



An Adaptive Schedule for TSCH Networks in the Industry 4.0

Pascale Minet, Zied Soua, Ines Khoufi

► To cite this version:

Pascale Minet, Zied Soua, Ines Khoufi. An Adaptive Schedule for TSCH Networks in the Industry 4.0. PEMWN 2018 - 7th IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, Sep 2018, Toulouse, France. pp.1-6. hal-01870252

HAL Id: hal-01870252

<https://hal.science/hal-01870252>

Submitted on 7 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Adaptive Schedule for TSCH Networks in the Industry 4.0

Pascale Minet, Zied Soua and Ines Khoufi

Inria, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France.

Email: pascale.minet@inria.fr, zied.soua@gmail.com, ines.khoufi@gmail.com

Abstract—TSCH (Time Slotted Channel Hopping) networks are excellent candidates to support applications of industry 4.0 with latency, energy and reliability requirements. However, such applications have to face topology, traffic or even application changes. How can TSCH networks adapt to such changes? We propose a solution ranging from network construction to data gathering. We show how the schedule can adapt to changing topology, traffic and application conditions. We compute the theoretical bounds with regard to key performance indicators and compare them with the values obtained by NS3 simulation.

Index Terms—TSCH network, Scheduling, Network building, Data gathering, Latency, Adaptivity, Industry 4.0.

I. CONTEXT AND MOTIVATION

The expected fourth industrial revolution, more known as Industry 4.0 [1], aims at producing higher quality products in a more efficient and flexible way. It relies on the Internet of Things that will enable a smarter monitoring and more self-organizing capabilities than traditional factories.

To produce better quality products and improve monitoring in Industry 4.0 [2], strong requirements in terms of latency, robustness and power autonomy have to be met by the networks supporting the Industry 4.0 applications. The wireless TSCH (Time Slotted Channel Hopping) network specified in the amendment [3] of the IEEE 802.15.4 standard [4] has many appealing properties. Its schedule of multichannel slotted data transmissions ensures the absence of collisions. Because there is no retransmission due to collisions, communication is faster. Since the devices save energy each time they do not take part in a transmission, the power autonomy of nodes is prolonged. Furthermore, channel hopping enables to mitigate multipath fading and interferences.

To increase the flexibility and the self-organizing capacities required by Industry 4.0, the networks have to be able to adapt to changes. These changes may concern the application itself, the network topology by adding or removing devices, the traffic generated by increasing or decreasing the device sampling frequency, for instance. That is why the flexibility of the schedule ruling all networks communications is needed. In this paper, we show how a TSCH network can adapt to such changes. Data gathering transmissions are performed according to a schedule. This schedule is computed by the network coordinator (i.e. CPAN), then installed and activated on all network devices.

In the state of the art, many researchers have investigated the schedule of network communications to meet some requirements in terms of latency [5], [6], energy efficiency [7], [8],

reliability [9], or scalability [10], etc.. The solutions proposed are either centralized like in [11], [12], distributed like in [13], [14] or autonomous like in [15]. The performances achieved strongly depend on the assumptions made concerning the network usage conditions (e.g. traffic load distribution, number of network devices). For instance, centralized solutions are able to compute the schedule minimizing the latency but they require a certain overhead to collect all the information needed to compute the schedule and then to install it on all the devices. As a consequence, they do not scale, unlike distributed solutions. Distributed solutions limit the messages exchanged but are usually unable to provide the optimal schedule and even in some cases are unable to bound how far they are from the optimal. Autonomous solutions can be considered as the extreme case of distributed ones: no message is exchanged between devices, the schedule is built from an implicit knowledge shared between nodes (e.g. knowledge of the cells implicitly assigned to a device for transmission to its parent). The main advantage of such solutions is their great flexibility: they are able to quickly adapt to traffic changes. However, these solutions work well only for very light network loads, for which the collision probability remains very low.

Very few researchers have studied how to schedule the transmissions in a network from the network creation. Devices associate to the TSCH network after having heard a beacon. Since the IEEE802.15.4 standard does not specify how the beacons are advertised, several solutions exist like random advertisement defined in [16], random vertical and random horizontal presented in [17], or the best solutions called MBS [18] and EDDBA [19]. The merits of this paper are to show how an adaptive schedule may be installed and activated on all devices, even in case of high traffic, starting from network construction. It presents both theoretical results and simulation results obtained with NS3 [20].

This paper is organized as follows. Section II deals with the construction of a TSCH network and highlights the main concepts used to minimize the time needed to obtain an operational TSCH network. Section III focuses on data gathering and more precisely the schedule used to transmit data generated throughout the TSCH network to the sink. We show how the schedule adapts to topology, traffic or application changes. In Section IV simulation results obtained with NS3 are reported and compared with theoretical bounds. Finally Section V concludes this paper.

II. NETWORK CONSTRUCTION

A. Schedule and Slotframes

Transmissions within a TSCH network are done according to a global schedule built from the superposition of periodic slotframes. In this paper we distinguish three types of slotframes, ordered by decreasing priority:

- *Beacon slotframe* consisting of a sequence of beacon slots followed by a sequence of free slots (i.e. empty slots). Each beacon slot is assigned to a unique transmitter device, all other devices are in receive mode.
- *Data slotframe* consisting of a sequence of data slots followed by a sequence of free slots. Each data slot is assigned to a unique transmitter device, only the receivers specified are in receive mode, whereas the other devices are sleeping to save energy.
- *Shared slotframe* consisting of a sequence of shared slots followed by a sequence of free slots. In a shared slot, any device is allowed to transmit, whereas all devices having nothing to transmit are in receive mode.

The priorities are such that a Beacon slot wins over a Data slot which wins over a Shared slot which wins over a free slot. To decrease the complexity, the Beacon slotframe size is assumed to be a multiple of both the Data slotframe size and the Shared slotframe size.

As an example, the global schedule resulting from the three Slotframes Beacon, Data and Shared is illustrated in Figure 1. A red slot is a Beacon slot, a green slot is a Data slot and a yellow slot is a Shared slot, whereas a slot with an X-cross denotes a free slot. Since according to the priorities, a Beacon slot wins over a Data slot which wins over a Shared slot which wins over a free slot, we get the following result: slots 0 to 7 are beacon slots; slot 8 is a Shared slot; slots 9 to 16 are Data slots, whereas slot 17 is a free one. Slots 18 to 25 are Data slots and slot 26 is a Shared slot. Then the schedule is repeated from slot 0.

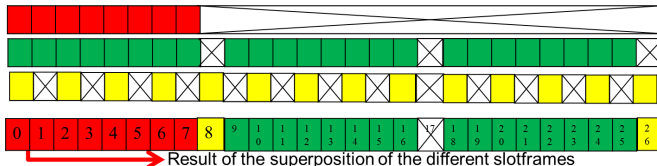


Fig. 1. The resulting schedule.

B. Node Association

To join the TSCH network, a device randomly selects a physical channel used by this network and listens to a beacon advertising this network. Since the physical channel on which the beacon is broadcast changes at each beacon slot due to channel hopping, the joining device will eventually hear a beacon sent by one of its neighbors. Upon receipt of a valid beacon, this device gets synchronized with the TSCH network. It then requests its association to the network.

To minimize the overhead induced by the use of long addresses coded over 6 bytes for network devices, short addresses

coded over 2 bytes are used. To ensure the uniqueness of short addresses, they are assigned by the CPAN (network coordinator) to each joining device when it is associated. As a consequence, the standardized association procedure is modified to become multihop: the association demand is forwarded from the joining node to the CPAN, and the associated response is forwarded from the CPAN to the joining node. In addition, a unique relative beacon index is assigned to the joining device by the CPAN. This beacon index is used by the joining device to compute the beacon slot in which it will broadcast its beacon.

C. Beacon Broadcast

Beacons are broadcast according to an ordered sequence given by the beacon indexes of devices: first the CPAN, then the nodes at depth 1 in the order specified by their index, then the nodes at depth 2 in the order specified by their index, and so on. Since on the one hand, the beacon index ensures that whatever the depth of any device N , N is the unique device to have this index at this depth, and on the other hand, beacons are broadcast according to the order given by the relative beacon index received from the CPAN, denoted $BeaconIndex(N)$ for node N . Hence, *beacons are transmitted without collisions*. The beacon slot offset used by a device N to broadcast its beacon is given by Equation 1:

$$BeaconSlot(N) = DevLess(Depth(N)) + BeaconIndex(N) \quad (1)$$

where $Depth(N)$ denotes the distance expressed in the number of hops from N to the CPAN, and $DevLess(Depth(N))$ is the number of devices at a depth strictly less than $Depth(N)$.

This broadcast of beacons ensures that *any node has received the beacon from its parent before broadcasting its own beacon*. This property enables a consistent information dissemination throughout the network.

III. SCHEDULING DATA GATHERING TRANSMISSIONS

A. Computation of the Data schedule

The CPAN is in charge of computing the schedule for data gathering. For that purpose, the CPAN needs to know for each TSCH device:

- its neighbors with their depth,
- its traffic.

When the CPAN has received the traffic and topology notifications from all TSCH devices, it computes the schedule for data gathering. In order to decrease the latency of data gathering, the schedule tries to minimize the total number of slots assigned for data gathering. However, to reduce complexity and to avoid frequent computations of the schedule due to topology changes only, spatial reuse is not applied in cells, where a cell is uniquely defined by the pair (Slot Offset, Channel Offset). No spatial reuse means that a cell is granted to only one transmitter. To compute a schedule with no spatial reuse, it is sufficient to know the traffic and the parent of each node. This decreases the overhead, since any node needs to

send only its potential parents instead of all its neighbors to the sink.

In the performance evaluation done, the schedule is computed on the CPAN by the debt-based scheduler providing no spatial reuse. Each TSCH device N , except the CPAN, has a debt that is equal to:

$$\begin{aligned} \text{Debt}(N) &= 0 \text{ if } N \text{ has no data message to transmit} \\ &= \text{Depth}(N) \times \text{Rem}(N) \text{ otherwise.} \end{aligned}$$

where $\text{Rem}(N)$ is the remaining number of data messages N has to transmit.

The debt-based scheduler schedules first the transmission of the device with the highest debt. To reduce latency, we assume that data generated in the network are timestamped when they are generated. To minimize the maximum latency that can be achieved by data gathering, Assumption 1 is used:

Assumption 1: Each device transmits first the message with the smallest timestamp.

This assumption ensures that when a device has messages to send that are generated locally or have been received from its children, it always selects the message containing the oldest data. In addition, if several devices have the same debt, the debt-based scheduler selects first the message with the smallest timestamp.

An example of a Data schedule is depicted in Figure 2, where the TSCH network uses 5 channels and the sink has 4 radio interfaces. 24 slots are used to transmit Data. The topology is that depicted in Figure 3 and traffic generation is given in Section IV-A.

Chann	Slot	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
Channel 0		18->0	18->0	18->0	18->0	31->18	18->0	32->18	18->0
Channel 1		14->0	14->0	14->0	14->0	9->0	13->0	14->0	5->0
Channel 2		1->0	3->0	5->0	7->0	10->0	19->0	2->0	6->0
Channel 3		2->0	4->0	6->0	8->0	11->0	1->0	3->0	7->0
Channel 4		26->17				12->0	23->14	4->0	24->14
Chann	Slot	Slot 8	Slot 9	Slot 10	Slot 11	Slot 12	Slot 13	Slot 14	Slot 15
Channel 0		31->18	18->0	32->18	18->0	31->18	18->0	32->18	18->0
Channel 1		14->0	11->0	14->0	3->0	6->0	10->0	13->0	1->0
Channel 2		8->0	12->0	19->0	4->0	7->0	11->0	14->0	2->0
Channel 3		9->0	13->0	1->0	5->0	8->0	12->0	17->0	3->0
Channel 4		10->0	27->14	2->0	28->14	9->0	30->14	19->0	
Chann	Slot	Slot 16	Slot 17	Slot 18	Slot 19	Slot 20	Slot 21	Slot 22	Slot 23
Channel 0		25->18	18->0	29->18	18->0	31->18	18->0	32->18	18->0
Channel 1		4->0	8->0	11->0	15->0	19->0			
Channel 2		5->0	9->0	12->0	16->0	20->0			
Channel 3		6->0	10->0	13->0	17->0	21->0			
Channel 4		7->0		14->0		22->0			

Fig. 2. An example of Data Schedule with 24 slots and 5 channels.

B. Schedule Installation

The intuitive idea is to take advantage of the ordered sequence of beacon broadcasts starting with the CPAN and progressively involving all devices by increasing depth, to broadcast the schedule to all devices. Since the Data schedule size may be large, it will be fragmented if needed. Each beacon contains a fragment of the schedule. For that purpose, Information Elements are used according to the TSCH specification. The size of the fragment is bounded by the available space in a beacon, denoted *BeaconSpace*. The number of fragments *Frag* to broadcast a schedule of size *ScheduleSize* is given by:

$$\text{Frag} = \lceil \frac{\text{ScheduleSize}}{\text{BeaconSpace}} \rceil \quad (2)$$

C. Schedule Activation

All devices simultaneously activate the new schedule at the Absolute Slot Number (ASN) corresponding to the next beacon slot assigned to the CPAN. Hence, all transmissions, including data, beacons and control messages are consistently done according to this active schedule.

D. Schedule Update

As soon as a device observes a change, it notifies the CPAN, using Shared slots. Notice that since this device may be several hops away from the CPAN, the notification should be forwarded if needed. Upon receipt of a notification, the CPAN waits for a given amount of time in order to collect other notifications, because usually, several devices notify the same change or changes sharing the same cause. Then the CPAN takes the decision to:

- either update the current Data schedule by modifying a limited number of cells. The schedule gives for each cell of the Data slotframe the transmitter and the receiver(s).
- or compute a new Data schedule.

The new schedule or the update of the current one is broadcast in the next beacon of the CPAN and forwarded to all network devices according to the ordered sequence defined in Subsection II-C. All network devices will then activate this new schedule or the updated one in the next beacon slot of the CPAN.

IV. PERFORMANCE EVALUATION

We used the NS3 simulation tool to evaluate the performances of the solution presented in Sections II and III. For that purpose, we developed and integrated the modules implementing the solution proposed.

A. Simulation parameters

The network is made up of the CPAN denoted as device 0 and 32 other devices. We consider three topologies:

- *The Regular topology*, where all devices are regularly spread on two circles of range R and $2R$, centered on the CPAN, where R denotes the radio range. This configuration is used as a reference to point out the impact of device density on network key performance indicators.
- *The Irregular1 topology*, where all devices are uniformly distributed in three sectors around the CPAN.
- *The Irregular2 topology*, depicted by Figure 3, is similar to the previous one, except that devices are distributed in 5 sectors around the CPAN and two-hop devices are more concentrated, resulting in a higher number of children per parent.

The simulation parameters used for the performance evaluation with NS3 are summarized in Table I. We distinguish between common parameters that are used for the three topologies considered, from those strongly depending on the configuration such as for example the number of Data slots. Devices 1 to 14, 18, 31 and 32 generate 400 samples per second, whereas the other devices generate 4 samples per second.

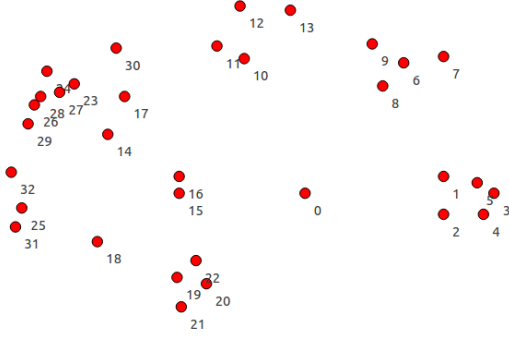


Fig. 3. The Irregular2 topology considered: 33 nodes spread in 5 sectors.

TABLE I
SIMULATION PARAMETERS.

Common parameters	Values
Number of CPAN radio interfaces	4
Number of channels used	5
Number of nodes	33
One-hop nodes	1 to 22
Two-hop nodes	23 to 32
Topology	Regular, Irregular1, Irregular2
SlotDuration	10 ms
BeaconUsed	33 slots
BeaconFrame size	43 slots during construction
Parameters for the Irregular2 topology	Values
BeaconFrame size	75 slots during data gathering
DataFrame size	25 slots
DataUsed	24 slots

The parent-child relationship for the Irregular2 topology is given by Table II.

TABLE II
PARENT-CHILD RELATIONSHIP FOR THE IRREGULAR2 TOPOLOGY.

Parent	Children
0	1 to 22
14	23, 24, 27, 28, 30
18	25, 29, 31, 32
17	26

B. Simulation results and theoretical bounds

In this subsection, we evaluate different times that are key performance indicators of the TSCH network. For each time evaluated, we give both an upper theoretical bound and the value obtained for the configuration described in subsection IV-A. These times are related to network building as well as data gathering and adaptivity to topology/traffic/application changes. For this evaluation, we assume that there is no wireless link/device failure. For network building, we also assume that all devices are simultaneously powered on, leading to a bottleneck on network association: all devices wanting to simultaneously join the network.

1) Network building:

The average number of neighbors for the three topologies studied is given by Table III.

TABLE III
AVERAGE NUMBER OF NEIGHBORS FOR THE THREE TOPOLOGIES.

Topology	Avg # of neighbors		
	for 1-hop dev.	for 2-hop dev.	1-hop from the CPAN for 1-hop dev.
Regular	11.5	7.2	11
Irregular1	14.31	8.5	10.95
Irregular2	17.13	19.5	12.36

Figures 4 and 5 depict the average delay obtained on 50 simulation runs by one-hop devices and two-hop devices, respectively, to:

- *receive their first beacon* on the channel they randomly chose. This time is measured from the time the device is powered on and averaged. For one-hop devices, this delay is shorter with the Regular and Irregular2 topologies that have the smallest average number of neighbors 1-hop away from the CPAN (see the last column of Table III). For two-hop devices, the delay is higher with the Regular topology that has the smallest average number of neighbors (see the third column of Table III). This can be explained by the fact that when the average number of neighbors increases, the probability of receiving a beacon on the channel selected by the joining device increases which leads to a smaller waiting time before the first beacon receipt. Furthermore, since only associated devices are allowed to send beacons, two-hop devices have to wait the association of their parent, explaining their higher time.
- *receive their association confirmation*. This time is measured from the time of the first association request of the device considered and averaged. Since the association of two-hop devices is multihop, two-hop devices have a longer association time than one-hop devices.
- *send their first beacon*. This time is measured from the receipt of the association confirmation of the device considered and averaged. The maximum time taken to send their beacon is upper bounded by the duration of two Beacon Slotframes, as verified by the values obtained by simulation with NS3.

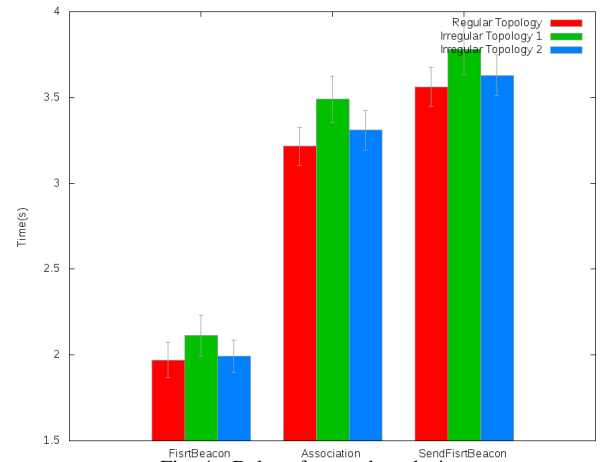


Fig. 4. Delays for one-hop devices.

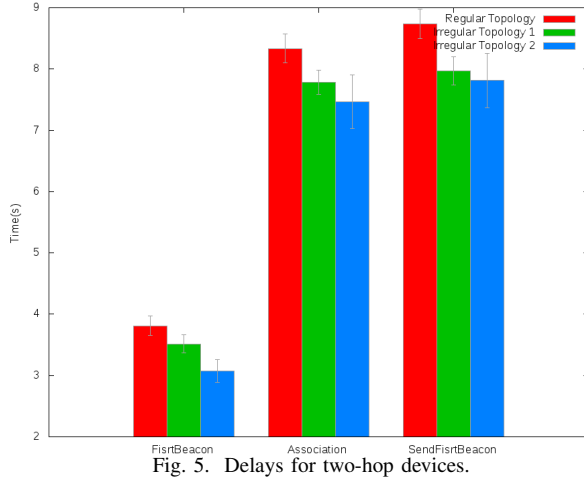


Fig. 5. Delays for two-hop devices.

Finally, *the delay needed by network building* is depicted by Figure 6. It is the time elapsed between the transmission by the CPAN of its first beacon and the receipt by the last device of its association confirmation. The greatest time consuming step is the association of network devices. All three topologies lead to similar results.

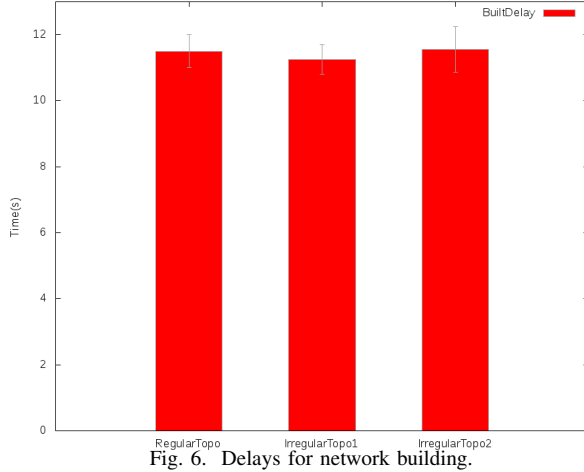


Fig. 6. Delays for network building.

The average number of association requests per node is depicted by Figure 6, where one-hop devices send similar numbers of requests in the three topologies considered, whereas two-hop devices send the highest number of requests for the Irregular2 topology, that has the highest average number of neighbors for two-hop devices (see the third column of Table III).

2) Data gathering:

The maximum time to install the schedule on all devices, denoted *Install*, is defined as the time elapsed between the CPAN broadcast of the beacon containing the first fragment of the schedule, and the receipt by the last device of the last fragment of the schedule. We have:

$$\begin{aligned}
 Install = & ((Frag - 1)BeaconFrameSize \\
 & + BeaconSlot(Parent(LastDev)))SlotDuration \\
 & + TransDelay
 \end{aligned}$$

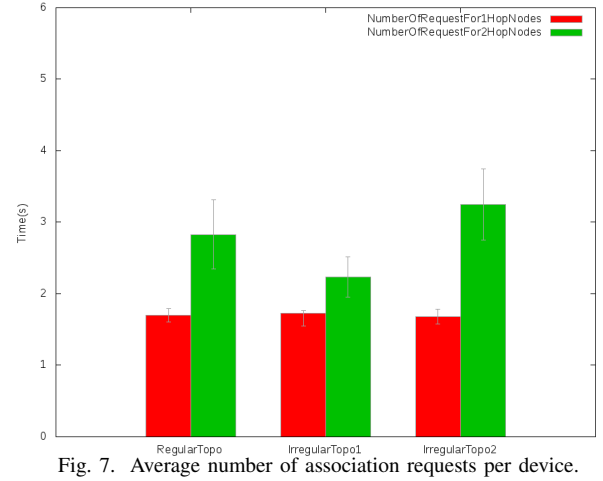


Fig. 7. Average number of association requests per device.

where $BeaconSlot(Parent(LastDev))$ denotes the beacon slot assigned to the parent of the last device receiving the beacon and $TransDelay$ is the transmission delay of the last schedule fragment.

As an example, we measured the *Install* time for the Data schedule depicted in Figure 2. For that schedule, 11 fragments are needed, taking into account a *BeaconSpace* enabling to transmit at most 10 data links (i.e. cells). We got 4.4947 s which corresponds to the theoretical bound computed previously.: $(10 * (43 + 19) * 10 + 0.0047)ms = 4.4947/s$. This value could be optimized by a better coding of the cell with its transmitter and receiver. The coding of Information Elements used to insert a schedule fragment in the beacon message could also be optimized for a better efficiency.

The maximum time to activate the schedule on all devices, denoted *Activate*, is defined as the time elapsed between the CPAN broadcast of the first fragment of the schedule and the broadcast of the first CPAN beacon following the broadcast of the last fragment; We then have:

$$Activate = Frag \cdot BeaconFrameSize \cdot SlotDuration \quad (3)$$

As an example, we measured the *Activate* time for the Data schedule depicted in Figure 2. We got 0.4730 s, which corresponds to the number of fragments 11 times the duration of the Beacon slotframe containing 43 slots of 10 ms.

The maximum latency for data gathering, denoted *Latency*, is defined as the time elapsed between data generation on a device node and its delivery to the sink. The worst case occurs when data are generated while beacons are transmitted, the device has to wait for the next Data slotframe that is not evicted by beacons. In the worst case and with assumption 1, it takes a number of slots equal to the number of Data slots in the Data slotframe to reach the CPAN. Hence, the upper bound:

$$\begin{aligned}
 Latency = & ((1 + \lceil \frac{BeaconUsed}{DataFrameSize} \rceil) \cdot DataFrameSize \\
 & + DataUsed) \cdot SlotDuration
 \end{aligned} \quad (5)$$

where *BeaconUsed* is equal to the total number of devices in the TSCH network, each of them has its own beacon slot, *DataUsed* is the number of slots used in the Data schedule. With the parameters given in Table I, the theoretical upper bound for the latency is equal to 0.99s. Simulation results corroborate this value. Hence, the time needed to adapt the schedule to topology or traffic changes can be bounded and the value obtained remains acceptable by most applications.

V. CONCLUSION

In this paper, we have shown how a TSCH network is autonomously built, supports data gathering and is able to adapt to changes in network topology, traffic and application requirements.

The solution proposed preserves the merits of TSCH network, that can be listed hereafter. The time-slotted multichannel medium access enables parallel transmissions on several channels, leading to shorter latency and higher throughputs. In addition, channel hopping mitigates interference and multipath effects. Furthermore, since transmissions are scheduled, a conflict-free schedule is computed by the network coordinator (i.e. the CPAN). Hence, no collision occurs during data gathering. The absence of collision leads to a higher throughput, because there is no retransmission due to collisions. It also preserves nodes power autonomy.

This simple solution based on the coexistence of several slotframes. We distinguish three slotframes, that are the Beacon Slotframe, the Data Slotframe and the Shared Slotframe. The network schedule corresponds to the superposition of the three schedules given by each slotframe, where the slotframe with the highest priority wins.

This solution ensures a collision-free dissemination over the whole network. Beacons are broadcast in sequence by increasing depth of devices. This broadcast is also used to disseminate Data Schedules (new schedule or update).

In addition, this solution is adaptive. Topology, traffic or application changes are notified to the CPAN. Depending on the changes notified, the CPAN updates the current schedule or recomputes a new one. Shared slots are used to cope with unexpected events.

Simulation results obtained with NS3 confirm the theoretical upper bounds computed for network construction and data gathering. Hence, TSCH networks are able to adapt to traffic or topology changes in a reasonable time which is a strong requirement of Industry 4.0 applications. In some further work, we will study how to improve this delay to support the most demanding applications.

REFERENCES

- [1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 working group," Frankfurt, Germany, Tech. Rep., 2013.
- [2] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS)*, ser. HICSS '16. Washington, DC, USA: IEEE Computer Society, 2016, pp. 3928–3937. [Online]. Available: <http://dx.doi.org/10.1109/HICSS.2016.488>
- [3] IEEE SA, "IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) — Amendment 1: MAC sublayer," IEEE, IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011), February 2012. [Online]. Available: <http://standards.ieee.org/getieee802/download/802.15.4e-2012.pdf>
- [4] —, "IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE, IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006), Sept 2011. [Online]. Available: <http://dx.doi.org/10.1109/ieeestd.2006.232110>
- [5] A. Yang, A. Sundararajan, C. Schindler and K. Pister, "Analysis of Low Latency TSCH Networks for Physical Event Detection," in *WCNC 2018, IEEE Wireless Communications and Networking Conference*, Barcelona, Spain, 2018.
- [6] S. Rekik, N. Baccour, M. Jmaiel and K. Drira, "A Performance Analysis of Orchestra Scheduling for TSCH network," in *Internet Technology Letters*. Wiley, 2017.
- [7] X. Vilajosana and Q. Wang and F. Chraim and T. Watteyne and T. Chang and K. Pister, "A Realistic Energy Consumption Model for TSCH Networks," *IEEE Sensors Journal*, vol. 14, no. 2, February 2009.
- [8] N. Choudhury, R. Matam, M. Mukherjee and L. Shu, "Beacon Synchronization and Duty-Cycling in IEEE 802.15.4 Cluster-Tree Networks: A Review," in *IEEE Internet of Things Journal*, 2018.
- [9] P. Minet, I. Khoufi and A. Laouiti, "Increasing reliability of a TSCH network for the industry 4.0," in *16th IEEE International Symposium on Network Computing and Applications (NCA 2017)*, Boston, Massachusetts, 2017.
- [10] S. Zats and R. Su and T. Watteyne and K. Pister, "Scalability of Time Synchronized Wireless Sensor Networking," in *IECON 2011, 37th Annual Conference on IEEE Industrial Electronics Society*, Melbourne, Australia, May 2011.
- [11] I. Khoufi, P. Minet and B. Rmili, "Scheduling transmissions with latency constraints in an IEEE 802.15.4e TSCH network," in *IEEE 86th Vehicular Technology Conference (VTC 2017-Fall)*, Toronto, Canada, 2017.
- [12] R. Soua, P. Minet, and E. Livolant, "MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks," in *IPCCC 2012 : 31st IEEE International Performance Computing and Communications Conference*. Austin, Texas, United States: IEEE, Dec. 2012, pp. 91 – 100. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00863360>
- [13] —, "Wave: A distributed scheduling algorithm for convergecast in ieee 802.15.4e tsch networks," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 4, pp. 557–575, Apr. 2016. [Online]. Available: <https://doi.org/10.1002/ett.2991>
- [14] N. Accettura, E. Vogli, M. Palattella, L. Grieco, G. Boggia, and M. Dohler, "Decentralized traffic aware scheduling in 6tisch networks: Design and experimental evaluation," vol. 2, 12 2015.
- [15] S. Rekik, N. Baccour, M. Jmaiel, K. Drira and L.A. Grieco, "Autonomous and traffic-aware scheduling for tsch networks," *Computer Networks*, vol. 135, pp. 201 – 212, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128618300963>
- [16] D. De Guglielmo and A. Seghetti and G. Anastasi and M. Conti, "A performance analysis of the network formation process in IEEE 802.15.4e TSCH wireless sensor/actuator networks," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, Madeira, Portugal, 2013.
- [17] E. Vogli and G. Ribezzo and L. Alfredo Grieco and G. Boggia, "Fast Join and Synchronization Schema in the IEEE 802.15.4e MAC," in *2015 IEEE Wireless Communications and Networking Conference (WCNC) - Workshop - Energy Efficiency in the Internet of Things*, New Orleans, LA, 2015.
- [18] D. De Guglielmo and S. Brienza and G. Anastasi, "A Model-based Beacon Scheduling algorithm for IEEE 802.15.4e TSCH networks," in *2016 IEEE 17th international symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Coimbra, Portugal, 2016.
- [19] I. Khoufi and P. Minet, "An enhanced deterministic beacon advertising algorithm for building TSCH networks," *Annals of Telecommunications*, pp. 1 – 13, May 2018. [Online]. Available: <https://doi.org/10.1007/s12243-018-0638-3>
- [20] <https://www.nsnam.org/>, "NS3 a discrete-event network simulator for internet systems. online."