a joining node located at the bottom-right corner of the topology in Fig. 4. It wakes up at ASN 24, starts scanning on physical channel 1, and can only hear blue 33 and green 29 Nodes. It receives Enh-Ack of blue Node 33 that notifies about the EB frame of orange Node 30 on physical channel 1 at ASN 35 (cell (5,0) in the next multislotframe, not shown in the figure). So, the joining node goes to sleep and wakes up at ASN 35. However, it does not receive the EB frame, since orange Node 30 is not in its communication range. It thus has to continue scanning until the reception of the EB frame in cell (11, 4) at ASN 41.

V. Evaluation

A. Evaluation setup

To evaluate the performance of the DataJoin scheme, we have run the discrete-time 6TiSCH simulator[2] enhanced with the Stripe protocol [18] for distributed scheduling of data transmissions. We use Stripe to generate the data schedule for two extreme data traffic cases: a high data rate of 1 packet per second per node and a low data rate with 1 packet per 1000 seconds. Knowing the data schedule, we select the cells for beacons according to the DataJoin scheme to obtain the global network schedule that feeds a discrete-event Python simulator to mimic the joining procedure of a node trying to join an operational TSCH network at random time offsets.

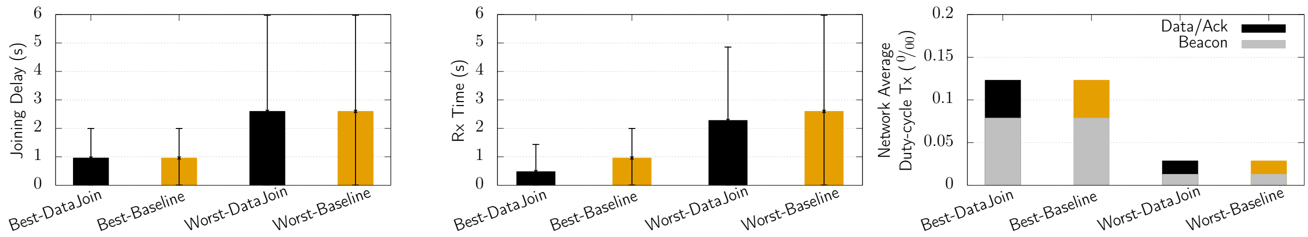
We compare DataJoin with the scheme called *Baseline* that corresponds to the RV beacon scheduling by Vogli et al. [16] slightly improved to make it directly comparable with DataJoin: i) we ensure beacons are on all physical channels (the original RV proposal actually selects cells based on channel offsets and not on physical channels) and ii) there is no beacon collision. In terms of energy consumption from the network point of view, scheduling on logical or physical channels is strictly equivalent, hence, we only adapt it to enable fair comparison with our proposal regarding the frame collision probability.

However, in terms of the joining node energy consumption, the Baseline scheme already outperforms RV since the joining node is certain to receive an EB frame on each channel in a shorter time since all nodes send beacons on all channels. Assuming that the multislotframe size and seq_length are co-prime, RV guarantees to join the network on a given channel during one multislotframe repetition. However, if they are not co-prime, a joining node might need to switch its listening channel because it may not receive any EB on the channel—it may not hear all nodes that send beacons, hence RV may be unsuccessful from the joining node point of view. Moreover, since we use a schedule that guarantees no collisions of beacons from neighbor nodes, the Baseline scheme already outperforms RV with respect to network energy consumption.

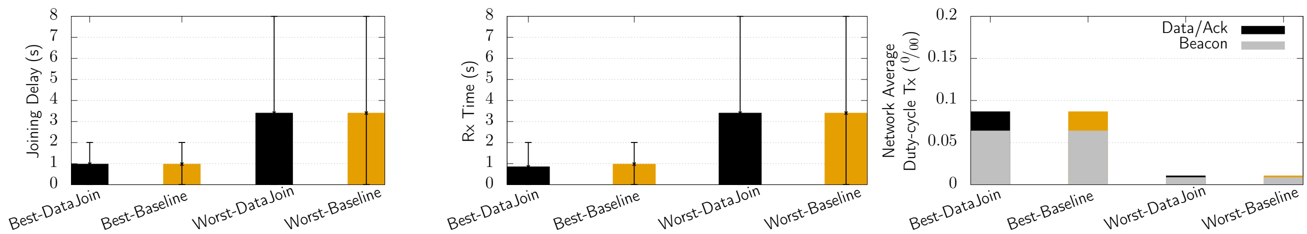
As in recent realistic energy consumption models [19], [20], [21], we use the time spent in reception (Rx) and transmission (Tx) as the metrics of energy consumption—we assume that the radio energy consumption varies linearly with the time the radio stays in the Rx or Tx states, the slope only being different for different hardware platforms. An eventual change of hardware would equally impact both schemes.

We simulate a random topology of 60 nodes and assume stable perfect links. For a given network schedule (placement of Data/Enh-Ack and EB in the multislotframe) and the con-

[2]https://bitbucket.org/6tisch/simulator/



(a) Performance for high rate traffic: 1 packet per second.



(b) Performance for low rate traffic: 1 packet per 1000 s

Fig. 5. Comparisons for two node positions (high and low node density) and for high and low data rates.

sidered joining strategy (DataJoin or Baseline), the simulator computes the Rx time and the joining delay experienced by a node waking up at a random instant (with time granularity of 0.1 ms). The results presented in figures correspond to the values averaged over all possible instants of the beginning of the joining procedure.

For a given random topology, we select two representative positions of the joining node: the "best" and the "worst" case for joining. The *best case* position corresponds to the situation in which the joining node hears a large number of advertising nodes (high node density). The *worst case* corresponds to low node density, e.g., next to leaf nodes.

B. Evaluation Results

Fig. 5 presents a performance comparison of the Baseline and DataJoin schemes for the best and worst position cases along with high and low traffic rates: joining delay d_r and the Rx time proportional to the energy consumption of the joining node. For the diagrams presented in Fig. 5, we have selected the value of $N_{adv} = 15$ for the best case and $N_{adv} = 2$ for the worst case.

Fig. 5 confirms that for a given network schedule, DataJoin and Baseline obtain similar results in terms of joining delay d_r (see left part of Fig. 5) and the average duty cycle in Tx from the network point of view (see right part of Fig. 5). The metrics are expected to be similar because both schemes use the same EB schedule and only the joining node strategy differs.

The central part of Fig. 5 shows that DataJoin spends less time in Rx, so consumes less energy, than Baseline in all configurations (the best or worst case, high or low data traffic). For instance, nodes under Best-DataJoin spend twice less time in Rx mode (0.5 vs. 1 s) than nodes under Best-Baseline for high data traffic (see Fig. 5a).

Note that the gain in the Rx time of DataJoin is higher in the best case than in the worse case because the best case corresponds to hearing more nodes, so the probability of overhearing an Enh-Ack is increased, which improves DataJoin performance.

Recall that we use the Rx and Tx time as the metrics of energy consumption, the simulations show that in all cases, DataJoin outperforms the Baseline scheme in terms of energy consumption from the joining node point of view for the similar network energy consumption and joining delay d_r.

To estimate energy consumption, we multiply the time spent in Rx by the power consumption of a radio platform in Rx mode. We use the value of the GREENNET platform [21] (19.26 mW in Rx mode [7]) to get the results presented in Fig. 6.

	Low Data Traffic		High Data Traffic	
	Best	Worst	Best	Worst
Joining delay d_r (s)	1	3.4	1	2.5
Rx Time (s)	0.9	3.3	0.5	2.2
Energy (mJ)	**17.3**	**63.6**	**9.6**	**42.4**

(a) DataJoin results

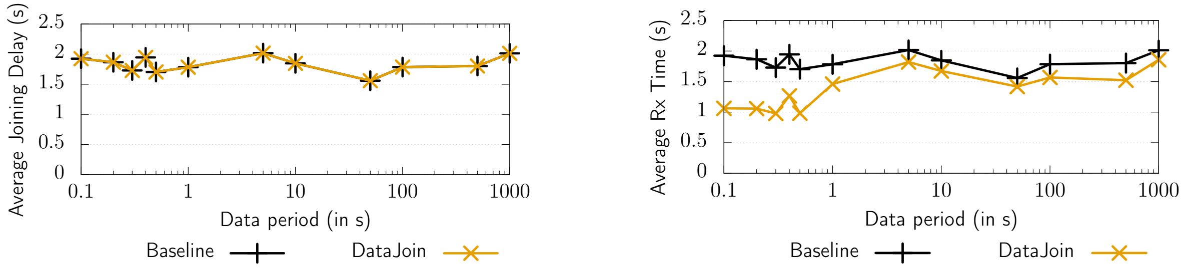
	Low Data Traffic		High Data Traffic	
	Best	Worst	Best	Worst
Joining delay d_r (s)	1	3.4	1	2.5
Rx Time (s)	1	3.3	1	2.5
Energy (mJ)	**19.26**	**63.6**	**19.26**	**48.1**

(b) Baseline results

Fig. 6. Summary of the results.

We observe that the performance gain of DataJoin increases with an increased rate of Data/Ack cells: for low data traffic (see Fig. 5b), Best-DataJoin only spends 10%-less time in Rx than for the Best-Baseline case (0.9 s vs. 1 s).

To confirm this trend, we have evaluated DataJoin for varying data rates in a random topology of 60 nodes. Fig. 7



(a) Joining delay d_r

(b) Rx time (energy consumption)

Fig. 7. Performance comparisons for an increasing data packet period in seconds.

presents the average values for 50 different random positions of the joining node for a given network schedule. We set the multislotframe size to 1000 slots and we assume a network that uses five channel offsets for five physical channels. Fig. 7 shows a shorter Rx time, hence lower energy consumption, for the DataJoin scheme and the same average joining delay. It also confirms the effect of Fig. 5: the gain of DataJoin is larger for the higher data rate.

VI. Conclusion

In this paper, we have proposed DataJoin, an energy-efficient scheme for joining a 802.15.4e TSCH network based on two ideas: i) we schedule the transmission of enhanced beacon frames in a smart way by taking advantage of the information on the common TSCH schedule and ii) we piggy-back the next beacon transmission times onto acknowledgment frames to allow a joining node to overhear this information. Thus, when scanning after a wake up, the joining node can receive an Enh-Ack and go to sleep until the next beacon to save energy.

The scheme takes advantage of the TSCH schedule properties so that nodes can send beacons without collisions over multiple channels with sufficient periodicity to limit the joining delay.

We have compared DataJoin with the Baseline scheme that corresponds to the state-of-the-art protocol for joining TSCH networks [16]. Simulation comparisons show lower energy consumption by the joining node under DataJoin along with similar joining delays and average network energy consumption for both schemes.

Acknowledgments

This work has been partially supported by the French Ministry of Research project LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

References

[1] IEEE, "IEEE Std 802.15.4-2015: IEEE Standard for Low-Rate Wireless Networks," 2015.

[2] P. Dutta and D. Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications," in *Proceedings of the 6th ACM SenSys*, 2008, pp. 71–84.

[3] P. H. Kindt, D. Yunge, G. Reinerth, and S. Chakraborty, "Griassdi: Mutually Assisted Slotless Neighbor Discovery," in *Proc. 16th ACM/IEEE IPSN*, 2017, pp. 93–104.

[4] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A Low-Latency Energy-Efficient Asynchronous Neighbor Discovery Protocol," in *Proc. 9th ACM/IEEE IPSN*, 2010, pp. 350–361.

[5] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't You Be My Neighbor?" in *Proc. 18th ACM MOBICOM*, 2012, pp. 185–196.

[6] T. Meng, F. Wu, and G. Chen, "On Designing Neighbor Discovery Protocols: a Code-Based Approach," in *Proc. IEEE INFOCOM*, 2014, pp. 1689–1697.

[7] E. Morin, M. Maman, R. Guizzetti, and A. Duda, "Comparison of the Device Lifetime in Wireless Networks for the Internet of Things," *IEEE Access*, vol. 5, pp. 7097–7114, 2017.

[8] M. J. McGlynn and S. A. Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks," in *Proc. 2nd ACM Mobihoc*, 2001, pp. 137–145.

[9] W.-S. Luk and T.-T. Wong, "Two New Quorum Based Algorithms for Distributed Mutual Exclusion," in *Proc. 17th ICDCS*. IEEE, 1997, pp. 100–106.

[10] Y. Tseng, C. Hsu, and T. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," in *Proc. IEEE INFOCOM*, 2002.

[11] J. Jiang, Y. Tseng, C. Hsu, and T. Lai, "Quorum-Based Asynchronous Power-Saving Protocols for IEEE 802.11 Ad Hoc Networks," *MONET*, vol. 10, no. 1-2, pp. 169–181, 2005.

[12] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: a Generic Flexible Protocol for Neighbor Discovery," in *Proc. IEEE INFOCOM*, 2014, pp. 540–548.

[13] R. Zheng, J. C. Hou, and L. Sha, "Optimal Block Design for Asynchronous Wake-Up Schedules and Its Applications in Multihop Wireless Networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 9, pp. 1228–1241, 2006.

[14] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk More Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks," in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.

[15] E. Morin, M. Maman, R. Guizzetti, and A. Duda, "Singer Cyclic Difference Sets for an Energy-Efficient Joining Scheme," in *IEEE WiMob*, 2017.

[16] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast Join and Synchronization Schema in the IEEE 802.15.4e MAC," in *IEEE WCNC Workshops*, 2015, pp. 85–90.

[17] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal 6TiSCH Configuration," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-minimal-21, Feb. 2017.

[18] I. Juc, O. Alphand, R. Guizzetti, M. Favre, and A. Duda, "Stripe: a Distributed Scheduling Protocol for 802.15.4e TSCH Networks," LIG, Research Report RR-LIG-54, HAL Open Archive, 2017. [Online]. Available: https://hal.inria.fr/hal-01585904

[19] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. Pister, "A Realistic Energy Consumption Model for TSCH Networks," *Sensors Journal, IEEE*, vol. 14, no. 2, pp. 482–489, 2014.

[20] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, 2015.

[21] L.-O. Varga, G. Romaniello, M. Vučinić, M. Favre, A. Banciu, R. Guizzetti, C. Planat, P. Urard, M. Heusse, F. Rousseau, O. Alphand, E. Dublé, and A. Duda, "GreenNet: an Energy Harvesting IP-enabled Wireless Sensor Network," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 412–426, 2015.