Collision Prevention in Distributed 6TiSCH Networks

Ali Fahs, Rodolphe Bertolini, Olivier Alphand, Franck Rousseau, Karine Altisen, Stéphane Devismes

Collision Prevention in Distributed 6TiSCH Networks

Ali J. Fahs*[‡§], Rodolphe Bertolini*[‡§], Olivier Alphand[†‡], Franck Rousseau[†‡], Karine Altisen[†§], Stéphane Devismes*[§]

Université de Grenoble Alpes, CNRS

*UGA, [†]Grenoble INP

[‡]LIG, [§]VERIMAG

F-38000 Grenoble, France

firstname.lastname@univ-grenoble-alpes.fr

Abstract—The IEEE802.15.4e standard for low power wireless sensor networks defines a new mode called Time Slotted Channel Hopping (TSCH) as Medium Access Control (MAC). TSCH allows highly efficient deterministic time-frequency schedules that are built and maintained by the 6TiSCH operation sublayer (6top). In this paper, we propose a solution to limit the allocation of identical cells to co-located pair of nodes by distributed TSCH scheduling algorithms. It consists of making nodes able to overhear past cell negotiations exchanged in shared cells by their neighbors and prevent the nodes from reusing already assigned cells in future allocations. Our mechanism has been tested through simulations that show a significant improvement with respect to random scheduling algorithms.

Keywords-IoT; IEEE802.15.4e; TSCH; 6TiSCH; 6top; Scheduling Functions;

I. INTRODUCTION

Network technologies have shifted from a limited number of interconnected expensive computers with high performance processing units to networks composed of a huge number of low cost devices with limited processing capabilities called *things*, which leads to the development of the *Internet of Things (IoT)*.

Among many standards dedicated to IoT, the initial version of IEEE802.15.4 [1] aimed at low power short range communications. However its MAC layer presented several limitations (no channel diversity, unbounded delay, and poor multi-hop topology support). To overcome these limitations, IEEE802.15.4e [2] extended the standard in 2012 by proposing a new mode called *TSCH (Time-Slotted Channel Hopping)*. TSCH defines a periodic slotframe consisting of fixed-size *cells*, each indexed by a specific timeslot offset and a channel offset. Each cell has the same duration and its length should be large enough to accommodate a maximum frame length and its acknowledgment, while ensuring at the same time synchronization between communicating nodes. TSCH also provides channel diversity, enabling parallel communications on different channels within the same timeslot, as well as channel hopping in the same cell from one slotframe to another. The channel of a cell is derived from the channel offset and the absolute slot number. There are two types of TSCH cells: *shared cells* where multiple nodes may either receive or transmit, and *dedicated cells* for contention-free link-local communications between two neighboring nodes.

Since we deal with a large number of nodes, sensors, and things that should be connected to the Internet, large scaling capabilities are mandatory. With this goal in mind, IETF (Internet Engineering Task Force) has standardized an IoT stack based on 802.15.4. The integration of TSCH in this stack is currently being carried out by the 6TiSCH IETF Working Group through the 6TiSCH operation sublayer called *6top*. It includes the definition of algorithms to schedule dedicated cells and a 6top protocol (*6P*) to negotiate cells between nodes [3].

In this paper, we focus on improving the distributed scheduling algorithms that are mainly designed to address the problem of convergecast traffic towards a unique sink in a multi-hop network. *RPL (Routing Protocol for Low-Power and Lossy Networks)* [4] organizes the network as a *DODAG (Destination-Oriented Directed Acyclic Graph)* topology along which the data generated by each node are relayed up to its root. Scheduling algorithms thus attempt to create efficient TSCH schedules along RPL routes.

Most of existing scheduling functions randomly select cells among the cells that are not already used by both negotiating nodes with their respective DODAG parents and children. However, the cell selection process should also preclude dedicated cells already reserved by other co-located neighboring nodes that may interfere leading to collisions and consequently to packet dropping. Our solution makes distributed scheduling algorithms aware of dedicated cells already assigned in the neighborhood by overhearing past cell negotiations exchanged with 6P in shared cells.

To evaluate our solution, we compare the performance of the *On-the-Fly* (OTF) [5] scheduling algorithm with and without our approach. Results show a significant improvement, in particular, a decrease of the number of colliding cells[1] and packets during the bootstrap phase.

The main contributions of our paper are the following:

- Overhearing of 6P transactions to collect dedicated cells reserved by neighbors without introducing any signaling overhead.
- Addition to 6P transactions of a short-term memory of

---

[1]A colliding cell is a dedicated cell reserved by multiple co-located pairs of nodes in which packets might collide in case of simultaneous transmissions.

past reserved cells to make the overhearing more robust.

- Modification of the cell selection algorithm in 6top to avoid the cells used by neighboring nodes.
- Comparative simulation-based evaluation of a random scheduling algorithm with and without our approach.

The paper is organized as follows: Section II gives an overview of 6TiSCH and the existing distributed scheduling algorithms. Section III describes our solution with its various features. Section IV details the changes of the 6TiSCH simulator as well as its parameters and presents the results of simulations. We make concluding remarks in Section V.

II. BACKGROUND

A. 6TiSCH Operation Sublayer (6top)

IETF puts a lot of effort in the standardization of an IoT stack compliant with the IEEE802.15.4 PHY/MAC layer. However, the original 802.15.4 MAC layer lacks some important features. Those gaps were filled by the TSCH mode of IEEE802.15.4e. To properly integrate TSCH within the IoT stack, the IETF 6TiSCH Work Group defines the 6TiSCH operation sublayer (6top). Figure 1 illustrates how 6top integrates with the IoT stack.
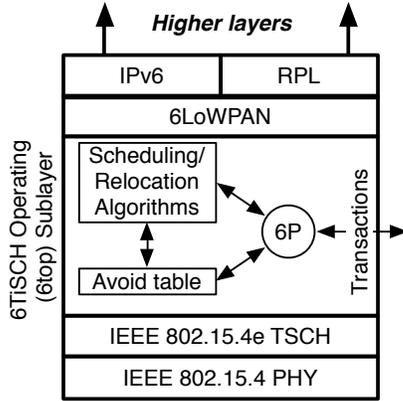

Fig. 1. TSCH integration with the IETF IoT stack

6top includes the definition of policies to manage TSCH schedules and the 6top protocol (6P) to control the communication for cell reservation and deletion between nodes.

If we consider a transmitting node (Tx) and a receiving node (Rx) that communicate in the 6TiSCH network (N.B.: Rx is a parent of Tx in the DODAG and, conversely, Tx is a child of Rx), 6P supports three types of operations that can be triggered by *Scheduling Functions* (SF):

1) If the *Scheduling Algorithm* in Tx determines that it needs more cells to communicate with Rx, it will issue a 6top transaction to reserve more cells in the TSCH schedule.
2) If the *Scheduling Algorithm* in Tx determines that it needs less cells to communicate with Rx, it will issue a 6top transaction to release some TSCH cells.
3) The *Relocation Algorithm* in Tx and/or Rx detects whether a dedicated cell is facing collisions or not. If

so, Tx or Rx replaces the defected cell through 6top transactions.

As explained before, each operation is encapsulated into a transaction. A 6top transaction consists of a negotiation between Tx and Rx that updates the TSCH table. This transaction takes place in two or three steps. In a 6top transaction, the scheduling function decides whether to use a 2-step transaction, a 3-step transaction, or a mix of them. Furthermore, the scheduling function has two options to communicate the list of cells:

1) Black listing: where the node sends all cells in use.
2) White listing: where the node sends a number of cells available for communication.
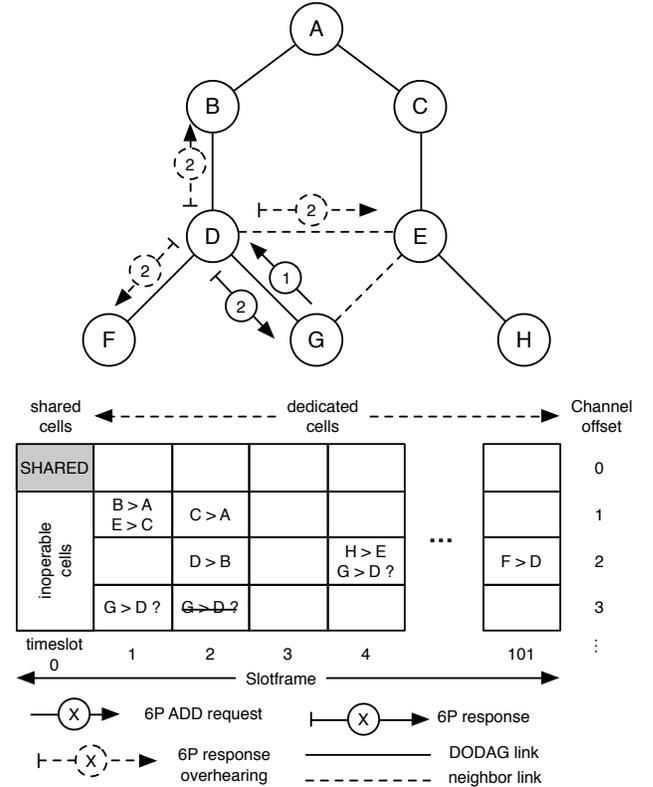

Fig. 2. 6P transactions and TSCH schedule update

For better understanding, we detail in Figure 2 the sequence of a 2-step transaction using white-listing to allocate two cells for link $G \to D$:

(1) The Scheduling Function (SF) of node G detects that two additional cells are needed and selects three candidate available cells $\{(1,3),(2,3),(4,2)\}$[2] before triggering a 6P ADD request towards node D in the shared slot.

(2) The SF running on node D precludes the cell $(2,2)$, selects 2 cells $\{(1,3),(4,2)\}$, and issues a 6P response to node G.

---

[2]The tuple (i, j) identifies the cell at timeslot i with channel offset j.

As we can see in this example, the cell selection is not done randomly among all the cells of the TSCH slotframe. The 6top negotiation natively precludes all the cells in timeslots in which both negotiating nodes communicate with their DODAG parents and children. The selection of the candidate cell is then randomly done in the set of remaining cells. In the previous negotiation, D will not consider timeslot 2 since it already uses it to communicate with its parent B. G has no dedicated cells so it does not remove any cells a priori.

In the next section, we will describe TSCH scheduling distributed algorithms and, in particular, how they tackle the problem of reducing scheduled colliding cells.

B. Related Work

First, we consider networks that are dynamic in terms of traffic and/or topology. So, centralized scheduling approaches should be given up, since gathering all information by a single node and redistributing the computed schedule to every node imply a huge overhead in terms of signaling. In such a context, distributed algorithms are much more interesting as cell scheduling is addressed autonomously by each node on the basis of local information exchanged with neighbors. Thus, the signaling traffic is reduced, but the probability of collisions is increased since nodes have only a local view of the network.

A lot of distributed TSCH scheduling algorithms only rely on the basic 6top random cell selection. The inherent colliding cells are reduced a posteriori by a reactive relocation algorithm.

On-the-Fly (OTF) [5] implements a bandwidth reservation mechanism that compares the traffic sent by a node to each of its neighbors and the corresponding scheduled cells. Above/under a certain threshold, it triggers the negotiation to add/delete cells. The resulting cell schedule is not conflict-free but a mechanism called *housekeeping* relocates under-performing cells [6]. Theoleyre and Papadopoulos provided two policies: the random strategy and the random contiguous one [7]. Both strategies reduce the number of colliding cells with a relocation algorithm. It is the same for the scheme by Domingo-prieto et al. [8]. The problem of such approaches is that the relocation of colliding cells takes time, implies extra 6top traffic, and could have been avoided earlier by not selecting those colliding cells initially in the scheduling algorithm.

Several proactive algorithms follow this latter rule. Among them, DeTAS [9] and Wave [10] proposed to build a collision-free schedule. Both algorithms are not entirely distributed, as each child needs some input from its parent to locally execute the algorithm.

The problem with Wave is that it requires the knowledge of the conflict graph, while no instruction is given on how to build this graph.

DeTAS avoids interference between devices by separating in time and frequency each branch of the DODAG and re-uses channels 3 hops away. However, such a solution limits the spatial reuse of timeslots and channels in the whole DODAG.

Compared to fully distributed algorithms, this approach allows to obtain collision-free schedules but incurs in return increased latency and signaling overhead. Instead, our approach is to improve fully distributed algorithms with lighter signaling. The next section details how our solution proactively reduces colliding cells in such random scheduling without adding any overhead.

III. PROPOSED MECHANISM

As mentioned in Section I, our approach is complementary to scheduling algorithms since we aim at preventing the scheduling function of nodes from considering cells already allocated in their neighborhood.

In the example detailed in Section II-A, the negotiation results in the allocation of cell (4,2) to link $G \rightarrow D$, which is already allocated to link $H \rightarrow E$. If both links are active in this cell, a collision occurs. Now, it could have been prevented if G was aware of the previous allocation negotiated by H with E, which is within the 1-hop neighborhood of G. More generally, this kind of collision occurs at the reception node Rx and the neighboring nodes of Rx are the only nodes that can cause collisions. This occurs when Rx has two or more neighbors transmitting in the same scheduled cell, one of these must be a child, and the rest are neighbors transmitting to their parents.

Our solution is based on the idea of local mutual exclusion (LME). Local means the neighborhood of the reception node, and mutual exclusion is applied to prevent nodes from accessing a common resource at the same time. When a pair of Rx and Tx nodes reserves a new communication cell, Rx must inform its neighbors of the set of the reserved cells.

A. Overhearing for Achieving LME

The idea behind our approach is to overhear 6top transactions from our neighbors. 6top transactions are sent as unicast in a shared slot and received by all the neighbors at the PHY layer. Then, all of them except the destination reject the packet due to the MAC filtering. If we apply a slight modification at the MAC sublayer, we can disable filtering those transactions, and consequently, collect the cells reserved by the neighbors. Moreover, it is achieved without introducing any overhead.

This mechanism is highly efficient at the network bootstrap since no dedicated cell has been scheduled. To allocate their first cells, nodes will send their 6P messages in the shared cells defined by the 6TiSCH minimal TSCH schedule [11]. Their location is either preconfigured or learnt by nodes at the bootstrap. One example of this schedule can be seen in Figure 2 where one shared cell is located at the first timeslot 0 of the slotframe with the channel offset 0. The access to shared cells follows slotted ALOHA rules. So, the higher the number of 6P transactions, there are more collisions and retransmissions in shared cells. Once dedicated cells are allocated between a pair of nodes, further 6P transactions should use dedicated cells. In our approach, we always consider 6P transactions that use shared cells to benefit from overhearing.

In the case the scheduling function uses 2-step transactions, we can collect the cells reserved by Rx from the ADD

k	8	9	10	11	12
P_o	94.23%	95.96%	97.17%	98.02%	98.61%

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Number of motes	100
Number of cycles per run	500
Number of runs per simulation	1000
Timeslot duration	10 ms
Slotframe length	101
Number of channels	16
Area	1 km × 1 km
Topology constraint	≥ 3 neighbors with PDR 50 %
Radio sensitivity	−97 dBm
Radio range	100 m
Traffic	1 packet/cycle for each node

response, using white listing. Similarly, this approach can be implemented with 3-step transactions by using black listing.

B. Avoid Table

The collected cells will be saved in a table called an *avoid table*. The structure of this table is similar to that of TSCH in which each cell is represented by a timeslot and a channel offset. Each cell in the avoid table can be marked as available or reserved. The cell is available if it is not used by any of the neighbor nodes. If it is reserved, it should be avoided because of possible collisions. The last step of the transaction is to force scheduling functions to avoid cells found in the avoid table.

C. Cell Buffer

Due to the unreliable nature of low power wireless links, some transactions might not be received by all neighbors leading to inconsistencies among neighbors. Therefore, those neighbors could reserve already allocated cells leading to collisions. To mitigate the effect of lost transactions, we add a cell buffer that stores the last k cells reserved by a node with its children, where k is the size of the buffer. Thus, whenever a cell is reserved, instead of sending only the recently reserved cells, the cell buffer will be sent. As a result, all the cells in the received buffer will be added to the avoid table. Technically, it means that each cell will be transmitted k times to the neighborhood. It effectively increases the number of successful receptions of the cell reserved by neighbors, and consequently reduces the number of collisions.

The value of k is calculated using a probabilistic model for the 6top transactions. We assume that the probability that a neighbor successfully receives the transaction is p and that the number of successfully received transmissions (Y) follows a binomial distribution law $\mathcal{B}(k,p)$ (each cell in the buffer will be transmitted k times). We denote by P_o the probability of at least one successful reception.

$$Y \sim \mathcal{B}(k,p) \quad P_o = P(Y \geq 1) = 1 - (1-p)^k$$

Solving this equality leads to:

$$k = \left\lceil \frac{\log (1 - P_0)}{\log (1 - p)} \right\rceil$$

By simulating example topologies, we have calculated the average value of the *Packet Delivery Ratio (PDR)* for the neighborhood of each node. To ensure our proposal, we have taken a low value of PDR ($p = 0.3$). According to the equations, we obtain the results presented in Table I:

Finally, with a buffer size of 10 (the value used in our simulations), we can ensure with 97 % confidence that the cell reserved will be received by all the neighbors.

IV. RESULTS

A. 6TiSCH Simulator

In our tests, we have used the 6TiSCH simulator[3], which is an implementation of IEEE802.15.4e that uses RPL for routing, 6top for the management of TSCH, and OTF (On-the-Fly) for scheduling [5].

We first had to overcome a limitation of the simulator that bypassed the exchange of 6P messages in the existing implementation. Cell negotiations between nodes were "instantaneous", not resulting in interference, collisions, and re-transmissions. We have fixed this issue so that cell negotiations trigger the exchange of 6P packets in shared cells. Second, we have implemented the mechanism of collision prevention in the 6top layer by adding a structure to collect cells at each node, and by adding the modifications explained previously in the 6top transactions.

The simulations were done over a wide range of topologies and each tested protocol was run on the same topologies to ensure fairness.

Table II summarizes the simulation parameters.

B. Comparisons without Housekeeping

First, we have compared the performance of OTF with and without our approach. Housekeeping is disabled. As explained before, housekeeping is based on a periodic function that will check the underperforming scheduled cells by tracking their PDR. Under a certain threshold, a cell will be relocated.

In Figure 3 and 4, *Random* stands for the original OTFprotocol in which nodes only avoid cells already reserved with parents and children. The *ME* curve corresponds to OTF with our local mutual exclusion mechanism but with no cell buffer while the curve *MECB* represents the behavior of the protocol with a cell buffer of size 10.

The simulation is done over 500 runs. Each run consists of 1000 cycles, the cycle refering to one slotframe.

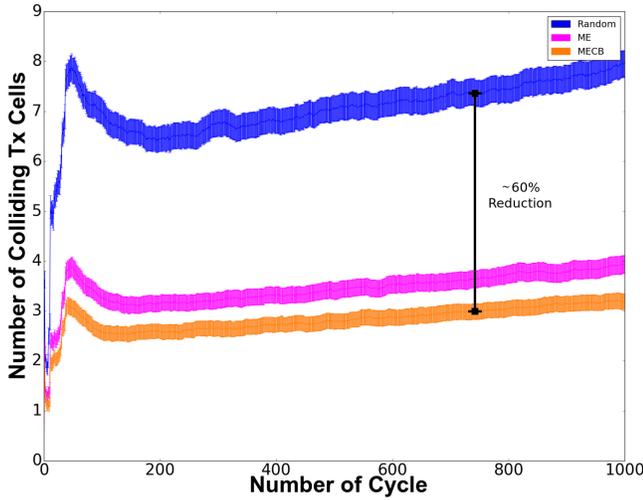[3]https://bitbucket.org/6tisch/simulator/src

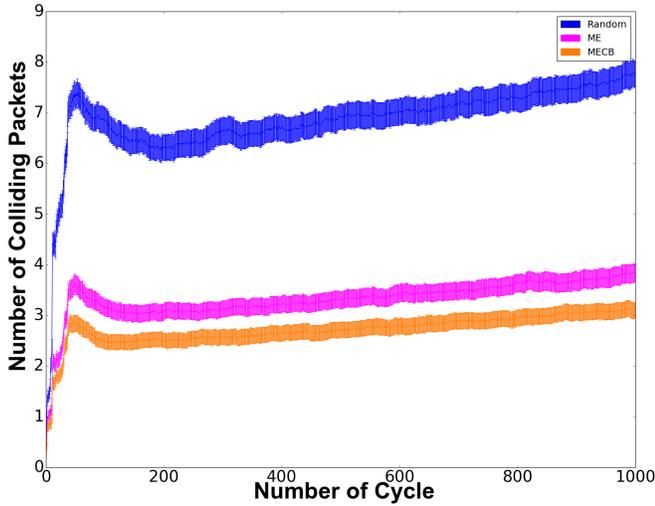Fig. 3. Number of colliding Tx cells as a function of cycles

Fig. 4. Number of colliding packets as a function of cycles

At the beginning of the simulation, the increase of collided cells corresponds to the bootstrap of the network: during the creation of the DODAG by RPL, all nodes start requesting dedicated cells by sending 6top messages in shared cells. Collisions in shared cells will be progressively resolved by CSMA/CA before being able to schedule the first dedicated cells. As we can see in Figure 3, the number of colliding dedicated cells increases and then stabilizes when all requested cells are scheduled.

The simulation shows a reduction of 62 % in the number of colliding Tx cells. The use of the cell buffer is responsible for a 12 % reduction between ME and MECB. We can conclude that the lost 6top transactions cause 12% of the collisions.

In Figure 4, we can see the reduction in the number of colliding packets resulting from our approach. The collided packets are directly related to the traffic of the network. In the simulation, we have a constant traffic of the nodes (1 packet/cycle for each node). A colliding Tx cell does not

necessarily mean that we will have a collision each cycle in this cell. For instance, one of the nodes will transmit packets while the other might have nothing to transmit.

Similarly, our approach shows a reduction of 60 % in the number of colliding packets. The local mutual exclusion mechanism with the cell buffer also shows an improvement over the local mutual exclusion without the cell buffer.

Despite the reduction of their number, collisions still remain for the following reasons:

- nodes still have a slight chance of losing 6P transactions.
- A specific configuration in which a node Tx2 belongs to the neighborhood of a node Rx1, but Rx2 (the parent of Tx2) has neither Tx1 or Rx1 in its neighborhood. If Rx2 first reserves a cell, Rx1 will not be aware of that transaction and will still be able to reserve the same cell.

The first issue should be reduced to a minimum with the mechanism of the cell buffer. The second issue can be solved by a reactive mechanism like housekeeping that is being evaluated in the next section.

C. Comparisons with Housekeeping

As seen in Section II-B, housekeeping is another solution to solve the problem of colliding cells. However, it is a reactive solution, which allows the collision to occur while our solution is proactive and prevents collisions to happen in the first place.

In Figures 5 and 6, we compare OTF with housekeeping alone to OTF with housekeeping and MECB.

The results show in particular a decrease of the peak of at least 60% of the number of colliding cells and also a 60% of colliding packets in the bootstrap phase.

Housekeeping alone will take some extra time to relocate the defected cells while MECB, with less colliding cells, will speed up the convergence of housekeeping.
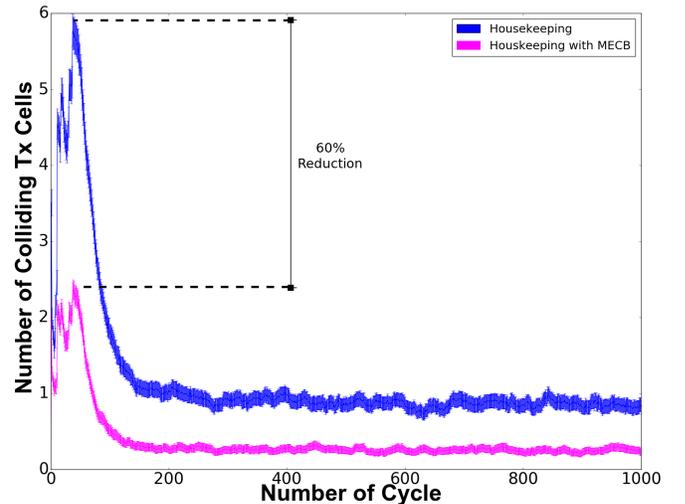

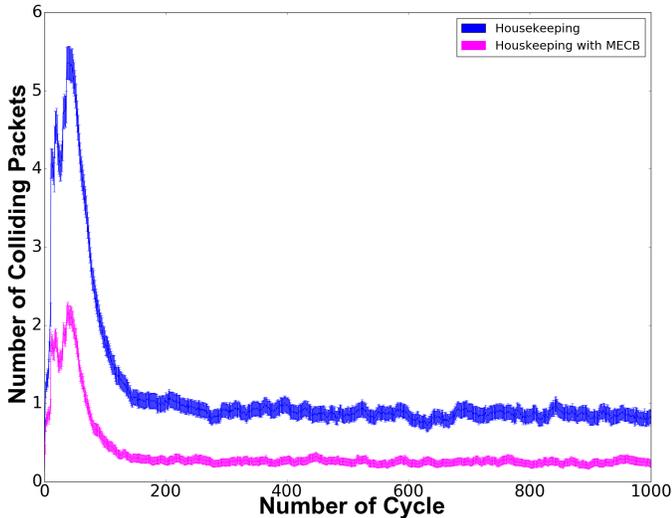Fig. 5. Number of colliding Tx cells as a function of cycles, OTF with housekeeping

Fig. 6. Number of colliding packets as a function of cycles, OTF with housekeeping and MECB

V. CONCLUSION

In this paper, we have considered the problem of conflicting TSCH schedules computed by current distributed scheduling algorithms. We have explained why a distributed scheduling may lead to collisions, yet this kind of scheduling still has a lot of benefits in terms of the time required to reserve a cell and the complexity of the reservation process. We have proposed a solution to reduce conflicting allocations in TSCH schedules created with 6TiSCH distributed algorithms. The idea is to achieve local mutual exclusion by making nodes able to overhear 6P transactions exchanged by their neighbors. This mechanism does not involve any overhead, provided the transactions are sent in shared slots. The 6P transactions have been also augmented with a short-term memory of past allocated cells, which makes overhearing more robust for the neighbors that may miss some transactions because of intermittent variations in the quality of their wireless links.

The performance of our approach has been evaluated through simulations. The results have shown significant reduction in the number of colliding packets.

In the future work, we plan to study the effects of traffic on different scheduling schemes. With the emergence of algorithms for traffic aware scheduling in IEEE802.15.4e networks, we also expect to improve this kind of protocols.

REFERENCES

[1] *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003): IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications f.* IEEE, 2006. [Online]. Available: https://books.google.fr/books?id=S-13nQAACAAJ

[2] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, Apr. 2012.

[3] Q. Wang, X. Vilajosana, and T. Watteyne, "6top Protocol (6P)," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-6top-protocol-05, May 2017, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-6top-protocol-05

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012. [Online]. Available: https://tools.ietf.org/html/rfc6550

[5] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, A. Grieco, and T. Engel, "On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks," *IEEE Sensors Journal*, vol. 16, no. 2, Jan. 2016.

[6] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, and K. S. Pister, "Simple Distributed Scheduling With Collision Detection in TSCH Networks," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 5848–5849, May 2016.

[7] F. Theoleyre and G. Z. Papadopoulos, "Experimental validation of a distributed self-configured 6tisch with traffic isolation in low power lossy networks," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '16, New York, NY, USA, 2016. [Online]. Available: http://doi.acm.org/10.1145/2988287.2989133

[8] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, "Distributed pid-based scheduling for 6tisch networks," *IEEE Communications Letters*, vol. 20, no. 5, pp. 1006–1009, 2016.

[9] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation," *IEEE Internet of Things Journal*, vol. 2, no. 6, Dec. 2015.

[10] R. Soua, P. Minet, and E. Livolant, "Wave: a distributed scheduling algorithm for convergecast in ieee 802.15. 4e tsch networks," *Transactions on Emerging Telecommunications Technologies*, 2015.

[11] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," RFC 8180, May 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8180.txt