Mobility-Focused Joining in TSCH Networks

Oliver Nielsen*, Lucas Koefoed Schnügger*, Charalampos Orfanidis, Xenofon Fafoutis
DTU Compute, Technical University of Denmark (DTU), Denmark
{`s164917, s164924, chaorf, xefa`}@dtu.dk

***Abstract*—TSCH is a MAC protocol able to provide reliable wireless communication to IoT networks. For applications where mobility is necessary research has shown that TSCH presents a suboptimal performance. One of the main issues is that the time required for a device to join a TSCH network is unpredictable and theoretically has no upper limit. The joining process to a TSCH network can take several minutes in some cases which can ensue significant delay and power overheads. In this paper we examine how TSCH can be adapted to support mobile applications, by improving the joining and re-joining procedure. Relevant information about the network topology is aggregated throughout the network to be utilized during the joining process. The results show that the joining time can be reduced by 50% and the re-joining time by 90%.**

***Index Terms*—Internet of Things, TSCH, Reliability, Dependable IoT**

I. INTRODUCTION

Time-Slotted Channel Hopping (TSCH) is a Media Access Control (MAC) protocol which is able to provide reliable wireless communication between low-power devices [1]. The main feature of TSCH is the utilization of a channel hopping function to mitigate the impact of noise, interference and multipath fading which are some of the most well-known reasons for deteriorating network performance in IoT networks [2]. To ensure reliability, TSCH offers a synchronization scheme among all nodes which have joined the TSCH network upon which a schedule is constructed. The schedule dictates if a node should be receiving or transmitting a packet on a specific channel or just sleep to conserve energy. TSCH is designed in a modular manner and it is commonly customized based on the application scenario since there might be different network topologies, traffic rates or routing requirements. At the moment there are few approaches which incorporated TSCH in scenarios including mobile communication. Unless there is a good network coverage, which is achieved with an adequate number of nodes, TSCH performance is far from reliable when mobile communication is required [3], [4].

Taking a closer look at the TSCH protocol it is clear that the process of joining a network is very inefficient. A joining node will listen for Enhanced Beacons (EBs), which is a required packet a new node must receive in order to join a TSCH network, according to the IEEE 802.15.4e standard [5]. During the joining process the joining node will constantly be scanning for EBs on random channels. Depending on the network size and the timing of the EB broadcasts, it may take several minutes for the joining node to capture an EB. Furthermore, the joining node should listen for multiple EBs in order to select the best parent, interrupted only by a timeout after which the node will associate to the best known parent. This means that the energy-consuming radio is constantly active during the entire joining process, and that more suitable parent-nodes may be overheard. Thus, this incurs significant overhead in terms of time delay and consumed energy for new nodes joining a TSCH network.

In this paper we propose a mechanism to improve the association of mobile nodes to TSCH networks by decreasing the joining time. The mechanism is based on a scheme which uses the existing infrastructure to aggregate relevant information about the network to predict the time and channel of future EB broadcasts. At the same time we are concerned about the overhead this mechanism may introduce in terms of interference, energy consumption and time delay. To this end, we investigated through several scenarios how mobile but also static nodes can be benefited from such a mechanism. Improving the joining process will benefit both static and mobile nodes but it is especially significant for the latter since they are more likely to disassociate and re-associate to the network during operation. To evaluate the proposed mechanism we used Contiki-NG and the Cooja simulator [6], which is a cross layer network simulator including different levels from physical to application layer. As a benchmark we used the default version of TSCH as it is described by IETF in [7].

II. RELATED WORK

Al-Nidawi et al. [8], evaluated TSCH protocol under mobility scenarios and found that the the scanning process is increased by the long channel offset and the mobile nodes cannot discover coordinators. Furthermore they recognise that the function of association and disassociation is not flexible and affects the performance of the network. In a follow up paper [9], the same authors propose to encapsulate EB packets into ACK packets and transmit them to a fixed channel. With this approach they manage to decrease the Radio Duty Cycle (RDC) by 30% on average and increase the connectivity by 25%. A combination of static and mobile nodes topology is considered to illustrate that TSCH can achieve adequate connectivity with the premise that there are adequate amount of nodes, either stationary or mobile, to enable a proper coverage in [3]. However, the increased amount of packets, used to ensure the synchronization have an impact on the energy consumption and the time delay.

---

* The first two authors contributed equally to this work.

Several TSCH schedules were evaluated on different mobility patterns and demonstrated that the performance is not reliable due to coverage issues, or the schedule cannot handle the mobility and most of the times is outdated in [4]. In [10] it is introduced a framework called active connectivity, which is able to regulate physical characteristics, like the speed, of autonomous vehicles to maintain connectivity in TSCH networks. This has an impact on other aspects of the application such as the time to cover a certain distance and in turn the overall cost which is investigated properly.

A thorough survey on mobility models and how they affect the RPL protocol is presented in [11]. The authors provide a taxonomy and a classification of the mobility models and a comparison based on their main specification. They evaluate RPL using numerous mobility models to quantify the power consumption, reliability, latency and control overhead. Barceló et al. introduce an extension of the 6TiSCH routing in [12] focusing on scenarios where static and mobile nodes co-exist. In static nodes the routing is handled by the regular 6TiSCH but in mobile nodes it is utilized the end-to-end reliability and a blacklisting function based on the position of the node. They mention that the reliability between the static and mobile nodes is increased even in the presence of high positioning errors in comparison with previous routing approaches. In [13] the authors try to improve RPL's mobile performance using a Software Defined Network (SDN) technique where a central controller has a holistic view of the network, can predict the handovers, and update the routing tables. The proposed approach can achieve higher reliability compared with the baseline RPL but also with the current state of the art.

III. DESIGN AND IMPLEMENTATION

The main idea of the proposed mechanism is *EB prediction*, meaning that if a joining node can predict which channel and timeslot the next EB will be broadcast on, then it can make sure to be listening on that channel and only during the correct timeslot. To be able to make this prediction the node must know the Absolute Slot Number (ASN), the channel offset of the link, the channel hopping sequence, and of course the schedule of the TSCH network. This means that *EB prediction* is only possible *after* the joining node has received the necessary information about the network, all of which can be included in an EB. Nonetheless, after receiving an EB this solution would improve the performance for the remainder of the joining-process significantly. Additionally, it would be applicable in re-joining scenarios, in which a node disassociates from the network and must re-join, by retaining network information for some time after disassociating. Thereby new nodes can reduce their initial join time, however, mobile nodes will further benefit from these properties as they are more likely to disassociate and re-associate from/to the network during operation.

Moreover, the solution would decrease the energy consumption of the initial scanning process as it will be shortened significantly resulting in reduced radio-on-time. Furthermore, after receiving the first EB it will be possible to predict the time and channel of the next EB and thus an amount of energy can be conserved by deactivating the radio during the time-period until the next EB is broadcast.

A. Topology Data

In order to aggregate the network topology data, the EBs were extended to include topology data as *MLME Vendor Specific Nested Information Elements* [14]. This type of Information Element (IE) was used since it allows for any *vendor specific* information to be serialized and added as an IE to the EB. The benefit of using the EBs to transmit the topology data is that EBs are already integrated in all implementations of TSCH, thus there is no implementation and design overhead. Furthermore, it is not required to allocate additional timeslots for aggregation of the data, since timeslots are already allocated for EBs. For this proposed mechanism it is necessary for all network nodes to keep a local topology data set. Just before broadcasting an EB, a node will always update its own entry in the topology data to keep the information updated. The information contains the *node ID*, its *parent ID*, if the node has *left the network*, the *channel offset* and the current *ASN*. Thus, after transmitting this information a receiving node will use the ASN to determine if the received data is up to date, allowing it to disregard or replace outdated information in its own local data set. A node is considered to have left the TSCH network if it has not communicated with its parent for 8000 ASNs. Therefore, if a node has moved out of the network, its parent will mark it as *left network* roughly by the time it would have disassociated from the network. It is important to track which nodes are still an active part of the network in order to ensure that joining nodes do not attempt to listen for EBs from inactive nodes. The purpose of this mechanism is to monitor the topology data of each node at any given time, as well as how new information in the topology data propagates throughout the network when a new node joins. Using this mechanism a joining node may join into the TSCH network faster but also discover more possible parents in a shorter time-span.

B. ASN Tracking

In order to accurately predict when an EB may be broadcast, it is necessary to know the current ASN of the network. Since relying on the network is inefficient, due to the communication overhead that will be generated, the joining node must be able to track the ASN locally to a certain precision determined by the guard times of the timeslots. The time interval between the time when a node starts listening and the time of the estimated arrival of a packet is called *guard time* and its purpose is to reduce the likelihood of missing a packet due to timing errors caused by clock drift. When the ASN of the network is yet unknown, the node will randomly select new channels for scanning at a fixed interval of 100 ms. Once an EB is received a timestamp is created which is used to monitor the ASN, utilizing a real time clock function. During timestamping

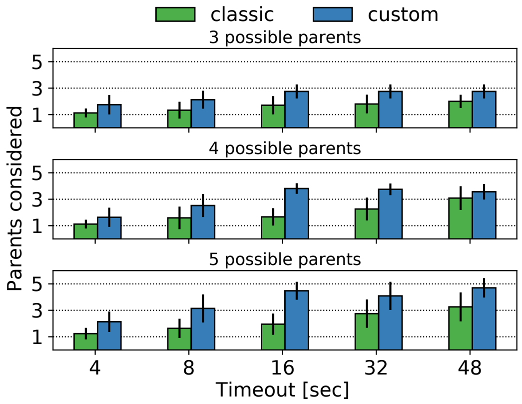

Fig. 1: The average number of parents considered before a joining node associates to an existing network of 3, 4 and 5 nodes given various configurations of the scanning timeout.

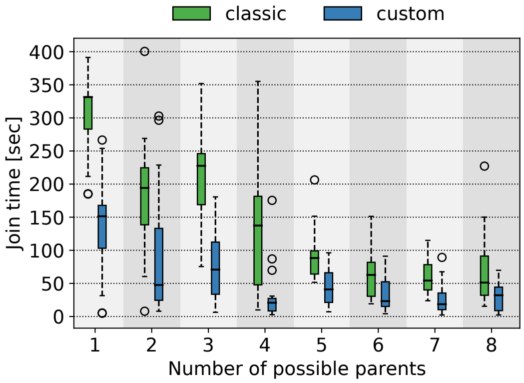

Fig. 2: The time required for a joining node to associate to an existing network given various amounts of reachable nodes.

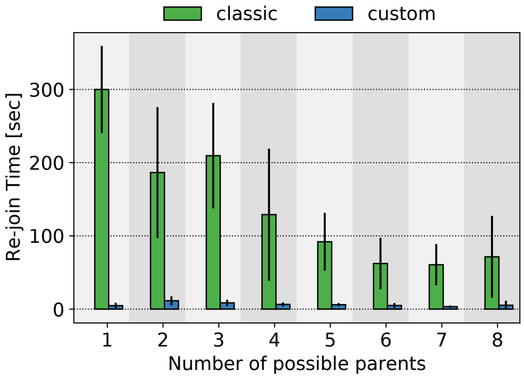

Fig. 3: The average re-join time for both the Custom and Classic versions in various network densities.

of the received EB packet, a time offset, caused by parsing it and its transmission delay, is estimated to ensure accuracy. The aforementioned mechanism enables nodes to track the ASN of a TSCH network locally, while still scanning for additional EBs. In order to reduce power consumption, the scanning mechanism is able to turn off the radio if the schedule and ASN of the network is known and no EB broadcast slot is scheduled in the current timeslot. Furthermore, tracking the ASN locally is continued upon disassociation in order to assist nodes in re-joining the network faster. This will allow recently disassociated nodes to scan more efficiently for EBs from the established network and thus be able to re-join the network faster.

IV. EVALUATION

This section presents and discusses the results obtained by comparing the proposed *custom* version of the TSCH association mechanism with standard, hereafter named as *classic*. Furthermore, it describes the methods and approaches used to test the implementation. In order to compare the two mechanisms we used Contiki-NG and the Cooja simulator [6] and we defined several scenarios to observe and quantify the scanning and joining processes of a new node introduced in an existing TSCH network, or a mobile node disassociating from a TSCH network in order to re-associate to it again later. During the evaluation the Minimal Scheduling Function [15] was used for every scenario and the duration of each simulation was 10 minutes. We used the Zolertia Z1 [16] mote as a platform in Cooja. Every test case menioned in this Section was simulated 15 times with different random seeds to avoid statistical bias, resulting in 1530 individual simulations in total.

One of the parameters that can be regulated during the TSCH joining process is the maximum amount of time a node should scan for potential parents after having discovered the first parent, in other words the *scan timeout*. A larger timeout is likely to result in more discovered parents and better parent selection based on the rank, but also it will result in larger energy and delay overhead. Note that IETF recommends to use 180 s as a scan timeout when using the minimal mode of operation. Also the maximum amount of nodes to search was left unlimited since one of the goals was to discover the maximum amount of available parents. The channel sequence was 4 channels and the rest of the configuration was the default Contiki-NG implementation. The scenario was set up such that a node would join an existing network of 3, 4, and 5 nodes. In Fig. 1 for all three graphs it is clear that the performance is increased when the timeout is increased and that more parents are considered when increasing the timeout up to 16 s. Afterwards the performance stabilizes since the maximum delay between EB broadcasts is 16 s. Comparing the classic with the custom mechanism we see that there is a performance increase in every case when the custom mechanism is used, with an average improvement of 40%.

Next we quantify the time it takes a single node to associate to an existing network for a set of topologies with different amount of nodes. The scenario is therefore set up such that the joining node is out of range of all other nodes, while the network forms. Once all other network nodes have associated to the network, the joining node is moved within range of all nodes in the network – for this reason the test is performed on networks consisting of various amount of nodes. Then, from this moment the time required for the joining node to associate to the network is measured. For the scan timeout we used 16 s for the custom version, which was obtained from the results in Fig. 1, and 180 s for the classic as it is recommended by IETF. After having received the first EB, a node may scan additionally for at most the amount of seconds defined by the scan timeout or until it has received EBs from a set maximum amount of distinct neighbors. The value for the amount of distinct neighbors is 2 for both TSCH versions and is also recommended by IETF. The join time is illustrated with a box plot in Fig. 2 and shows that the custom version requires approximately half the time of the classic version to associate to the network. While the absolute improvement decreases as more parents are available, the relative improvement remains fairly constant at approximately 50%.

To evaluate the case when a node disassociates and re-associates due to mobility, or any other reason, we defined the following scenario. A node starts within a TSCH network and once the network is completely formed, the node is moved out of range for 2.5 minutes after which it will have disassociated. According to TSCH a node will disassociate if it is unable to to re-synchronize its time source for 8000 ASNs (2 minutes on the Zolertia Z1 platform). The node is then moved back within range of the rest of the nodes of the network and the node starts the association process again, this time utilizing the network information which the re-joining node already possesses from prior network association. The configuration parameters are also the same in this scenario, the size of the channel hopping sequence is 16, the value for the maximum amount of distinct neighbors to scan for is 2 and the scan timeout is 16 s. Fig. 3 presents the results and it is evident that the custom version performs significantly better than the classic one since it takes advantage of the knowledge about when and where the the next EB will be transmitted and does not follow a random approach like the classic version. More specifically the time to re-join for the custom version is 92% to 98.5% less than the classic one, depending on the amount of available parents in the network.

V. CONCLUSION

The focus of this paper was to improve the association mechanism of TSCH as it may result in increased delay and wasted energy. Especially for application scenarios that include mobile nodes the joining procedure can be a major issue towards the network performance. To this end, additional network data is continuously collected and propagated throughout the existing network, encapsulated into the EBs. By using the existing slotframe scheduling infrastructure we alleviate the requirement of extensive changes but introduce a slight overhead in longer transmissions caused by additional network data. The network data is used to predict the timeslot and channel of the next EB. Another feature added is the ability to retain network information upon disassociation from a network in order to further improve a re-joining process. The evaluation show significant results for both the initial joining process and the re-joining process. The initial joining process reduces the join time by approximately 50% or more compared to a classic TSCH implementation, depending on the network topology and configuration. The re-joining process shows even better results, with a re-join time reduction of 90% or more.

VI. ACKNOWLEDGEMENTS

This research was partially funded by Innovation Fund Denmark, grant 9092-00007B AgroRobottiFleet.

REFERENCES

[1] S. Duquennoy, A. Elsts, B. A. Nahas, and G. C. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 11–18, 2017.

[2] A. Hithnawi, H. Shafagh, and S. Duquennoy, "Understanding the Impact of Cross Technology Interference on IEEE 802.15.4," in *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiNTECH '14, (New York, NY, USA), p. 49–56, Association for Computing Machinery, 2014.

[3] S. Raza, T. V. D. Lee, G. Exarchakos, and M. Güneş, "A Reliability Analysis of TSCH Protocol in a Mobile Scenario," in *16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2019.

[4] C. Orfanidis, A. Elsts, P. Pop, and X. Fafoutis, "TSCH Evaluation under Heterogeneous Mobile Scenarios," *IoT*, vol. 2, no. 4, pp. 656–668, 2021.

[5] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.

[6] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pp. 641–648, 2006.

[7] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration." RFC 8180, 2017.

[8] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the IoT MAC infrastructure: IEEE 802.15.4e TSCH and LLDN platform," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015.

[9] Y. Al-Nidawi and A. H. Kemp, "Mobility Aware Framework for Timeslotted Channel Hopping IEEE 802.15.4e Sensor Networks," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7112–7125, 2015.

[10] C. Orfanidis, P. Pop, and X. Fafoutis, "Active Connectivity Fundamentals for TSCH Networks of Mobile Robots," in *18th Int. Conf. Distributed Computing in Sensor Systems (DCOSS) Workshops*, 2022.

[11] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali, "Impacts of Mobility Models on RPL-Based Mobile IoT Infrastructures: An Evaluative Comparison and Survey," *IEEE Access*, vol. 8, pp. 167779–167829, 2020.

[12] M. Barcelo, A. Correa, X. Vilajosana, J. L. Vicario, and A. Morell, "Novel Routing Approach for the TSCH Mode of IEEE 802.15.14e in Wireless Sensor Networks with Mobile Nodes," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pp. 1–5, 2014.

[13] I. Rabet, S. P. Selvaraju, M. H. Adeli, H. Fotouhi, A. Balador, M. Vahabi, M. Alves, and M. Björkman, "Pushing IoT Mobility Management to the Edge: Granting RPL Accurate Localization and Routing," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pp. 338–343, 2021.

[14] "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, 2012.

[15] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)." RFC 9033, May 2021.

[16] Zolertia, "Z1 Datasheet." http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf, 2010. [Online; accessed 29-March-2022].