This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199

Time Hopping: an Efficient Technique for Reliable Coexistence of TSCH-based IoT Networks

Majid Nabi, *Member, IEEE*, Mina Habibollahi, and Hossein Saidi

***Abstract*—Escalation in the use of Internet-of-Things (IoT) devices gives rise to the number of networks operating in the license-free 2.4 GHz frequency band. This prepares the ground for networks to experience interference from coexisting networks and thus performance degradation. Time Slotted Channel Hopping (TSCH), as an operational medium access mode of the IEEE 802.15.4 technology, was introduced to ensure the reliability of IoT networks when they undergo coexistence. It uses frequency hopping as a protective strategy against long-term packet losses due to interference. However, when several independent TSCH networks coexist, they are prone to interfere with one another. In extreme scenarios, coexisting TSCH networks may block links of one another for an extended duration of time, leading to application failure. In this paper, we propose a novel technique called time hopping to secure the reliability of coexisting TSCH networks. The developed technique synchronously and periodically alters the timing of nodes within a TSCH network to avoid coexisting TSCH networks from getting stuck in extreme coexistence scenarios and long-term continuous collisions. We evaluate the effectiveness of the proposed technique through extensive simulations. The results clearly show that the proposed time hopping technique substantially improves the worst-case inter-network collision ratio, with as much as 50% improvement in some tested scenarios. The implementation of the technique is very simple, with almost no communication or computation overhead for the constrained wireless nodes; it is done and tested on real nodes for proof of concept.**

***Index Terms*—IEEE 802.15.4, TSCH, Coexistence, Time hopping, Channel hopping, reliability.**

I. INTRODUCTION

THANKS to the Internet-of-Things (IoT), it is possible to connect every device to the Internet and other devices. To achieve this goal, IoT deploys Wireless Sensor Networks (WSNs) composed of myriad autonomous devices for monitoring various parameters. IEEE 802.15.4 is a standard technology developed for Low-Rate Wireless Personal Area Networks (LR-WPAN); it specifies the physical and Medium Access Control (MAC) layers. Like many other short-range communication technologies, IEEE 802.15.4 operates in the 2.4 GHz license-free ISM band for communication. While using the ISM bands greatly eases deployments of personal WSNs, their availability to several wireless technologies (e.g., Bluetooth classic and low energy, IEEE 802.11 Wi-Fi, etc.) makes the band quite busy, causing Cross-Technology Interference (CTI) to be a major issue for communication reliability of such WSNs. To address this, new releases of the IEEE 802.15.4 standard introduce specifications of a MAC operational mode called Time Slotted Channel Hopping (TSCH) fortified by effective mechanisms to combat CTI.

TSCH enables nodes to share the medium with the Time Division Multiple Access (TDMA) mechanism. Internal collisions of packet transmissions within a TSCH network are avoided by such a TDMA-based channel access, resulting in more predictability and determinism for industrial applications. Also, the TSCH protocol employs frequency channel hopping to mitigate multi-path fading effect and external interference. Combining TDMA with channel hopping, TSCH enhances network capacity and reliability along with a reduction in energy consumption due to fewer collisions. These attributes have given rise to the widespread adoption of this technology in many applications. Consequently, the coexistence of numerous independent (thus asynchronous) TSCH networks is highly probable in some locations. As an example, consider hospitals or elderly care centers, wherein TSCH is employed as the communication standard for Wireless Body Area Network (WBAN) [1] [2]. In such places, the likelihood of individuals wearing TSCH-based devices being in close proximity is high. As another example, TSCH is very likely to be used for mobile devices such as intra-vehicle communication [3] [4] and moving robots [5]. In consequence, when several cars are close enough in some cases, like in heavy traffic, their TSCH networks may interfere with one another. These examples imply that the coexistence of TSCH networks is bound to happen, causing the networks to be subject to consecutive collisions and long-term disconnections.

The behavior of co-located TSCH networks in all possible scenarios is extensively investigated in [6]. Despite its rather effective effort to mitigate interference coming from coexisting TSCH networks, channel hopping alone is not able to prevent extreme coexistence scenarios in which several TSCH networks interfere with one another in consecutive timeslots for an extended duration of time. As the number of co-located asynchronous TSCH networks increases, consecutive collisions generate scenarios in which networks overlap both in time and frequency for a long time. Such cross-TSCH interferences disrupt data transmission to the extent that it may lead to the failure of IoT applications.

In this paper, we propose *time hopping*, a new technique that effectively prevents TSCH networks from getting trapped in the extreme coexistence scenarios when several TSCH

Majid Nabi is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, and also with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran (e-mail: m.nabi@tue.nl).

Mina Habibollahi and Hossein Saidi are with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran (emails: habibollahi@ec.iut.ac.ir, h.saidi@iut.ac.ir).

networks are expected to co-locate. The contributions of this manuscript are as follows.

1) A time hopping mechanism is proposed by which all nodes in a TSCH network shift the beginning of their next timeslots by inserting particular time delays synchronously every certain number of timeslots. This is done using a time hopping list which is known by all the nodes in a TSCH network. Inserting such periodic time shift into the timeslot structure, TSCH networks escape from conditions in which they overlap in time and frequency with other TSCH networks in their vicinity. This mechanism has a very simple and light implementation with no need for any extra control packet exchange or any computation overhead.
2) The proposed time hopping mechanism is evaluated through extensive computer simulations using the TSCH coexistence simulator developed in [6] to investigate its effectiveness in avoiding extreme coexistence scenarios. The results show that time hopping is able to substantially improve the worst-case coexistence of multiple TSCH networks, reducing the worst-case packet collision ratio by even 50% in some scenarios; the achieved enhancements vary depending on the number of coexisting TSCH networks.
3) The impact of the mechanism parameters is evaluated to explore the trade-offs in term of reliability improvement and overheard, and to provide guidelines for configuring such parameters.
4) As proof of concept, the mechanism is implemented and integrated in the Contiki [7] operating system and tested targeting platforms such as the NXP JN5168 transceiver [8] using the Cooja [9] simulator.

The rest of the paper is organized as follows. In Section II, the necessary background on the TSCH technology is presented. In Section III, the related work on the coexistence of short-range IoT communication technologies is reviewed. Section IV describes the proposed time hopping technique in detail. In Section V, results on the effectiveness of the time hopping technique are presented, and the configuration of the mechanism is discussed. Section VI discusses the real implementation of the technique and its validation through Cooja simulations. Section VII concludes.

II. TSCH BACKGROUND

IEEE enhanced the MAC layer of IEEE 802.15.4 in order to reduce the standard's vulnerability to multi-path fading and external interference. Among others, TSCH is a MAC mode that uses TDMA-based channel access together with channel hopping to enhance the reliability and efficiency of communications. It divides time into equal length timeslots, a number of them creating a slotframe that repeats over time. The length of a timeslot (T_{ts}) is long enough for sending a maximum size packet (133 bytes) and receiving its related acknowledgment (Ack); the default length is $T_{ts} = 10ms$. Each timeslot is allocated for the transmission of one or more nodes via a scheduling mechanism, which is not a part of the TSCH standard; it is left for upper layers in the protocol stack.

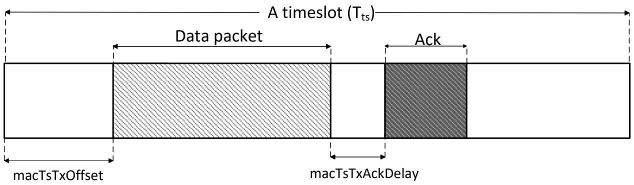


Fig. 1: General structure of a typical TSCH timeslot

Fig. 1 shows the structure of a TSCH timeslot. It starts with a gap specified by parameter macTsTxOffset. The transmitter node starts its packet transmission after macTsTxOffset after the beginning of the timeslot with reference to its own timing. Receivers start to listen to the channel well before macTsTxOffset to compensate for small clock drifts between their clock and that of the transmitter. If an Ack is requested by the transmitter, the receiver waits for macTsTxAckDelay after receiving the data packet and then sends an Ack.

TSCH nodes jump to different frequency channels (channel hopping) in each timeslot. By such a mechanism, nodes send their packets in various frequency channels and avoid getting stuck in using a noisy frequency channel. Using diverse frequency channels reduces the impact of multi-path fading and external interference. TSCH exploits the 16 frequency channels in 2.4 GHz band specified by the IEEE 802.15.4 standard. Nodes use Eqn. 1 to find out which frequency channel they should use for their communication in each timeslot.

$$CH = HSL\ [\ (ASN + Ch\_Off)\ \%\ |HSL|\ ] \qquad (1)$$

HSL is an ordered list of channels to be used and can comprise up to the 16 available channels; $|HSL|$ shows the number of channels in this list. Absolute Slot Number (ASN) is the total number of timeslots since the start of the network, which is a synchronized parameter all over the TSCH network. Ch_Off stands for Channel Offset and gives the possibility of parallel transmissions in different channels, increasing the capacity of a TSCH network for data delivery.

Synchronization of the nodes in a TSCH network is crucial to have the TDMA and channel hopping mechanism properly functioning. To keep the nodes synchronized, the standard specifies two methods for time synchronization, namely frame-based and Ack-based synchronization. During every packet exchange, the transmitter and receiver synchronize to each other to keep the timeslots aligned; one of them plays the role of time-source. In the former method, the transmitter of a packet is the time-source, and the receiver follows the timing of the transmitter, while in the Ack-based synchronization the transmitter aligns its timing with that of the receiver. Besides time synchronization that is needed for TDMA mechanism, all the nodes in the network need to have a synchronized ASN value to correctly implement channel hopping.

III. RELATED WORK

The popularity of the license-free 2.4 GHz ISM band and the operation of major IoT wireless technologies in this band result in the coexistence of different technologies, and therefore spectrum congestion. Several pieces of research have been carried out to either study and analyze the impact of such coexistence or to address it with various mechanisms aiming to ensure the reliability of packet delivery in coexisting networks.

Due to the severity of the impact of Wi-Fi signals on IoT networks, it has been the topic of many investigations. [10] develops a coexistence model of IEEE 802.15.4 and IEEE 802.11b/g by taking into account timing and power aspects. It introduces the concept of coexistence range and investigates the interaction behavior of coexisting standards in three different ranges. As prominent widespread IoT communication technologies, [11] conducts a study on coexistence of IEEE 802.15.4, Bluetooth Low Energy (BLE), and IEEE 802.11 Wi-Fi. The results of this study show that Wi-Fi communications have a stronger impact on BLE compared to IEEE 802.15.4, thanks to the techniques used in the physical layer of IEEE 802.15.4, such as spread spectrum. However, the channel hopping mechanism used in BLE effectively alleviates the interference impact.

In [12], authors propose a solution for the coexistence of 15.4 and 802.11b/g, in which 15.4 nodes smartly switch to different frequency channels, based on the observed Packet Delivery Ratio (PDR) and Received Signal Strength Indicator (RSSI). [13] tries to improve the performance of IEEE 802.15.4 in the presence of Wi-Fi interference by altering some parameters of the PHY/MAC layers of the Wi-Fi network in such a way that opportunities are provided for the 15.4 networks to send its packets. In an attempt to improve coexistence in favor of 15.4, [14] and [15] both cancel Wi-Fi signals when Wi-Fi and 15.4 have coexistence issues.

[16] propose a cross-technology synchronization mechanism for the coexistence of TSCH and IEEE802.11. This mechanism minimizes the TSCH packet loss rate by enabling Wi-Fi to refrain from sending packets during transmission periods of TSCH. [17] use an adaptive channel selection algorithm to protect TSCH against interference. The algorithm was tested under Wi-Fi interference, resulting in the augmentation of TSCH throughput. [18] presents a cooperative coexistence mechanism for BLE and TSCH networks by developing a scheduling matrix made of coexisting networks' resources (time and frequency). A coordinator arranges this matrix and reschedules coexisting networks' transmissions in case they are predicted to have overlap.

The review so far gives some insight into the coexistence of different technologies. However, multiple independent networks of the same type (TSCH networks as the focus of this work) may also coexist and make interference for one another. [19] is one of the first research regarding the coexistence of multiple TSCH networks. It demonstrates that collisions go up with a rise in the number of coexisting TSCH networks. Also, it shows that the number of collisions is substantially higher when multiple asynchronous TSCH networks coexist compared to the case they are synchronized. This paper supposes that packets of coexisting networks collide even if their timeslots partially overlap. However, [6] shows that there are still quite some chances of successful transmissions when timeslots of two TSCH networks overlap in time and frequency. [20] is the other work that investigates the coexistence of TSCH networks. It proves that the clock drift between coexisting TSCH networks makes the inter-network interference periodic and concludes that networks can increase the chance of successful transmissions by exploiting more frequency channels. [6] studies the coexistence of multiple asynchronous TSCH networks. This paper develops a simulator that estimates the Collision-Free Ratio (CFR) of multiple TSCH networks through Monte Carlo simulations. It analyzes various cases of coexistence, including the impact of clock drift on the performance of coexisting TSCH networks via the developed simulator.

As it is shown in [6], when a number of independent TSCH networks get close to one another, a vast distribution of impact may happen depending on the relative timing and channel hopping sequences of the involved TSCH networks. There are cases in which there is no overlap in time nor in frequency leading to perfect coexistence without collisions. On the other extreme, there can be cases wherein various TSCH networks overlap in time and frequency for an extended duration of time, resulting in long-term link blockage and substantial degradation of packet delivery reliability. In this paper, we aim to avoid such extreme cases to assure continuous data delivery in case several TSCH networks coexist. The proposed time hopping technique efficiently and effectively prevents such extreme scenarios.

IV. TIME HOPPING TECHNIQUE

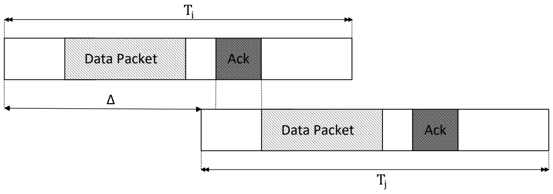
In this section, we first discuss the general idea of time hopping and the impact it can have on the coexistence of TSCH networks. Then, we describe the developed mechanism in detail.

A. Time Hopping Proposal

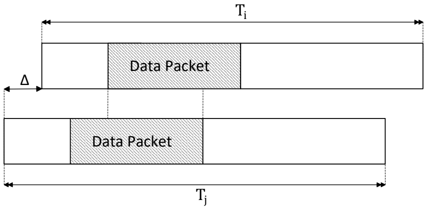
Frequency hopping is employed in several IoT communication technologies in order to mitigate the detrimental effect of the coexistence of wireless technologies operating in the same frequency band, including multiple co-located TSCH networks. Although it greatly improves the reliability of such networks, there is still chances of getting trapped in situations in which several networks consecutively collide with each other, meaning that they happen to use the same channel sequence and overlapped timing. An extensive study in [6] shows that such scenarios are not that rare, and the chance increases when more TSCH networks coexist. To state the problem clearly, we first take a closer look at the coexistence of TSCH networks from the time standpoint.

In the coexistence of TSCH networks, both frequency and time of transmissions are contributing factors in a successful transmission. Packets of various networks collide when they are transmitted with time and frequency overlaps. If the coexisting networks use different frequency channels while transmitting their packets simultaneously, they can transmit without collision. [6] investigates the chance of frequency channel overlap for several coexisting TSCH networks. It shows that as the number of coexisting networks increases, the probability of having more overlapping channels goes up. For example, when seven networks are in the range of one

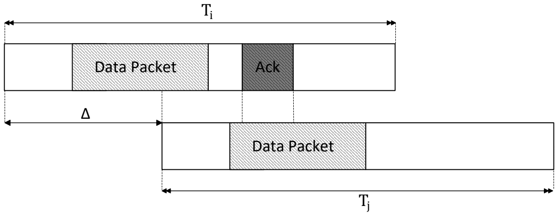
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199



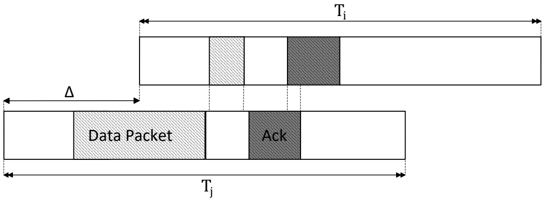
(a) Two networks transmit without collision despite timeslot overlap.



(b) Data packets are collided, thus no Ack packet is sent.



(c) One network's Ack transmission collides with another network's data transmission



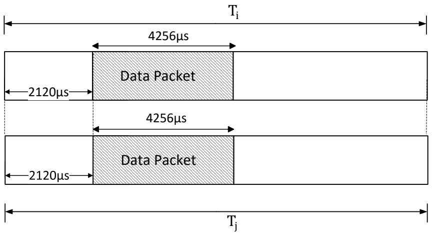
(d) Ack packets of the two networks collide (rare case)

Fig. 2: Illustration of some time overlap cases when two TSCH networks coexist depending on the relative time deviation of their timeslots.

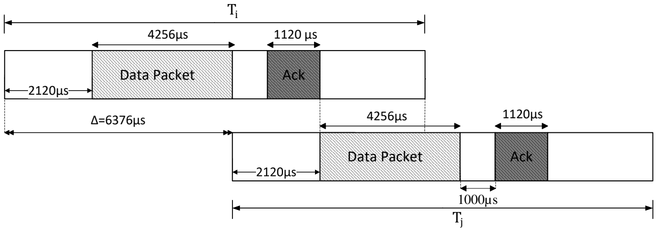
another, the chance of having no channel overlap is almost zero. In the case of channel overlap, the sole factor that can save coexisting networks from collisions is time; the relative position of timeslots plays a decisive role. Fig. 2 illustrates various coexisting cases of two TSCH networks that operate in the same channel. T_i and T_j stand for timeslot length of network i and j, respectively. Time deviation (Δ) shows the time difference between the start of two networks' timeslot, ranging from T_i to T_j $[-T_i < \Delta < T_j]$, assuming that network i is the time reference.

Based on the value of time deviation, the networks may collide. Fig. 2a illustrates a case in which timeslots of the two networks do not have time overlap resulting in collision-free packet transmissions. In Fig. 2b, two data packets overlap in time and thus collide. Assuming that this collision leads to packet losses in both networks, no Ack packet is transmitted in this case. Collisions occur not only because data packets collide, but a collision with Ack packets can also lead to packet loss or retransmissions. In Fig. 2c, the Ack packet of a network overlaps in time with packet transmission in the other network.

There are other factors that influence the chance of overlaps. Among others, the length of timeslots and packets affect the change of collision-free transmission of co-channel networks [6]. This probability decreases as the networks' packet length increases. According to [6], at best, the probability of collision-free transmission for two coexisting TSCH networks is about 80% when the two networks transmit packets of only 20 bytes. This probability goes down to 40%, at worst, when packets' length grows to 120 bytes. Note that such probability is actually the chance of time overlap when two independent TSCH networks get close to each other.



(a) In both scenario timeslot's packets transmission at the start of coexistence are simultaneous



(b) In second scenario after elapsing 10627 timeslots, networks' packet transmission are not simultaneous.

Fig. 3: Impact of clock drift on networks' coexistence

The big concern here is that when time overlap happens, it will remain there while the timeslot structure of the involved networks remains the same. This means that when a number of TSCH networks are in the range of one another, there are chances that they get stuck in a situation wherein they cannot transfer their data for an extended duration of time. This is the main concern that is being addressed in this work. The good news is that the relative time deviation of timeslots of the coexisting networks (Δ) may change over time. Although the TSCH protocol has a synchronization mechanism to combat clock drifts within a TSCH network, the clock drifts can cause displacement of the timeslots over time. This lead to variations of Δ over time. These changes may alleviate the extreme situation. This effect is also confirmed in the result presented in [6]. In the following, we analyze to what extent the clock drifts can help in this regard.

A typical clock drift for a crystal clock is ±30 ppm. Consequently, independent TSCH networks can take an inter-network clock drift value in range of [-60ppm , +60ppm] [6]. Consider a scenario in which two co-located TSCH networks jump to the same sequence of frequency channels. These two ack-enabled networks both transmit full-size (133 bytes) data packets and 35 bytes (extended) Ack packets within $10ms$ long timeslots. Assume that the time deviation between these

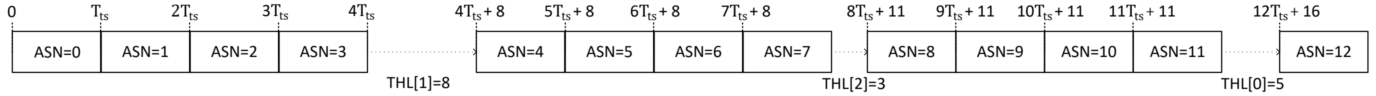


Fig. 4: As illustrative example of execution of the time hopping mechanism ($THL = \{8ms, 3ms, 5ms\}$ and $N_{TH} = 4$).

networks is initially zero (Δ=0). Thus, their transmissions have full overlap in time (Fig. 3a). If Δ remains constant, the networks stay in such a situation. Assuming the maximum clock drift between the two networks (±60ppm), Δ will increase by $0.6\mu s$ in each timeslot. These shifts gradually add up and take the networks into a non-overlapping state (Fig. 3b). The required time deviation for a collision-free transmission is at least $6376\mu s$ (macTsTxOffset + Data packet duration). Thus, at least 10627 timeslots should elapse in order to make a time deviation that exceeds $6376\mu s$ (6376/0.6). It means that the natural clock drifts need quite some time to be able to rescue networks from consecutive collisions.

The main idea of time hopping for enhancing the coexistence of independent TSCH networks is inspired by the impact of clock drifts. This technique can make the required time deviations to avoid the extreme cases of consecutive time overlaps. It is very important to realize that time hopping influences network performance in the same way the channel hopping itself works. Time hopping aims at avoiding stuck at overlap situations, while channel hopping aims at avoiding stuck at noisy channels. Both techniques improve the extremely harsh scenarios. However, sometimes such techniques worsen the good extremes. For time hopping, it happens when transmissions of co-channel networks already do not have time overlap, and time hopping changes this situation to other time overlap cases for a period of time. A similar case for channel hopping is the case in which the channel being used is perfect, but channel hopping causes switching to other channels that may not be as good as the original channel. These are all to avoid extremely bad scenarios.

B. Developed Time Hopping Mechanism

The time hopping mechanism acts on the basis of changing timeslots boundaries in a similar way as clock drift, except that time hopping does it in a highly agile way. Different approaches may be taken to implement adding such intentional jumps in time. In this paper, we developed a synchronous time hopping mechanism in which all nodes belonging to a TSCH network regularly shift the start of their timeslots simultaneously based on a predetermined Time Hopping List (THL). This list consists of a number of time values in range of $(0 \quad T_{ts})$. THL is consistent and known by all the nodes within a TSCH network (like HSL). In fact, each time value shows the delay or time deviation that nodes must make. The nodes perform such time hopping after every N_{TH} timeslots, referred to as the *time hopping interval.*

The proposed time hopping mechanism includes two steps. In the first step, nodes find out whether they have to perform time hopping within their current timeslot. Then, if it is the right timeslot for performing time deviation (i.e., N_{TH} timeslots have elapsed after the last time hopping), nodes proceed with the second step in which they simultaneously choose the same time deviation value from THL, and delay their timeslot in accordance with the chosen value. Eqn. 2 gives the delay at each time hopping event. Since this equation works based on the globally synchronized ASN, all nodes will change their timing with the same amount, avoiding deviations between the nodes in the network. This is actually similar to the way the nodes select the same channel in the channel hopping mechanism.

$$D = \begin{cases} THL[\ \frac{ASN}{N_{TH}}\ \%\ |THL|\ ] & \text{ASN}\ \%\ N_{TH} = 0 \\ \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

According to Eqn. 2, in timeslots in which ASN is not divisible to the time hopping interval (N_{TH}), no time hopping happens. When this is not the case (every N_{TH} timeslots), nodes divide ASN by N_{TH} and then divide the quotient by the size of the time hoping list ($|THL|$). The remainder of the last division gives the index of THL for accessing the right time hopping value. The nodes then delay the start of their timeslot by D.

As an example, consider a network that utilizes THL=$\{5ms, 8ms, 3ms\}$ as its time hopping list, and N_{TH} = 4, meaning that time hopping is set to occur every four timeslots when ASN is a multiple of 4. Fig. 4 illustrates time hopping for this setting. For the timeslot with ASN=4, for instance, the remainder of ASN over N_{TH} is zero; thus the nodes perform time hopping. According to Eqn. 2, THL[1] is then selected as the time hipping step, which corresponds to $8ms$ of delay. Accordingly, when ASN is 8 and 12, nodes delay their clocks as much as $3ms$ and $5ms$, respectively. This pattern is then repeated.

The choice of time deviation values in THL needs to be done in such a way that the probability with which coexisting independent TSCH networks have the same THL is quite low. Towards this, a practice is to fill THL with different values from the whole range of $(0 \quad T_{ts})$ in a shuffled way. A guideline to this end is to divide the range $(0 \quad T_{ts})$ into $|THL|$ equal size parts and then randomly select a value from each part. Then the selected values are shuffled and put into THL. Doing so, various independent networks will have different time deviations with a high probability. It is worth mentioning that even if the time hopping values of the two coexisting networks are close during a time hopping, it still does not mean that they overlap in time since different networks are not synchronous in time, and they do not perform time hopping necessarily at the same moment. These randomized behaviours together with frequency channel hopping of the TSCH protocol dramatically reduces the chance of overlap in time and frequency, leading to much more reliable packet delivery in the involved networks.

C. Cost of Time Hopping

Implementing the proposed time hopping mechanism does not impose extra computation for the embedded processor of wireless nodes. Moreover, the mechanism does not require any extra packet exchange for its operation; it uses the already existing notion of the global time in the TSCH network (i.e., ASN) and needs no more alignment between the nodes. However, since the technique inserts a time delay every N_{TH} timeslots, it has a minor impact on the nominal throughput of the network. This is analyzed in this section.

We refer to nominal throughput as the number of data packets that can be transmitted in the TSCH network in a given time (say a second). Assuming that all timeslots in a TSCH network is used for data packet exchange (ignoring control packets), one packet can be transmitted in each timeslot in a neighborhood of the network. Therefore, the nominal throughput of the TSCH network without the time hopping mechanism is $\frac{1}{T_{ts}}$ packets per second (pps). When time hopping is enabled, a delay is added to the timeline every N_{TH} timeslot, the value of which is taken from the time hopping list. It means that every second, the network experiences $\frac{\alpha}{N_{TH} \times T_{ts}}$ total inserted delay, where α is the average added time delay taken from Eqn. 3.

$$\alpha = \frac{1}{|THL|} \sum_{i=0}^{|THL|-1} THL[i] \tag{3}$$

Therefore, the nominal throughput of the time hopping TSCH network is given by Eqn. 4.

$$Throughput = \frac{1}{T_{ts}} \left(1 - \frac{\alpha}{N_{TH} \times T_{ts}}\right) \tag{4}$$

It is clear from Eqn. 4 that having less frequent time hopping results in a lower impact on the throughput; the impact goes to zero when $N_{TH} \to \infty$ (no time hopping). on the other extreme, we may do time hopping every two timeslots ($N_{TH} = 2$). In this case the throughput impact of time hopping is $\frac{\alpha}{2T_{ts}}$. Assuming that a balanced set of time delay values are in THL, and considering that all values are in the range of (0 , T_{ts}), we have $\alpha = \frac{T_{ts}}{2}$. Thus, the throughput overhead will be at most 25%. It is very important to realize that this is a very extreme estimation of the overhead since the time hopping does not need to be that frequent. On the other hand, note that in the proposed time hopping mechanism, only positive delays are inserted with the assumption that the timeslot length is carefully selected and no time can be reduced from it. Also, improving the packet delivery performance will improve the actual throughput of the network achieved by the time hopping technique. In the following sections, we experimentally investigate the trade-off made by the time hopping frequency in terms of nominal throughput overhead as well as the impact on the coexistence of TSCH networks.

V. PERFORMANCE EVALUATION

In this section, we aim to evaluate the effectiveness of the time hopping mechanism when several TSCH networks coexist. First, the considered evaluation metrics are presented following by describing the simulation setup. Then the achieved results are analyzed and discussed.

A. Evaluation Metrics

1) Collision-Free Ratio (CFR): Reliability of data delivery is the distinguishing characteristic of TSCH networks. From the perspective of coexistence analysis, inter-networks collision-free packet transmission is the metric of interest for reliability. It is defined as the ratio of all data packets transmitted in a network without time and frequency overlap to total packets transmitted in the considered time frame.

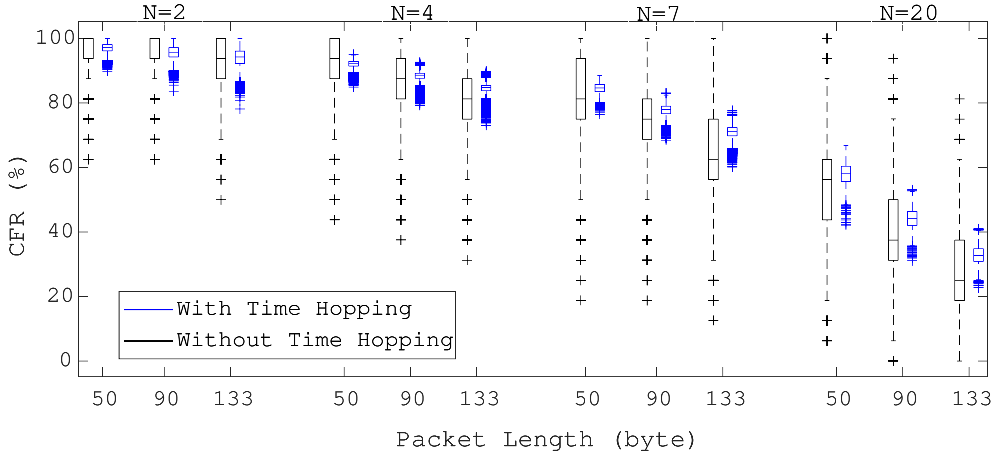
2) Burst Collisions: The collision-free ratio gives an average of packet overlap states over a time frame and does not represent a distribution of collisions. For evaluating the effects of the time hopping mechanism on the distribution of collisions, another metric is the number of burst collisions which is defined as the number of times that more than one subsequent packet collisions are observed during an experiment. This metric is useful and important for many industrial applications.

B. Simulation Setups

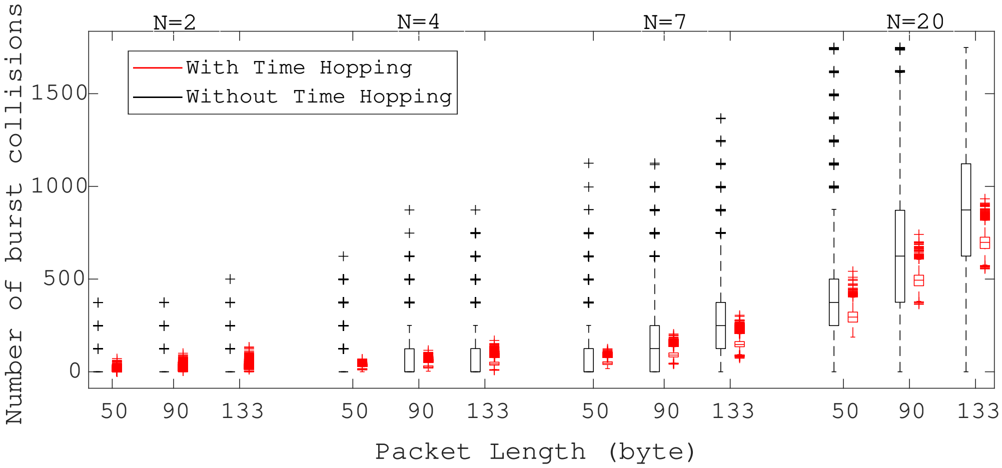
When a number of TSCH networks get close to one another, many coexistence scenarios may happen depending on the relative time difference of their timeslots. To be able to have a reliable investigation including many scenarios, fast Monte Carlo simulations are necessary. For that, we use the Multi-TSCH coexistence simulator that we developed earlier in [6]. We enhanced this simulator by implementing the time hopping mechanism. This simulator gets parameters like data packet length (L_{pkt}), ack packet length (L_{ack}), timeslot length (T_{ts}), experiment duration (T_{sim}), number of Monte Carlo runs (N_r), number of co-located networks (N), time hopping interval (N_{TH}), and the time hopping list (THL). The simulator keeps these inputs constant for all runs of the Monte Carlo simulation. For all simulations, three time-hopping values are randomly selected from the range of $(0 \;\; T_{ts})$ to form different THLs for the coexisting TSCH networks ($|THL| = 3$ for all networks). For each individual run, the simulator sets a randomly formed HSL (a shuffled set out of all 16 available channels in the 2.4GHz band) for each involved TSCH network and sets random relative time deviation (Δ) between the timeslot schedule of the coexisting TSCH networks. Running the simulation with the selected HSLs and Δ values, it calculates the collision-free ratio and the number of burst collisions. This simulation is repeated $N_r = 20000$ times. Table I gives settings of all simulated cases. The number of TSCH networks that are in the range of each other is an important parameter. We tested four cases in which 2, 4, 7, and 20 TSCH networks are coexisting. For each case, three sizes of the data packets ($L_{pkt} = 50, 90, 133$) are tested, resulting in 12 simulation setups in total. Each simulation setup is conducted twice: one with time hopping mechanism enabled and the other without time hopping mechanism to investigate the effectiveness of the proposed mechanism.

In the performed simulations, it is supposed that all nodes in all the coexisting TSCH networks are in the communication range of one another. An important fact about the performed simulations is that they are done from the perspective of the MAC layer to find out the number of packet transmissions

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199



(a) CFR for different number of coexisting TSCH networks and packet lengths



(b) Number of burst collisions for different number of coexisting TSCH networks and packet lengths

Fig. 5: Simulation results showing the impact of time hopping on the packet delivery performance of coexisting TSCH networks.

in different TSCH networks overlapping in both time and frequency (collisions). Whether such collisions lead to packet drop is a physical layer aspect depending on the relative Signal-to-Noise Ratio (SNR), receivers' sensitivity, and other physical layers effects. Therefore, the real distance between the nodes is not a matter here; being in the communication range is the assumption made. Moreover, it is assumed that all timeslots are used for packet transmissions in the involved TSCH networks. It may be because of frequent packet transmissions of a few nodes in a TSCH network or because of being a large network containing many wireless nodes. The important fact here is to evaluate the chance of collisions if there are packet transmissions in all the timeslots in all the coexisting networks.

C. Results on Effectiveness of Time Hopping

Fig. 5a presents the box-plot of the CFR values out of the performed Monte Carlo simulations for all 12 simulation setups. For each setup, the results for both cases wherein time hopping is enabled or disabled are shown next to each other for easier comparison. Note that each box-plot presents the distribution of the results out of $N_r = 20000$ simulation runs.

Parameter	Values	Description
T_{ts}	10 ms	timeslot length
T_{sim}	20 sec	time of one simulation run
N_r	20000	number of runs for each setup
N_{TH}	4	time hopping interval
L_{pkt}	50,90,133 Bytes	length of data packets in PHY layer
L_{Ack}	11 Bytes	length of Ack packets in PHY layer
N	2, 4, 7, 20	number of coexisting TSCH networks

TABLE I: Parameter values for Monte Carlo simulations

The first and most important observation is that when time hopping is enabled, the worst-case coexistence scenarios have been greatly improved. As an instance, when seven TSCH networks coexist for which the packet length is set to its maximum value (133 bytes), CFR is down to even 10% in some simulation runs when time hopping is not enabled. For the same scenario with time hopping enabled, the lowest experienced CFR is above 60%, showing a considerable improvement (around 50% increase). The same achievement holds for all other scenarios in Fig. 5a, confirming the hypothesis based on which we proposed the time hopping technique. The distribution of the results is clearly different with and without

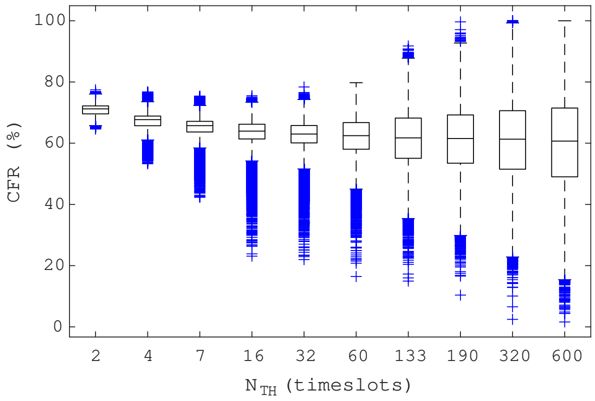
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199

time hopping; it is very much dense around the mean with a quite lower standard deviation. This means more predictability and determinism is provided for coexisting TSCH networks by the time hopping technique.

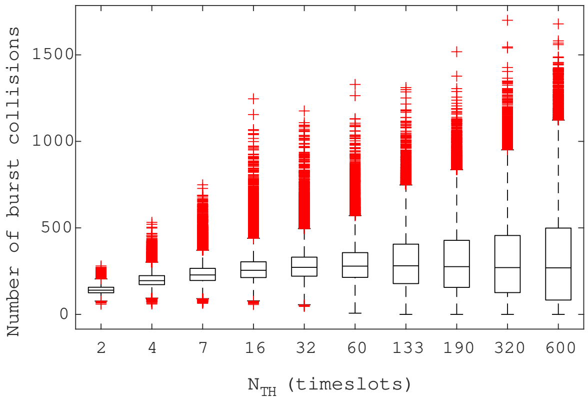
Another, yet expected observation, is that activating time hopping has degraded the best CFR cases. The reason is that if HSLs and the time deviations between the networks in some simulation runs are such that no frequency or time overlap occurs, then no collision is experienced when time hopping is disabled. However, time hopping prevents such scenarios as well by changing the time structure of different TSCH networks. This is the cost we actually pay to avoid worst-case coexistence scenarios in which almost no communication can succeed for an extended duration of time, which can lead to application failure. This is exactly the same phenomenon as in using channel hopping to avoid getting stuck at noisy channels; it may degrade the performance when the single channel is already the best clean channel. The important matter is that many applications can tolerate some packet losses and can compensate them with mechanisms such as retransmissions, while long-term disconnections cannot be tolerated by most applications.

As another investigation, the scalability of the technique with respect to the number of coexisting TSCH networks can be observed from the results presented in Fig. 5a. As the number of coexisting networks goes up, the time hopping mechanism becomes more effective and leads to more gain in avoiding the worst cases, thus more necessary. For instance, when two TSCH networks are involved, around 25% CFR enhancement has been achieved. However, this value rises to 55% when seven networks coexist. By increasing the number of networks further to $N = 20$, the coexistence enters a region in which the chance of collision-free communication considerably decreases to even 0% in some cases. The time hopping mechanism can rescue the involved networks in such a crowded scenario as well, increasing the worst-case CFR to 25%, which is a considerable enhancement (compared to totally blocked links with CFR=0% when time hopping is not enabled). It is important to notice that having a greater number of TSCH networks with fully occupied transmission slots in the communication range of one another may be not realistic due to the short-range communication nature of the standard. However, the scenario with N=20 is tested for the sake of scalability analysis of the time hopping technique. Moving forward to even higher number of involved TSCH networks will substantially decrease the chance of successful packet transmission to the extent that the time hopping becomes even ineffective since there is no capacity in the network for communications due to very high number of independent networks. Such scenarios are considered out of the realistic scenarios or with extremely low chance of occurrence.

Time hopping affects not only CFR but also the number of burst collisions. Fig. 5b presents the number of burst collisions experienced during the performed simulations, showing that time hopping remarkably improves worst cases as well as the mean for the number of burst collisions. For instance, when packet length is 133 bytes and seven TSCH networks coexist, time hopping lessens the worst-case number of burst collisions from around 1500 instances to less than 300 cases. Again, with an increase in the length of packets and the number of coexisting TSCH networks, time hopping provides a greater positive impact showing more necessity for such a mechanism.



(a) CFR of four coexisting TSCH networks



(b) Number of burst collisions of four coexisting TSCH networks

Fig. 6: Simulation results showing the impact of time hopping interval (N_{TH}) on the performance of coexisting TSCH networks.

D. Time Hopping Configuration

The time hopping interval (N_{TH}) is an influencing parameter of the time hopping mechanism that needs to be set. In this section, we study the impact of the setting of this parameter on the data delivery reliability of coexisting TSCH networks as well as the nominal throughput to investigate the trade-off this parameter makes. This can be seen as a general guideline for configuring the time hopping mechanism. Simulations of this section were done by values mentioned in Table I.

Fig. 6 shows the achieved CFR and number of burst collisions when four TSCH networks coexist. Separate simulations with different values of N_{TH} ranging from 2 to 600 timeslots are performed. Such results show that more frequent time hopping (lower N_{TH}) enhances its impact and provides improved worst-case scenarios. This is an expected result since, with a lower time hopping interval, the mechanism will be more agile and can rescue TSCH networks from extreme coexistence scenarios in a shorter time. However, while the mean results remain almost unchanged for different N_{TH} values, lower

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199

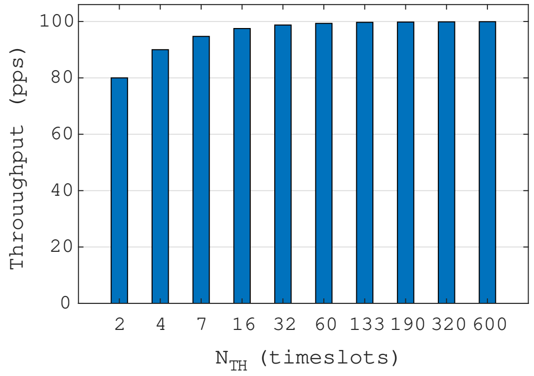


Fig. 7: Effect of N_{TH} on nominal TSCH throughput

N_{TH} values also decrease the best achieved CFR. Increasing the time hopping interval leads to a vaster distribution of CFR in different runs, which means stronger extremes on both up and down sides.

The CFR results may suggest performing time hopping very frequently to best avoid extremely low CFR due to interference from other TSCH networks. However, as discussed in Section IV-C, more frequent time hopping imposes a higher throughput overhead. To display the impact of time hopping on nominal throughput, Fig. 7 depicts the achieved throughout (timeslots or packets per second) for different time hopping intervals, using Eqn. 4 and the settings listed in Table I. Given that timeslot's length is 10 ms and provided that the network can send data packets in all timeslots, the maximum nominal throughput is 100 packets per second. As the Fig. 7 shows, with an increase in N_{TH}, throughput increases, whereas this increase can undermine the performance and effectiveness of time hopping. For $N_{TH} = 7$ and $N_{TH} = 16$, throughput overhead is as low as 3% and 1.5%, respectively. Considering the impact of CFR, N_{TH} values in range of 7 to 32 seem to be the more reasonable choices providing proper trade-off between the capability of avoiding extreme coexistence scenarios and the throughput overhead. Based on the results shown in Fig. 6 and Fig. 7, the network designers can decide about the value of N_{TH}. For such decision, specifications of the underlying application need to be taken into consideration. Specifications include the reliability requirement, sensitivity of the application to disconnections in coexisting scenarios, the expectation about the possibility that the TSCH network coexists with other TSCH networks, and the expected time duration of such envisioned coexistence scenarios.

Other settings of the time hopping mechanism are the size of THL and the time deviation values in this list. As discussed, the time deviation values should be in the range of $(0\ \ T_{ts})$. To explore the impact of these two settings on the performance of time hopping, we run a number of simulations with various settings. Two cases of $N = 7$ and $N = 20$ coexisting TSCH networks are tested each with two different values for the size of the list ($|THL| = 4$ and $|THL| = 8$) resulting in four cases. For each of these four cases, THLs are filled by three methods. As the first method, the THLs of all the coexisting TSCH networks are fixed as $THL = \{1ms, 3ms, 6ms, 9ms\}$ and $THL = \{1ms, 2ms, 3ms, 4ms, 6ms, 7ms, 8ms, 9ms\}$, respectively for cases with $|THL| = 4$ and $|THL| = 8$. As the second method, the same lists as the first method but shuffled are used for different TSCH networks. As the last tested method, the THL lists for various involved networks are filled independently by uniformly randomly picked time deviations from the range $(0\ \ 10ms)$, following the guideline mentioned in Section IV-B. Monte Carlo simulations with 20000 iterations are done for each case. Fig. 8 shows the results of these experiments. The first observation is that there is no consistent change in the performance when going from $|THL| = 4$ to $|THL| = 8$. It means that while at least four time deviation values are there in the list, the exact size of the list does not affect the effectiveness of the time hopping technique. The other observation about the time deviation values reveals that no meaningful performance difference can be seen between the first two cases in which the time deviations are fixed. However, randomly filling THLs has shown to be the most effective since these experiments clearly have resulted in the best distributions among all the performed experiments. The mean values of the collision free ratios remain very close for similar experiments. These results suggest that the best way is that each TSCH network independently sets its THL with some random time deviations from the whole range. The reason is that this way the chance that independent TSCH networks use the same THLs is very low.

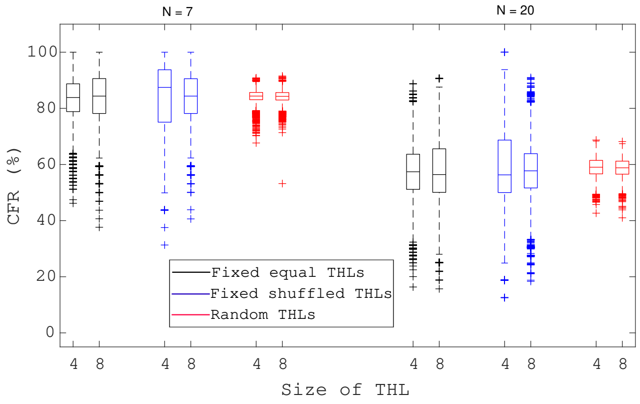


Fig. 8: The impact of the THL composition on the performance of the time hopping technique

VI. Real Implementation and Evaluations

In the previous section, the effectiveness of the time hopping technique was investigated using Monte Carlo simulations. For proof of concept and further performance analysis, the mechanism is implemented in the Contiki [7] operating system. The Contiki implementation provides the firmware that can be programmed into wireless platforms such as Z1 motes or NXP JN5168 [8] dongles. Also, the Cooja [9] simulator of Contiki allows more realistic and accurate simulations based on the specifications of real nodes. In the following, we inspect the impact of time hopping on three different scenarios using Cooja simulations. In all experiments, networks make use of settings mentioned in Table Table II. These simulations are done under controlled scenarios to be able to narrow down into details of the behavior and impact of time hopping.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199

Parameter	Value
T_{ts}	15 ms
Slotframe size	5
Mote	Z_1
N_{TH}	8
Ack transmission	Enabled
L_{pkt}	90 bytes
L_{Ack}	11 bytes

TABLE II: Simulation settings for experiments done in Cooja

A. Coexisting TSCH Networks with Channel & Time Overlap

In some extreme cases, coexisting TSCH networks may experience overlaps in both time and channel. This is the worst case of coexistence that can happen to co-located TSCH networks. When such a predicament happens, nodes are not able to communicate with their time source; they even cannot send their Enhanced Beacons (EB) and keep-alive packets. Consequently, they gradually de-synchronize until they are disassociated. This is because the time duration in which nodes have not received any packet from their time source exceeds a specified threshold defined in TSCH standard, leading nodes to leave their network. Under these circumstances, disassociated nodes can barely get re-associated to their network because incessant collisions do not allow nodes to receive EBs and join their network.

Time hopping mechanism is a way out of this dire mess. To assess the effect and an end node (of time hopping in such cases, we carried out an experiment. Three co-located and co-channel networks, each consisting of a coordinator and an end node (node1, node2, and node3 belonging to Net1, Net2, and Net3, respectively), start their slotframes simultaneously ($\Delta = 0$) with the same HSLs, and therefore have channel and time overlap. Coordinators send EBs only within the first two minutes of the experiment. Thereupon, nodes periodically (every five timeslots) send unicast packets to their coordinators. Unicast packets of the three networks collide consistently, and it is impossible for nodes to send their packets and receive Ack without collision. Fig. 9 presents a snapshot of the Cooja simulator graphical tool. As time passes, nodes go out of synchronization since none of their packets (unicasts, keep-alives, and EBs) can be sent. If networks do not perform time hopping, these ceaseless collisions happen until nodes get disassociated and leave their network. In the above-mentioned experiment, time hopping is enabled at the third minute of the experiment. It can be seen from Fig. 9 that time hopping alters the boundaries of networks' timeslots at the third minute and brings about a profound change, rescuing networks from continuous collisions, synchronization issues, and disassociation.

B. Coexisting TSCH Networks with Time Overlap

When timeslots of a number of coexisting networks overlap in time, they are still able to send and receive without collision if they select disparate frequency channels. But, as the number of coexisting networks goes up, the probability of channel overlap between some networks increases. For instance, the probability of having no channel overlaps is almost zero when the number of coexisting networks goes beyond four [6]. In these cases, the time hopping mechanism works as a practical solution as well. To examine the extent to which time hopping can help networks avoid collisions due to timeslot's overlap, we performed an experiment 12 times (Exp1 - Exp12). For each experiment, seven coexisting networks whose timeslots completely overlap one another in time (Δ=0) select a different HSL. Apart from Exp1 and Exp2, HSL of the networks in other experiments (Exp3 - Exp12) are selected randomly. For Exp1 and Exp2, HSLs are set manually to provide extremely good and bad channel overlap scenarios, respectively. Such cases are not easy to be caught when HSLs are set randomly. We set them manually to study such cases in the experiments.

In this set of experiments, each network consists of one coordinator and one not joined node. Coordinators send EBs during the first eight minutes of their operation to ensure the joining of nodes. From minute eight, associated nodes start sensing unicast packets to their coordinator every five timeslots and keep doing this for 4 minutes without the time hopping mechanism enabled. In the last four minutes of the experiments, networks perform time hopping. During these eight minutes of data packet transmissions (the second eight minutes of the whole experiment), nodes do not send EB and keep-alive packets.

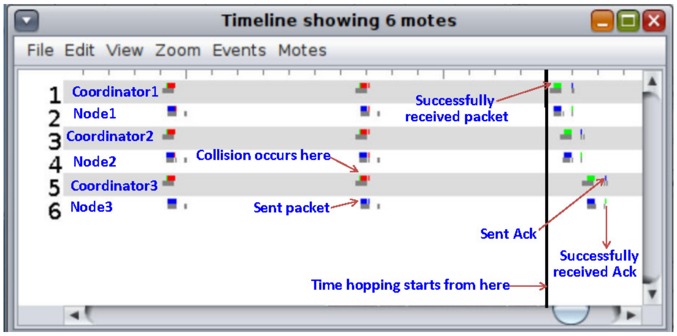


Fig. 9: A snapshot of the Cooja simulator output for the case in which networks have both time and channel overlap. Networks encounter continuous collisions until time hopping is performed from the third minute.

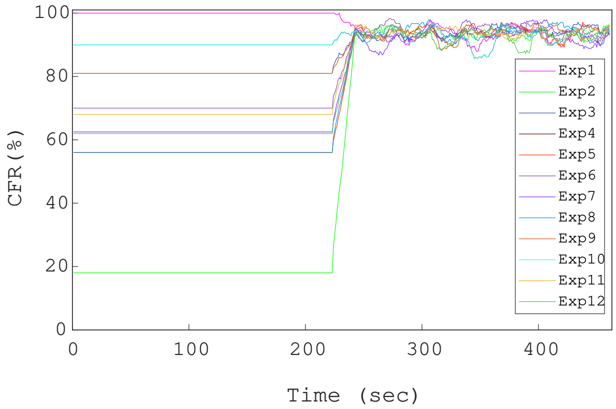


Fig. 10: Effect of time hopping on CFR of coexisting networks in Cooja whose timeslots fully overlap in time. Time hopping is disabled in the first 4 minutes and is enabled afterwards. Seven networks are coexisting and the 12 experiments are with different HSLs.

N	Fig	Network (s)	CFR W/o TH	CFR W/ TH
2	○○	NET 1-2	100	97
	●●	NET 1-2	0	95
4	○○○○	NET 1-4	100	87.6
	●●○○	NET 1-2	0	87.2
		NET 3-4	100	87.9
	●●●○	NET 1-3	0	88
		NET 4	100	88.4
	●●●●	NET 1-4	0	87.6
7	○○○○○○○	NET 1-7	100	77.5
	●●○○○○○	NET 1-2	0	77
		NET 3-7	100	77.8
	●●●○○○○	NET 1-3	0	77
		NET 4-7	100	77.8
	●●●●○○○	NET 1-4	0	77.3
		NET 5-7	100	77.6
	●●●●●○○	NET 1-5	0	77.9
		NET 6-7	100	78.5
	●●●●●●○	NET 1-6	0	78.8
		NET 7	100	79.1
	●●●●●●●	NET 1-7	0	78.6

TABLE III: Simulation results of co-located and co-channel networks with and without time hopping

The CFR values over time from the viewpoint of the first network averaged over a time window of 20 seconds are shown in Fig. 10. It is observed that prior to time hopping, CFR of experiments done with a randomly selected frequency hopping sequence list (Exp3 - Exp12) varies from 56% to 90%, yet when networks perform time hopping at the fourth minute, their CFR is improved considerably, and varies from 86% to 98% (it fluctuates around 94%). From the start of time hopping, CFR of Exp1 has decreased by approximately 6%. This is because the regular movements of timeslots made by time hopping lead networks to experience collisions at a particular point in time; afterward, networks are able to send without collision. On the other extreme, CFR of Exp12 is as low as 20%, which goes up to around 94% when time hopping is enabled.

C. Coexisting TSCH Networks with Channel Overlap

On some occasions, co-located networks switch to the same set of frequency channels; this puts networks in a precarious situation. In such cases, the relative position of their timeslots plays a pivotal role in their successful transmissions. To confirm the effectiveness of time hopping when networks have channel overlap, we carried out some experiments using 2, 4, and 7 co-located networks. In these experiments, co-located networks hop to the same sequence of channels while their timeslots' positions create different scenarios. Table III shows all possible scenarios. For instance, timeslots' position of four co-located and co-channel networks can create four scenarios: 1) no overlapped transmissions without collisions, 2) two networks with overlapped transmissions and the other two networks without collisions, 3) three networks with overlapped transmissions but the fourth without collision, and finally 4) all networks transmit with time overlap, thus collisions.

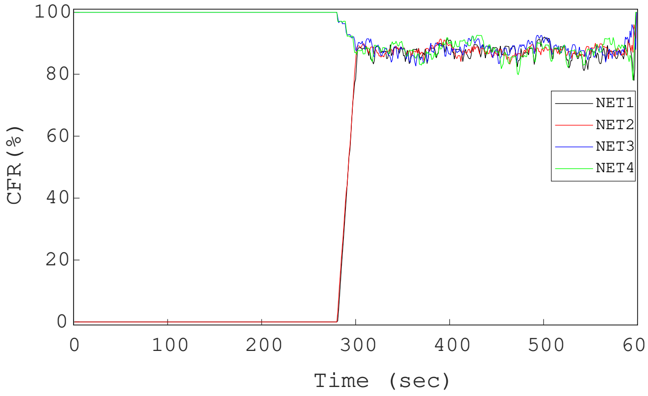


Fig. 11: Effect of time hopping on CFR of 4 co-located and co-channel networks

In this set of Cooja experiments, each network consists of one coordinator and a not joined node; they start their slotframes with different time delays to create intended scenarios in Table III. Nodes join their networks in the first eight minutes of the experiments. Then, the joined nodes start sending unicast packets every five timeslots for ten minutes. In the first five minutes of data transmission, time hopping is disabled while it is activated for the last five minutes of data transmissions. Table III shows CFR results of experiments. The first two columns show the number of co-located networks and the time overlap scenario (number of networks with time overlap). The third column shows the classification of networks in each scenario: networks that have collisions are represented in a cell, whereas networks that have no collision are categorized in another cell. In the fourth and fifth columns, the average CFR of the network in the corresponding class shown in the third column is presented. The fourth column gives CFR of networks in the first five minutes when time hopping is disabled, and the fifth column presents CFR of networks in the second five minutes when time hopping is enabled. As we expected, after using time hopping, CFR of networks that have time overlap (CFR=0) has increased significantly, whereas CFR of those without time overlap (CFR=100) has slightly decreased. Simply put, time hopping mechanism brings CFR of networks close to a specific range, so networks can rest assured that they do not experience very low CFRs due to the coexistence.

To take a closer look at the behavior and CFR of these coexisting networks, Fig. 11 gives a moving average of CFRs (with 20-second window length). In the first five minutes, Net1 and Net2 have no chance to send their packets due to collisions, while Net3 and Net4 send their packets without collisions. Networks perform time hopping starting from minute five. From then on, despite causing a modest decrease in CFR of Net3 and Net4, time hopping bails out Net1 and Net2 and substantially enhances their CFR. During the performance of time hopping, the position of networks' timeslots with respect to each other changes frequently, resulting in an oscillation of networks' CFR between 78% and 100%. Generally speaking, CFR of networks oscillates around 87%, meaning that on average, networks' CFR is 87%.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263199

VII. CONCLUSION

This paper proposes a new technique called time hopping to prevent the extreme coexistence of multiple independent TSCH networks. When several TSCH networks co-locate, there are chances of frequency and time overlaps for extended durations of time. It can substantially affect the reliability of data delivery in the TSCH networks. Time hopping inserts intentional time deviations within the coexisting TSCH networks to avoid consecutive inter-TSCH collisions. The effectiveness of the mechanism in avoiding such collisions is shown by extensive Monte Carlo simulations. Moreover, real implementation of the mechanism is done, and its impact is further investigated in more realistic settings. Results show that time hopping, on the whole, effectively improves the worst-case data delivery performance of coexisting TSCH networks. The mechanism is very light for implementation, has no computation overhead, and does not need extra packet exchange for its operation.

REFERENCES

[1] F. Veisi, M. Nabi, and H. Saidi, "An empirical study of the performance of IEEE 802.15.4e TSCH for wireless body area networks," in *Wireless Communications and Networking Conference (WCNC)*, vol. 8, pp. 1–6, IEEE, 2019.

[2] A. Elsts, X. Fafoutis, G. Oikonomou, R. Piechocki, and I. Craddock, "TSCH networks for health IoT: Design, evaluation, and trials in the wild," *ACM Transactions on Internet of Things*, vol. 1, no. 2, pp. 1–27, 2020.

[3] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, "Topology management and TSCH scheduling for low-latency convergecast in in-vehicle WSNs," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1082–1093, 2018.

[4] F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos, and N. Montavont, "From wired to wireless BMS in electric vehicles," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 255–262, 2021.

[5] C. Orfanidis, P. Pop, and X. Fafoutis, "Active connectivity fundamentals for TSCH networks of mobile robots," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 191–198, 2022.

[6] F. Veisi, M. Nabi, and H. Saidi, "Coexistence analysis of multiple asynchronous IEEE 802.15. 4 TSCH-based networks," *IEEE Access*, vol. 8, pp. 150573–150585, 2020.

[7] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceeding of the 29th IEEE Conf. on Local Computer Networks (LCN)*, IEEE, 2004.

[8] NXP Semiconductors, "JN516x IEEE802.15.4 Wireless Microcontroller." Accessed: Oct. 2018.

[9] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proceeding of the 31th IEEE Conference on Local Computer Networks (LCN)*, pp. 641–648, IEEE, 2006.

[10] W. Yuan, X. Wang, and J.-P. M. Linnartz, "A coexistence model of IEEE 802.15.4 and IEEE 802.11 b/g," in *2007 14th IEEE Symposium on Communications and Vehicular Technology in the Benelux*, pp. 1–5, IEEE, 2007.

[11] R. Natarajan, P. Zand, and M. Nabi, "Analysis of coexistence between IEEE 802.15. 4, BLE and IEEE 802.11 in the 2.4 GHz ISM band," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6025–6032, IEEE, 2016.

[12] K. P. Subbu and S. Soman, "An interference mitigation scheme for IEEE 802.15. 4 networks under IEEE 802.11 b/g interference," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, 2014.

[13] Q. Javed and R. Prakash, "Exploring alternatives to channel change for improving the coexistence of IEEE 802.15. 4 and IEEE 802.11," in *2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 45–52, IEEE, 2012.

[14] Y. Yan, P. Yang, X.-Y. Li, Y. Zhang, J. Lu, L. You, J. Wang, J. Han, and Y. Xiong, "Wizbee: Wise ZigBee coexistence via interference cancellation with single antenna," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2590–2603, 2014.

[15] P. Yang, Y. Yan, X.-Y. Li, Y. Zhang, Y. Tao, and L. You, "Taming cross-technology interference for Wi-Fi and ZigBee coexistence networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 1009–1021, 2015.

[16] M. Chwalisz and A. Wolisz, "Towards efficient coexistence of IEEE 802.15.4e TSCH and IEEE 802.11," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–7, IEEE, 2018.

[17] P. Li, T. Vermeulen, H. Liy, and S. Pollin, "An adaptive channel selection scheme for reliable TSCH-based communication," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 511–515, IEEE, 2015.

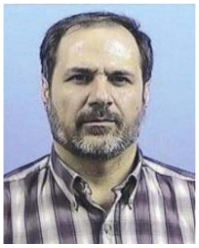
[18] O. Carhacioglu, P. Zand, and M. Nabi, "Cooperative coexistence of BLE and time slotted channel hopping networks," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–7, IEEE, 2018.

[19] S. B. Yaala, F. Théoleyre, and R. Bouallegue, "Performance study of co-located IEEE 802.15.4-TSCH networks: Interference and coexistence," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 513–518, IEEE, 2016.

[20] T. van der Lee, S. Raza, G. Exarchakos, and M. Gunes, "Towards co-located TSCH networks: An inter-network interference perspective," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2018.

Majid Nabi (S'08-M'13) received the B.Sc. and M.Sc. degrees both in computer engineering from Isfahan University of Technology and Tehran University, respectively. He received the Ph.D. degree in electrical and computer engineering from Eindhoven University of Technology (TU/e), Eindhoven, the Netherlands in 2013. He is currently an assistant professor with the Department of Electrical Engineering at TU/e, and Isfahan University of Technology. His research interests include efficient and reliable networked embedded systems, low-power wireless sensor networks, and internet-of-things. He is a member of IEEE.

Mina Habibollahi received the B.Sc. degree in telecommunication engineering from the Kermanshah University of Technology, Kermanshah, Iran, in 2016, and the M.Sc. degree in telecommunication engineering from the Isfahan University of Technology, Isfahan, Iran in 2022. Her research interests include wireless networking, internet-of-things and machine learning.



Hossein Saidi received B.S and M.S. degrees in Electrical Eng. in 1986 and 1989 respectively, both from Isfahan University of Technology (IUT), Isfahan Iran. He also received D.Sc. in Electrical Eng. from Washington University in St. Louis, USA in 1994. Since 1995 he has been with the Dept. of Electrical and Computer Engineering at IUT, where he is currently Full Professor. His research interest includes high speed switches and routers, communication networks, QoS in networks, security, queueing system and information theory. He holds 4 USA and one international patents and has published more than 100 scientific papers. He is the recipient of several awards including: 2006 ASPA award (The 1st Asian Science Park Association leaders award) and the Certificate award at 1st National Festival of Information and Communication Technology (ICT 2011) both as the CEO of SarvNet Telecommunication Inc.