Fully autonomous scheduling for heterogeneous traffic through opportunistic transmissions

Andreas R. Urke

I. Introduction

Proposal for an area of investigation which might be applicable for a bachelor project or master thesis.

II. Background

A. TSCH

The Time Slotted Channel Hopping (TSCH) medium access control has garnered significant attention in the wireless sensor network research community. It combines channel-hopping, to increase resilience against wireless fading events, with contention-free allocation of time-slots. It is selected as the MAC in IETFs ongoing work on 6TiSCH - an IPv6-enabled stack for industrial wireless low-power networks [1].

B. Scheduling

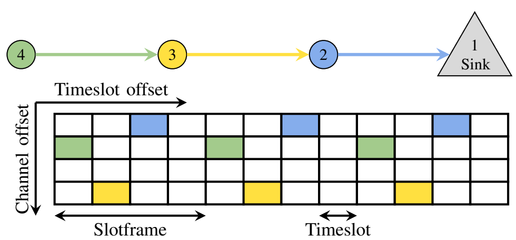

Fig. 1. Simple wireless network topology with example TSCH schedule

Nodes in a TSCH network operate according to a schedule implemented as one or more *slotframes*. These repeat over time and dictate if a node is allowed to transmit or receive. Figure 1 shows an example network with an accompanying schedule. Time is divided into timeslots on the horizontal axis, while the available channels are shown in the vertical. A *cell* is defined by a pair of timeslot- and channel-offset coordinates, and it allows for transmitting or receiving one frame with an optional acknowledgment.

Key in TSCH is the establishment and maintenance of the slotframe content, i.e. how cells are allocated among nodes in the network. This is achieved by a *scheduler*, which may operate in a centralized, collaborative or autonomous fashion. With an autonomous scheduler, nodes create their schedules independently without any information exchanged between schedulers on the nodes. This simplifies the configuration, avoids signaling overhead, and increases the fault tolerance since no signaling is needed to utilize new links.

The current state-of-the-art autonomous scheduler Orchestra [2] lets nodes utilize their own and neighbor IDs to allocate cells. When setup in a contention-free configuration, Orchestra maps each node ID to a unique cell used by the given node to transmit. For nodes wanting to receive, they simply add a RX cell matching the transmitter ID. The coordination is obtained through a routing protocol such as RPL.

III. Problem statement

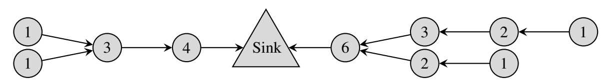

Fig. 2. Funneling effect, number indicates cumulative offered traffic

Typical autonomous schedulers statically assign a fixed amount of resources to each node and does not adapt to variations in traffic intensity. This leaves it vulnerable to reduced performance in e.g. applications with heterogeneous traffic intensity, scenarios with differing link qualities yielding a varying demand for retransmissions, and convergecast applications where a funneling effect increases traffic intensity close to the destination, as illustrated in Figure 2.

IV. Related Work

A survey on autonomous scheduling in [3] lists three which focus on traffic adaptations: TESLA [4], PAAS [5] and e-TSCH-Orch [6]. However, all of these require exchange of information which: 1) Adds overhead for link establishment, which introduces convergence periods, and 2) Increases complexity, making it more difficult for operator to understand the network operations, i.e. the "visibility problem" [3], thus reducing some of the key benefits of autonomous scheduling.

V. Proposal

The essence of the proposal is for node to opportunistically try extra transmissions after their regular transmission. I.e. if node 2 is scheduled to transmit in a cell at timeslot 2, it may also utilize timeslot 3 and onwards for extra transmissions until its queue is empty. An overview schedule can be seen in Figure 3. This immediately creates three key challenges which we will address with a novel combination of solutions:

Firstly, how does the receiving node know that its neighbor want's to utilize its extra cells or not? Here the receiver can exploit two inputs:

1) The transmitting node set the Frame Pending bit in the 802.15.4 header, indicating it has additional packets in its queue. This behavior is according to the standard.
2) The receiver does not receive a valid packet, yet senses there is energy on the channel. This indicates a possible failed transmission attempt which induces a retransmission. (see [7]).

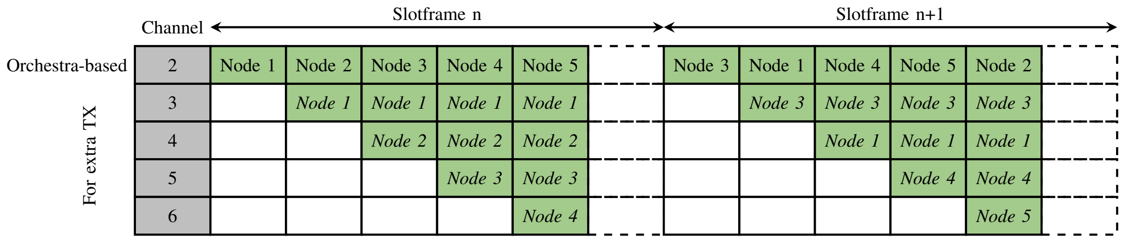

Fig. 3. Orchestra-based application slotframe with additional cells for extra transmissions in *italics*.

As long as any of these signals are true, the receiver should listen on the next timeslot. This should keep idle listening to a minimum while allowing a neighbor to transmit an arbitrary amount of additional packets.

Secondly, the cells utilized for the extra transmissions may collide with cells already in the schedule of either the transmitter or the receiver. Consider for example if node 2 transmits towards node 3 at timeslot 2, and wants an extra transmission in timeslot 3. However, if node 3 has its own regular transmission cell in timeslot 3, it would not be able to listen for the extra transmission.

Such synchronization can be avoided by adding time as an input when calculating the timeslot offset. Instead of the timeslot being scheduled according to node id alone, it would be set by node id + time (i.e. ASN (absolute slot number, a global ever-increasing timeslot counter starting at 0 when network is deployed)). This causes every slotframe to be unique as shown in the bottom of Figure 3. Such a technique is employed in the ALICE [8] scheduler to avoid continuously overlapping cells. This may also be exploited to avoid synchronization in the channel calculation.

Thirdly, since the extra transmissions are decided autonomously, there is a chance that the they will interfere with the transmissions of other nodes. To avoid this we can exploit the low band occupancy of Orchestra (only channels 0-2 are by default used): Any cells intended for extra transmissions will utilize channel 3-15 (or a sub-set). Thus there is no increased contention for the regularly scheduled cells. Notice how the extra transmissions cells in Figure 3 are in separate channels.

There is however still a possibility of multiple nodes having extra transmissions at the same time, and also selecting the same channel, all the while being in interference range of each other. The solution to this problem is an open item: As above, time/ASN could be used as input into the channel selection algorithm, to avoid persistent synchronization and collisions between nodes. Another option could be to tie the channel selection to the rank, e.g. such that nodes on neighboring ranks does not interfere. However, that would increase the probability of interference within a rank. Lastly, it could be that selecting based solely on node ID yields sufficient performance (probability of collisions would depend on traffic pattern and profile, network density, and number of channels employed).

This proposal is a fully autonomous approach, i.e. no additional information needs to be exchanged between nodes. It can in theory be applied to any scheduler. The proposal involves an opportunistic addition of cells, thus the performance of these cells are probabilistic. The logical step is therefore to apply it onto Orchestra or ALICE, which would make it inline with PAAS, e-TSCH-Orch and TESLA (which are all based on Orchestra).

VI. Open questions

Which channel selection algorithm should be utilized?

What is the individual contribution of each of the three mechanisms in the proposal?

How does the proposal impact reliability and latency, vs. increased energy consumption? At which limits of the parameters (network density, traffic intensity, etc.) does the proposal provide a benefit, and in which does it not. Or in other words, what are its applications? How does it perform in the scenarios treated by Orchestra/ALICE?

Note that the essence of the proposal is for nodes to opportunistically grab resources. Any gained knowledge from an evaluation, and the proposal itself, is probably generic such that it is applicable to any other scheduling scheme which could accommodate opportunistic extra transmissions.

References

[1] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, “IETF 6TiSCH: A Tutorial,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[2] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’15, ACM. New York, NY, USA: ACM, Nov. 2015, pp. 337–350. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809714

[3] A. Elsts, S. Kim, H. Kim, and C. Kim, “An Empirical Survey of Autonomous Scheduling Methods for TSCH,” *IEEE Access*, vol. 8, pp. 67 147–67 165, 2020.

[4] S. Jeong, J. Paek, H. Kim, and S. Bahk, “TESLA: Traffic-Aware Elastic Slotframe Adjustment in TSCH Networks,” *IEEE Access*, vol. 7, pp. 130 468–130 483, Sep. 2019.

[5] J. Jung, D. Kim, J. Hong, J. Kang, and Y. Yi, “Parameterized slot scheduling for adaptive and autonomous TSCH networks,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018, pp. 76–81.

[6] 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

[7] G. Cena, S. Scanzio, L. Seno, A. Valenzano, and C. Zunino, “Energy-efficient link capacity overprovisioning in time slotted channel hopping networks,” in *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*, 2020, pp. 1–8.

[8] S. Kim, H. Kim, and C. Kim, “ALICE: Autonomous Link-based Cell Scheduling for TSCH,” in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2019, pp. 121–132.