Lu Wang, Andreas Reinhardt: A Simulative Study of Network Association Delays in IEEE 802.15.4e TSCH Networks. In: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1-3, 2017.

A Simulative Study of Network Association Delays in IEEE 802.15.4e TSCH Networks

Lu Wang
Institut für Informatik, TU Clausthal
Clausthal-Zellerfeld, Germany
lu.wang.1@tu-clausthal.de

Andreas Reinhardt
Institut für Informatik, TU Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

***Abstract*—Several communication protocols based on channel hopping have been proposed for wireless sensor networks. One prominent example is IEEE 802.15.4e, which relies on the concept of time-slotted channel hopping (TSCH). TSCH effectively mitigates poor channel conditions (caused, e.g., by fading) by means of channel changes according to a pre-defined hopping sequence. In order for nodes to participate in TSCH networks, however, a synchronization procedure is necessary; to this end, newly booted nodes will listen on a default channel for beacons transmitted by nodes that are already synchronized. This process suffers from two drawbacks: Firstly, synchronized nodes do not continuously send beacons, but wait for a time period in-between their beacon transmissions. Secondly, in large networks, beacon collisions are very likely to occur, which also slow the association process down. In this paper, we study both impacts in simulated scenarios in order to derive potential for improvement.**

I. Introduction

Channel-hopping medium access control (MAC) protocols for wireless sensor networks (WSNs) target to attain increased communication reliability, even in harsh environments [1, 2]. The IEEE 802.15.4e standard [3] proposes one such MAC protocol called time-slotted channel hopping (TSCH). Nodes in a TSCH network simultaneously change their wireless communication channel according to a pre-defined hopping sequence. They remain on a given channel only for a short time, during which data transfers can take place between synchronized nodes. Newly joining nodes are generally unaware of the channel hopping sequence, however. Thus, they need to wait for announcements carrying synchronization information, so called *Enhanced Beacons* (or *EB*s for short). EBs are transmitted periodically by all nodes already synchronized to the TSCH network, and sent on the wireless channel selected according to the channel hopping sequence.

The inclusion of a new device into an existing network may lead to a considerable energy consumption when choosing an unfavorable strategy. The preferred approach (cf. [3]) to obtain synchronization information from a TSCH network is *passive scanning*, i.e., configuring the radio transceiver to a given channel and listening for EBs. Upon reception of an EB, the newly introduced node can extract all required information about the time slotting and the channel hopping sequence. Once this information has been processed, the node is synchronized to the TSCH network and can begin periodically transmitting EBs with synchronization information itself.

978-1-5386-2723-5/17/$31.00 ©2017 IEEE

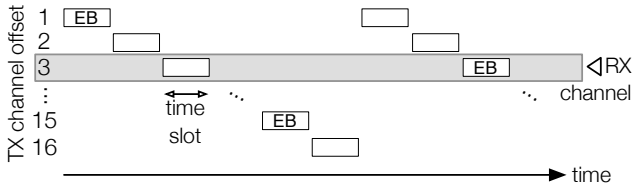

Fig. 1. Simplified example of the synchronization process in a TSCH network.

An example of how passive scanning is used to join a TSCH network is shown in Fig. 1. A single sender (TX) node follows a channel hopping schedule[1] and sends EBs periodically. The receiver node (RX) configures its wireless transceiver to use one channel to listen for EBs (channel offset 3 in the figure). Consequently, beacons transmitted on other channels cannot be used to synchronize the node to the network. It also becomes apparent from the figure that synchronized nodes do not send EBs in every time slot. Besides avoiding perpetual beacon collisions this way, sending beacons less frequently also frees up more of the spectrum for actual data transmissions.

In this paper, we thus conduct an analysis of the delays encountered when adding nodes to an existing (synchronized) TSCH network. Throughout the remainder of this paper, we refer to the time period between activating the node (turning on its power supply) and its successful synchronization to the TSCH networks as the *joining time* (JT). The same metric has been analyzed by De Guglielmo et al. in [4], yet only in an analytical fashion. In contrast, we present results achieved from simulations of an actual TSCH implementation in this paper. Thus, we expect to complement the state of the art by practically-oriented insights.

II. Time-Slotted Channel Hopping

As shown in Fig. 1, time is divided into *time slot*s in TSCH. Using slotted time requires a tight clock synchronization among participating nodes. This is accomplished through transmission of EBs by synchronized nodes. All relevant time information (including the length of a time slot and the ASN) is contained in an EB, and can be extracted by nodes wishing to update their time information upon its receipt.

[1]For the sake of simplicity we assume sequential channel hopping here. The actual channel hopping sequence in IEEE 802.15.4e is determined by the output of a linear feedback shift register [3].

TSCH relies on the concept of channel hopping. In order to determine the next wireless channel to use, TSCH makes use of the *Absolute Slot Number* (ASN), a unique value that is maintained throughout the entire network lifetime. Initially set to zero, the ASN is incremented in each time slot. The channel to use for communications is then selected according to Eq. (1), where channelOffset takes a fixed value to enable the co-existence of several TSCH networks, and n_{Freq} is the number of available frequencies (16 if all channels of IEEE 802.15.4 in the 2.4 GHz band are used). The array F contains the list of channels in the order defined by the hopping sequence [3].

$$\text{frequency} = F\big[(\text{ASN} + \text{channelOffset}) \mod n_{Freq}\big] \quad (1)$$

III. Joining a TSCH Network in Practice

The process of joining a TSCH network is the area under investigation in this paper. As outlined above, a node can only successfully synchronize to the network when it receives an EB on the channel it is scanning. However, the IEEE 802.15.4e standard does not define a transmission strategy for EBs. Thus, synchronized nodes can send EBs at their own pace, as long as they are being transmitted on the frequency defined by the channel hopping sequence. The used beaconing strategy, however, has an immediate impact on the JT for new nodes.

For our analysis of how the beaconing interval impacts the JT distribution in a realistic setting, we rely on the implementation of TSCH as part of the Contiki operating system[2]. More specifically, Contiki's `RPL-TSCH` example has been used as the foundation for our analysis. Its default configuration, including its EB interval of 4 seconds, has been maintained[3]. However, this interval represents an upper bound; in the default configuration, the inter-EB interval is determined randomly in the range between $[0.75, 1.0)$ of the specified value. At the used time slot length of 15 ms[4], the ASN value is incremented $66.\bar{6}$ times each second, equivalent to 200–266 channel changes in-between EB transmissions.

In case the synchronized network is composed of multiple nodes, each of these devices will periodically send EBs; their total number is thus proportional to the size of the network that is already synchronized. In case the number of neighbors (i.e., nodes in direct communication range) grows too large, however, EB collisions are likely to occur. We thus expect joining times to roughly follow the graph depicted in Fig. 2; our actual analysis is presented in the next section.

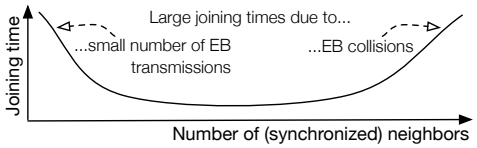

Fig. 2. Expected trend of TSCH network joining times.

[2]Cf. http://www.contiki-os.org

[3]At the time of analysis, the TSCH EB period was set to 4 s; meanwhile the value has been increased to 16 s for the TSCH implementation in Contiki.

[4]Default time slot length for the Zolertia Z1 platform, which we used here.

IV. Simulative Joining Time Evaluation

Our simulation results for the network joining time when a newly added node joins a synchronized network are presented as follows. All simulation results were attained from the COOJA simulator [5] with at least seven repetitions conducted for each simulated setting. The `RPL-TSCH` code was compiled for the Zolertia Z1 using a patched version the MSPGCCX compiler to overcome the platform's memory limitations.

A. Fixed EB Transmission Period

Our first experiment is the analysis of the JT when a new node is introduced into an existing synchronized network. In order to get a baseline of the JT distribution, we use a fixed EB transmission period of 4 s with no random variation. The simulated network is composed of between 1 and 60 nodes in randomized locations, but all within at most two-hop connectivity of each other. Once the existing network is fully synchronized, a single node is booted. We measure the time required until the newly added node has been synchronized and begins sending EBs itself. Results for this simulation are shown in the plot in Fig. 3, where the size of the synchronized network is stated on the x-axis, and error bars indicate encountered extremal values. The diagram indicates that the time for joining a single node network is comparably coherent at around 70 s. As soon as more than one node is actively sending EBs, the figure shows decreasing JTs, but their larger variance also becomes apparent. This can likely be attributed to the point in time at which the devices in the synchronized network have been booted, as this influences their EB transmission periods. A second insight gained from the analysis is that networks with high node density (visible from the graph when 50 or 60 devices were in proximity), a larger JT mean value and variance can be observed again. We attribute this to persistent EB collisions (due to the fixed EB transmission interval) which slow the synchronization process down. The results thus confirm our hypothesis in Fig. 2.

B. Alternative EB Transmission Periods

In our second experiment, we again assume a synchronized network of varying size (between one and sixty nodes) and analyze the JT when adding a single new device to the network. Moreover, we also modify the EB transmission period between different values: Besides the fixed period of 4 s (like in the experiment described in Sec. IV-A), we also analyze random EB periods in the value ranges between $[3.5, 4)$ s, $[3.25, 4)$ s, and $[3, 4)$ s (as in the `RPL-TSCH` implementation).

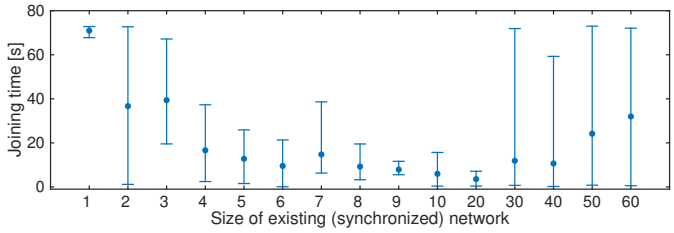

Fig. 3. JT results with a synchronized network of the given size, a fixed EB period of 4 s and a single node joining the network.

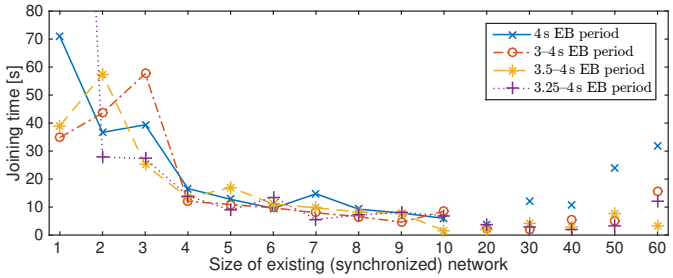

Fig. 4. JT results with a synchronized network of the given size, EB periods in the specified ranges, and a single node joining the network. Note: The average joining time is 307 s in the network of one advertising node with a random period between 3.25 and 4 seconds. For the sake of better readability, we did not mark this point in the figure.

The results of these minor deviations from the baseline value are shown in Fig. 4, in which only the average JTs have been plotted for visual clarity. Even though a similar trend can be observed across all four evaluated configurations, differences between the different EB periods become apparent when the synchronized network contains more than 20 nodes. In these cases, applying randomness to the EB periods leads to smaller JTs, most likely caused by the fact that more collisions occur in the same time slots whereas other EB transmissions succeed without collisions and thus allow for the successful synchronization of the new nodes.

A second insight from the figure is that JTs experience a significant drop when moving from a very small neighborhood (with three or less synchronized neighbors) to an area with four or more neighbors. Compared to a setting with just three neighbors, the JT in the network of four advertising nodes is decreased by 47–48% when the period ranges between 3.25 s or 3.5 s and 4 s. Even more significantly, the JT reduces by 79% when a random period between 3 and 4 s is used.

C. Interpretation of the Results

Based on our analysis of network joining times in different scenarios, we have gained insights into the synchronization process that allow us to derive directions for EB transmission strategies. Most importantly, the absence of a central entity to schedule EB transmissions necessitates a distributed algorithm to decide on EB transmissions, which can only take locally available information into consideration. Our evaluations have shown the number of neighbors to be an important factor in achieving quick network synchronization. In particular, we have observed comparable JT performances for network densities in the range from 4 to 20 neighbors, regardless of the random value used to modify this period. However, both in sparse (three or less synchronized neighbors) or dense (50 or more synchronized neighbors) topologies, periodic EB transmissions (with or without a random offset) show a degraded JT performance. The development of an EB transmission strategy that takes the number of neighbors into consideration when choosing its EB period could thus be an avenue of research to pursue, possibly applying concepts that adapt transmissions according to the traffic around a potential EB transmitter, such as the notion of *polite gossip* proposed in [6].

As mentioned before, a closer analysis of introducing random delays into the EB period also appears to be a promising topic for future work. Particularly in large networks, perpetual EB collisions can arise when using a fixed period for their transmission. A conjoint optimization of both EB transmission period and the amount of randomness in the EB transmissions would thus need to be conducted in order to achieve reliable and repeatable synchronization performance for newly joining nodes.

V. Conclusion

TSCH is one of the core concepts of IEEE802.15.4e. It combines the notions of time slots and multi-channel hopping to achieve low-power operation while enabling high reliability in WSNs. In this paper, we have analyzed the performance of the joining process through measurements of the synchronization delay when running the TSCH implementation that ships with the Contiki operating system. Through the analysis of network joining times in different configurations, we have identified both the neighborhood size as well as the EB transmission period as two important influences on how quickly a node is synchronized to the network.

In case of the network with a fixed EB period of 4 s, the average joining time drops by more than 90% when the number of a node's neighbors is increased from just 1 to 20. However, in larger neighborhoods, EB collisions become more prevalent and represent a major factor for worsening joining times. Our analysis has furthermore shown that random delays in EB transmissions have a slightly positively impact the joining times in dense networks because they lower the probability of continuously colliding EBs. In the future, we plan to extend our simulations by new parameter configurations and also evaluate JTs when multiple nodes join the network simultaneously.

References

[1] T. Watteyne, A. Mehta, and K. Pister, "Reliability through Frequency Diversity: Why Channel Hopping Makes Sense," in *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2009, pp. 116–123.

[2] T. Watteyne, S. Lanzisera, A. Mehta, and K. S. Pister, "Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2010, pp. 1–5.

[3] IEEE, "802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC Sublayer," 2012.

[4] D. De Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, "A Performance Analysis of the Network Formation Process in IEEE 802.15.4e TSCH Wireless Sensor/Actuator Networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2014, pp. 1–6.

[5] J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. J. Marrón, "COOJA/MSPSim: Interoperability Testing for Wireless Sensor Networks," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SimuTools)*, 2009, pp. 27:1–27:7.

[6] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A Self-regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in *Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 1–14.