Poster: Is Less More? Comparing TSCH and CSL on the FIT IoT-LAB

Konrad-Felix Krentz
konrad-felix.krentz@sit.fraunhofer.de
Fraunhofer SIT | ATHENE
Heilbronn, Germany

Michael Eckel
michael.eckel@sit.fraunhofer.de
Fraunhofer SIT | ATHENE
Darmstadt, Germany

Abstract

Timeslotted channel hopping (TSCH) is a standardized medium access control (MAC) protocol for IEEE 802.15.4 networks. Its approach is to proactively avoid internal interference through careful scheduling. Conversely, coordinated sampled listening (CSL), another standardized MAC protocol for IEEE 802.15.4 networks, works in a contention-based manner. Thus far, only Zhou et al. empirically compared TSCH and CSL to the best of our knowledge. However, they tested implementations of TSCH and CSL that lacked recent enhancements, and they only employed small-scale Cooja simulations. Our poster expands on their work by reporting on real-world benchmarks of up-to-date implementations of TSCH and CSL. Diverging from Zhou et al.'s results, CSL reduced duty cycling by up to 80.0% compared to TSCH. On the other hand, CSL generally incurred longer delays and higher jitters than TSCH, diverging from Zhou et al.'s results, as well. When accelerating CSL's wake ups to match the delays and jitters of TSCH, CSL still lowered duty cycling between 50.2% and 54.9%, depending on the traffic rate.

CCS Concepts

• **Networks → Link-layer protocols**; **Network experimentation**; • **Computer systems organization → Sensor networks**.

Keywords

Low-power wireless networks, channel hopping, security

ACM Reference Format:
Konrad-Felix Krentz and Michael Eckel. 2018. Poster: Is Less More? Comparing TSCH and CSL on the FIT IoT-LAB. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Motivation

A common theme in networking and real-time systems is that determinism provides for sustained performance under load, yet sacrifices performance under other workloads. Thus, similar trade-offs are to be expected when choosing between the timeslotted channel hopping (TSCH) and coordinated sampled listening (CSL) medium access control (MAC) protocols. TSCH and CSL are two standardized MAC protocols for IEEE 802.15.4 networks. Both are duty-cycling MAC protocols, i.e., they leave the radio chip in an energy-saving sleep mode most of the time. More specifically, TSCH belongs to the class of synchronous duty-cycling MAC protocols [6]. These use network- or cluster-wide time synchronization to schedule sleeping periods and often use time synchronization to avoid intra-network interference, too. Asynchronous duty-cycling MAC protocols, such as CSL, forgo network- and cluster-wide time synchronization. Those MAC protocols typically rely on clear channel assessments (CCAs) and exponential backoffs to resolve contention.

To date, performance comparisons between TSCH and CSL are sparse. Only Zhou et al. seem to have worked in this direction [2]. Their results suggest that TSCH does less duty cycling than CSL and that CSL attains shorter delays than TSCH. Yet, they only used 4-node Cooja simulations. Moreover, their implementations lacked enhancements, such as autonomous scheduling for TSCH [8], or machine learning (ML)-based channel hopping for CSL [3].

Our contribution is to benchmark full-fledged implementations of TSCH and CSL on the FIT IoT-LAB [1]. More specifically, we benchmark Contiki-NG implementations of ALICE, an advanced autonomous scheduler for TSCH [7], and Hasso Plattner Institute MAC (HPI-MAC), a version of CSL, which features denial-of-sleep resilience, compromise resilience, ML-based channel hopping, adaptive synchronization, as well as burst forwarding [3, 4].

2 Experimental Setup

The experiments were conducted using 31 OpenMote-Bs of the FIT IoT-LAB site in Strasbourg. In each experiment, one OpenMote-B acted as IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) root and the other 30 OpenMote-Bs sent User Datagram Protocol (UDP) datagrams towards that root. Every sender initially waited 15 min and then sent UDP datagrams with random pauses T in between for 15 min. T was uniformly distributed over $[0, 2d]$, where d was 2.5, 5, 10, 20, and 40 seconds in successive experiments. All experiments were run with four different configurations, namely with (i) ALICE, (ii) HPI-MAC and blind channel hopping, (iii) HPI-MAC and ML-based channel hopping, and (iv) HPI-MAC, ML-based channel hopping, and an increased wake-up rate of 32 Hz instead of 8 Hz. For every distinct combination of d and configuration, 10 repetitions were run. Throughout, ALICE and HPI-MAC retransmitted the UDP datagrams 3 times at most, Contiki-NG's queuebuf accommodated 64 frames maximum, the IEEE 802.15.4 security level was 1, and RPL operated in storing mode. Other configuration settings were left at their defaults. During the experiments, the OpenMote-Bs logged transmissions, receptions, as well as their time spent for

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

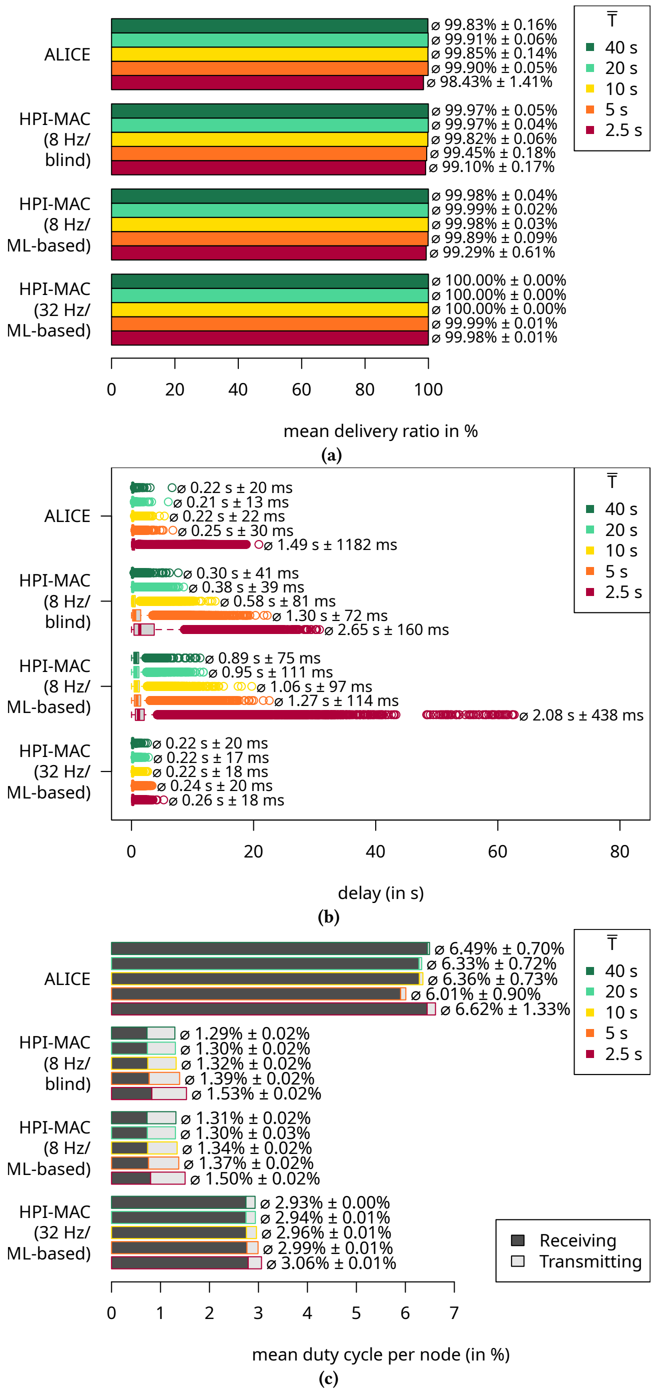

Figure 1: Experimental results. Behind each bar, we report on the sample means and the associated 95% confidence intervals over the sample means from the 10 repetitions per traffic rate and configuration. At a wake-up rate of 32 Hz, HPI-MAC performs similar or better than ALICE across the board.

transmitting and receiving. The experimental scripts are available at https://github.com/kkrentz/contiki-ng/tree/ewsn2026.

3 Results

Figure 1 shows the results. ALICE and HPI-MAC approach or reach delivery ratios of 100%, respectively. HPI-MAC's ML-based channel hopping slightly improves delivery ratios over blind channel hopping, yet generally incurs longer delays due to waiting for a receiver to wake up on a channel that the ML algorithm has proposed. At $\bar{T} = 2.5$ s, by contrast, ML-based channel hopping reduces the mean delays, apparently because reducing retransmissions is particularly beneficial under contention. Compared to ALICE, 8 Hz HPI-MAC incurs much longer delays and much higher jitters. We attribute this to HPI-MAC running into collisions and backing off exponentially as a result. On the other hand, ALICE does much more duty cycling than 8 Hz HPI-MAC. The traffic rate has only a small effect on duty cycles. This is because the bulk of the time spent in receive mode stems from periodic wake ups. HPI-MAC, in particular, also spends much time broadcasting so-called `HELLO`s for neighbor discovery. In the delay- and jitter-normalized 32 Hz/ML-based configuration, HPI-MAC still reduces duty cycling significantly compared to ALICE. Also, in this configuration, HPI-MAC seldomly loses a UDP datagram as accelerating wake ups reduces chances for collisions.

4 Conclusions and Future Work

While ALICE has attained short delays and low jitters, HPI-MAC has excelled at reducing duty cycles. At a wake-up rate of 32 Hz, HPI-MAC has provided superior delivery ratios and duty cycles than ALICE, as well as superior delays and jitters at the highest traffic rate. Besides, since synchronous MAC protocols are susceptible to time synchronization attacks, denial of sleep, as well as node compromises [4, 5, 9], HPI-MAC is favorable in security-sensitive scenarios, too. For future work, we plan to reuse HPI-MAC's local information about the wake-up times of neighboring nodes for providing network-wide time synchronization.

Acknowledgments

The ATHENE National Research Center for Applied Cybersecurity supported this research.

References

[1] Cedric Adjih et al. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *Proc. of WF-IoT*. IEEE, 459–464. doi:10.1109/WF-IoT.2015.7389098

[2] Jianwei Zhou et al. 2014. Comparison of IEEE 802.15.4e MAC features. In *Proc. of WF-IoT*. IEEE, 203–207. doi:10.1109/WF-IoT.2014.6803159

[3] Konrad-Felix Krentz et al. 2021. Multi-armed bandit-based channel hopping: Implementation on embedded devices. In *Proc. of MLN'2021*. Springer, 29–47. doi:10.1007/978-3-030-98978-1_3

[4] Konrad-Felix Krentz. 2019. *A denial-of-sleep-resilient medium access control layer for IEEE 802.15.4 networks*. Ph. D. Dissertation. Potsdam University. doi:10.25932/publishup-43930

[5] Nicolas López et al. 2019. On the performance of 6LoWPAN using TSCH/Orchestra mode against a jamming attack. In *Proc. of CHILECON*. IEEE, 1–5. doi:10.1109/CHILECON47746.2019.8988035

[6] Pei Huang et al. 2013. The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials* 15, 1 (2013), 101–120. doi:10.1109/SURV.2012.040412.00105

[7] Seohyang Kim et al. 2019. ALICE: Autonomous link-based cell scheduling for TSCH. In *Proc. of IPSN '19*. ACM, 121–132. doi:10.1145/3302506.3310394

[8] Simon Duquennoy et al. 2015. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proc. of SenSys '15*. ACM, New York, NY, USA, 337–350. doi:10.1145/2809695.2809714

[9] Wei Yang et al. 2014. Time synchronization attacks in IEEE802.15.4e networks. In *Proc. of IIKI 2014*. IEEE, 166–169. doi:10.1109/IIKI.2014.41