Cooperative Coexistence of BLE and Time Slotted Channel Hopping Networks

Onur Carhacioglu*†, Pouria Zand* , Majid Nabi†
*Holst Centre / IMEC-NL, High Tech Campus 31, 5656 AE Eindhoven, The Netherlands
† Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands
Email: onur.carhacioglu.ext@imec-nl.nl, pouria.zand@imec-nl.nl, m.nabi@tue.nl

***Abstract*—The Bluetooth Low Energy (BLE) and Time Slotted Channel Hopping (TSCH) mode of the IEEE 802.15.4 are two of the most widely used technology standards for Wireless Sensor Networks (WSNs). In many applications, both technologies need to be used in the same environment to fulfill application requirements. However, since they share the same 2.4 GHz ISM band, such networks may suffer from cross-technology interference, which decreases the reliability of the network. To solve this problem, we propose a cooperative coexistence solution for BLE and TSCH networks in which joint time-slot and channel hopping synchronization are performed. The proposed solution uses a scheduling matrix to model the resource usage of the networks. Following this, the overlaps in this matrix are eliminated by rescheduling the transmissions of the networks. The proposed solution does not require any protocol change. The performance of the proposed cooperative coexistence mechanism is evaluated using experiments with real wireless devices. The results of those show that our proposed solution considerably decreases Packet Error Rate (PER); an improvement of up to 45% PER is observed.**

***Index Terms*—Cooperative Coexistence, Bluetooth Low Energy, IEEE 802.15.4e, TSCH, Wireless Sensor Networks**

I. Introduction

With the emergence of Internet-of-Things (IoT), the number of connected devices has rapidly grown. The communication between these devices is standardized by several technology standards. Each standard aims to provide connectivity for special use cases. For indoor Wireless Sensor Networks (WSNs), the most widely used specifications are IEEE 802.15.4 and Bluetooth Low Energy (BLE). TSCH is an amendment to the MAC operation of IEEE 802.15.4, which improves reliability by using both time division and channel hopping.

Many IoT applications require a highly reliable communication. For example, in-vehicle networks require high reliability, since missing data may pose a high risk. High reliability is also important in industrial automation in which failure in a small part can damage the whole system. To ensure high reliability, the TSCH standard is equipped with physical and Medium Access Control (MAC) layer mechanisms. On the other hand, the in-vehicle applications may need to have connectivity with smart devices of the driver to exchange status and to control information. BLE is a low-power communication standard that provides an easy connection of mobile hand-held devices to sensors and actuators in a personal environment. In short, both technologies are very likely to be used by a single system in the same environment. The in-vehicle network is presented here only as an example of the usage of such systems with both communication technologies.

TSCH uses time division by allocating periodic timeslots for each pair of nodes in the network. Furthermore, each timeslot uses a different frequency channel after each period (called slotframe). BLE uses Connection Intervals (CIs) to provide periodic transmissions. Also, BLE uses adaptive frequency hopping in which the transmission channel changes after every CI. Therefore, both TSCH and BLE use methods that are a combination of time and frequency division mechanisms. These mechanisms prevent packet collisions within their own network. However, TSCH and BLE use the same 2.4 GHz ISM band. If both of these networks are used in the same location, Cross-Technology Interference (CTI) decreases their reliability.

In order to cope with the CTI problem between BLE and TSCH networks, a coexistence mechanism is required. In such a mechanism, a network predicts the transmission of the other network. Based on this prediction, the network schedules its own transmissions in such a way that these will not collide with the predicted communications of the other network. In this paper, we propose to fortify this system with a smart multi-radio gateway that supports both BLE and TSCH standards. This smart gateway implements a *cooperative coexistence* in which the networks exchange their transmission schedules. This way, the communication schedules of both networks are established by the smart gateway. This makes it possible to develop conflict-free schedules for both, coexisting, technologies.

To enable the cooperative coexistence, we use a scheduling matrix that models the upcoming time and frequency use of each network. In order to create this matrix, the anchor point of connection-oriented BLE and the beginning of the first time-slot of TSCH need to be aligned in time. That way, each element of the matrix

The first two authors have equally contributed to this work represents the occupation of a specific time interval and frequency channel. An upcoming interference is detected if both the TSCH and BLE networks occupy the same matrix element. Possible interference is then avoided by rescheduling the transmission pattern of one of the networks.

This paper is organized as follows. The upcoming section discusses relevant preceding research that touches upon our area of inquiry. Following this, Section III provides more of an insight into the specifications of the BLE and TSCH standards. After discussing the coexistence problem in more depth, Section IV presents the proposed cooperative coexistence solution. The details of the experimental setup and the achieved results are presented in Section V. Section VI summarizes the main findings while also discussing possibilities for future research.

II. Related Work

The detrimental effect of CTI between co-located radios has been widely researched, with many authors subscribing to the notion that MAC layer coexistence solutions can alleviate the effect of CTI. In [1], a collaborative coexistence mechanism between co-located WLAN, classic Bluetooth and WiMAX radios is introduced. In this work, a synchronized activity bitmap generation is used to avoid CTI, where the radio manager receives the traffic pattern from the first radio (e.g., Bluetooth). Accordingly, the radio manager generates the traffic pattern bitmaps. Since the scheduler (or base-station) of the second radio is not co-located in the smart-device, it sends the bitmaps as a recommendation to the non-co-located base-station of the second radio (e.g., WiMAX). In [2], a collocated collaborative coexistence mechanism between BLE and LTE radios is introduced, where because of imperfect TX filter design, simultaneous transmission and reception of these two radios can suffer from Adjacent-Channel Interference (ACI). To tackle this problem, [2] presents some coexistence methods, where a BLE radio dynamically adjust Time Inter Frame Space (T-IFS) to mitigate coexistence interference. However, the proposed methods imposed changes in the BLE standard.

In [3] and [4], the effect of CTI is minimized for TSCH networks, by using a dynamic channel whitelisting technique. In this process, the TSCH channels are frequently sensed in order to assess their quality, after which only good quality channels are selected for communication. In this method, sampling and whitelisting are either done by the TSCH coordinator [3] or by each TSCH node in the network in a distributed manner [4]. However, both [3] and [4] use non-cooperative coexistence, since the interference from any other technology is predicted by the TSCH network. Furthermore, and unlike our proposed solution, these two technologies do not share information.

There are other studies, i.e. [5] and [6], about cooperative coexistence of different communication protocols that relate to our work. In [5], two methods are proposed to ensure a smooth time domain cooperative coexistence of BLE and IEEE 802.15.4. The first method entails that BLE transmissions adapt to the IEEE 802.15.4 schedule, while the second method ensures the opposite: IEEE 802.15.4 transmissions adapt to the BLE schedule. Thus, only one technology is active at a time: while the other one sleeps. [6] proposes a cooperative coexistence mechanism for IEEE 802.15.3 and 802.15.4 protocols, based on the physical and MAC layer characteristics of both protocols. They assume a beacon-enabled mode of IEEE 802.15.4, and exchange synchronization information between the two networks. However, both [5] and [6] do not consider the frequency domain characteristics of the networks, which can increase the number of coexistence options.

In [7], a radio scheduler receives both time and frequency information from the connected WLAN, classic Bluetooth and WiMAX radios. The scheduler creates a time and frequency mask based on the time and frequency intervals of the lowest prioritized technology. Using this matrix, the lowest prioritized radio avoids transmissions when there is a possible collision. This method is an efficient way to model channel occupancy. Our proposed method also uses a scheduling TSCH-BLE matrix, the dimensions of which are calculated based on the greatest common divider of the dimension of the matrix in both technologies. When a possible collision is predicted, our scheduler reschedules the transmissions.

III. Background

A. Bluetooth Low Energy (BLE)

BLE is a type of wireless personal area network technology, which operates in two different modes: connection-less broadcasting and connection-oriented periodic data exchange. This paper focuses on the connection-oriented mode of BLE, as the only way to provide time structured multi-channel communication using duty cycles and channel hopping.

The duty cycles and channel hopping of BLE are depicted in Fig. 1, in which the connection event represents the packet transmission times, the connection interval represents the period of the transmissions, and f(k) function represents the channel. Connection interval can take some possible values that are given by the Bluetooth standard, in the range between 7.5 ms and 4 s (with increments of 1.25 ms). BLE has 40 back-to-back channels in the 2.4 GHz ISM band, each with 2 MHz bandwidth, as is shown in Fig 3. Up to 37 of these channels can be used for data communication by performing channel hopping. After each CI, BLE hops from one channel to another in a deterministic way, using its channel selection algorithm as defined by the standard.

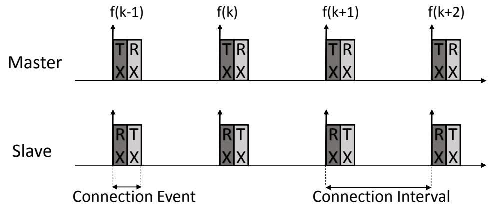


Fig. 1. Connection timing and channel hopping in BLE

B. Time Slotted Channel Hopping (TSCH)

TSCH communication is divided into both time and frequency domains. In the time domain, time is divided into timeslots ($10ms$). From the initiation of the connection, TSCH timeslots are counted by an index, named Absolute Slot Number (ASN). In the frequency domain, TSCH transmission is divided into 16 channels in the 2450 MHz band with 2 MHz channel bandwidth, as is shown in Fig 3. In a network, the communication of two devices with a specific channel offset and timeslot is called a link. In a link, transmission is repeated with a TSCH period, which is called slotframe. The transmission channel is calculated by considering ASN, a predefined channel offset value, $offset_{CH}$, and the number of active TSCH channels, num_{CH} for a given $macHoppingSequenceList$ as following: $CH = List[(ASN + offset_{CH})\%num_{CH}]$.

IV. Cooperative Coexistence of BLE-TSCH

This work focuses on applications in which both BLE and TSCH networks exist in a single system. Thus, we are able to have a multi-radio gateway operating as a coordinator/master for both TSCH and BLE networks. An in-vehicle WSN is an example of such a system. The multi-radio gateway designs the communication schedule of both networks in such a way that the chances of collision are minimal. It is done by extracting a scheduling matrix of the usage of communication resources (channels and communication interval) by the two networks. Subsequently, certain adaptations are made to the network schedules to avoid collisions.

Fig. 2 shows the structure of a multi-radio gateway that includes a TSCH coordinator and a BLE central. The gateway further includes an orchestrator, which schedules the communications of BLE and TSCH. The orchestrator receives usage information from BLE and TSCH, and in turn, produces a time and channel scheduling for each radio.

In this section, we first introduce and explain how our proposed multi-radio gateway constructs the scheduling matrix and the way it predicts upcoming collisions. Then we propose solutions for cooperative coexistence in certain scenarios.

A. Scheduling Matrix

The orchestrator constructs a scheduling matrix, which can predict the upcoming communication for

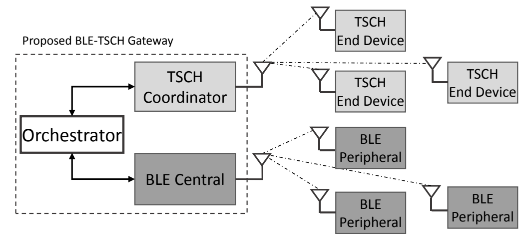


Fig. 2. A multi-radio BLE-TSCH Gateway and end-devices

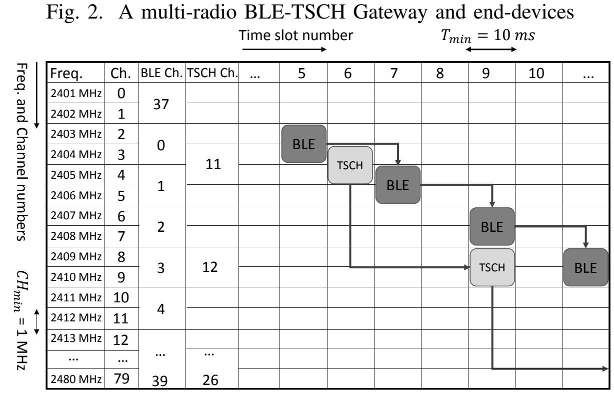


Fig. 3. A sample of BLE-TSCH scheduling matrix, where channel width (CH_{min}) is 1MHz and time-slot width (T_{min}) is 10ms

both BLE and TSCH networks. Thanks to this matrix, the orchestrator can re-schedule future communications based on different objective functions (e.g., to improve reliability or to decrease latency). Based on the received communication parameters, the orchestrator is able to determine a minimum time slot (T_{min}) and a minimum frequency (or channel) bandwidth (CH_{min}), which are respectively equal to the time-width and channel-width of the scheduling matrix. T_{min} is the Greatest Common Divisor (GCD) of the communication interval of both BLE (i.e., connection interval) and TSCH (i.e., timeslot size). CH_{min} is calculated based on the GCD of the channel bandwidth and the channel offset of both BLE and TSCH.

The orchestrator aligns the anchor point of BLE with the beginning of the first timeslot of TSCH. Then, it creates a scheduling matrix with a unit cell and a unit width equal to CH_{min} and T_{min}. Fig. 3 shows an example of such a scheduling matrix. In this figure, the scheduler minimum timeslot length is $T_{min} = 10ms$, and the minimum frequency bandwidth is $CH_{min} = 1MHz$. The figure also depicts BLE communication for one pair and TSCH communication for another pair, both hopping over channels. In this example, BLE has a communication interval of 20ms while TSCH has a slotframe of size 30 ms (3 timeslots).

The orchestrator can be adapted to consider different objective functions while scheduling the communications between radios (for instance to improve reliability or to decrease latency). For example, in order to provide reliable communication, the orchestrator can schedule the transmissions in the scheduling matrix in such a way that they rarely (ideally never) overlap. For this objective function, the scheduler minimizes the overlaps in the scheduling matrix. Similarly, in order to improve latency, the scheduler can maximize overlap to create a communication blocker or minimize the horizontal distance between the transmissions to minimize end-to-end latency. In this paper, we only consider the objective function that minimizes the overlaps in the scheduling matrix (higher reliability as the objective).

B. *Coexistence Solutions*

Our BLE and TSCH cooperative coexistence solutions can be split into two categories: static and dynamic solutions. In the static one, the TSCH link never interferes with BLE. Therefore, a static solution, i.e. one requiring only one update to avoid interference, suffices to decrease the Packet Error Rate (PER). In the dynamic case, there are TSCH links that occasionally do not interfere with BLE, although none of the TSCH links will display the behavior mentioned in the static case. Using the scheduling matrix, it is possible to predict which link will collide with the BLE connection event and at which time-slot and channel number. In order to avoid this type of interference, a dynamic solution, which requires periodic monitoring and updating of the scheduled cells, is needed.

In line with our focus on the above-mentioned two scenarios, we propose four different cooperative coexistence solutions to avoid interference. Frequency, time, and pattern division provide solutions for the static case, requiring only one link update. A dynamic-link update solution provides cooperative coexistence for the dynamic case, demanding periodic link updates. These solutions are illustrated in Fig. 4 and explained in the following.

a) Frequency Division Solution: The first solution involves separating the channels in the 2.4 GHz spectrum to prevent any TSCH channels from interfering with the BLE channels under use. This example is a frequency domain solution. As such, it does not consider the time domain characteristics of any of the used protocols. Fig. 4(a) depicts this solution.

b) Time Division Solution: In order to avoid interference, a specific TSCH link, which is never scheduled simultaneously with the BLE connection event, is added in the link tables of both the TSCH coordinator and the TSCH end device. As a result, this link never interferes with BLE. In other words, TSCH transmissions take place in timeslots that will never overlap with BLE transmissions in the time domain. In short, this set up offers a time domain solution that will not consider the frequency domain characteristics of any of the protocols. Fig. 4(b) depicts this solution.

c) Pattern Division Solution: BLE and TSCH transmissions may overlap in the time domain. However, when they overlap in time, they should use different

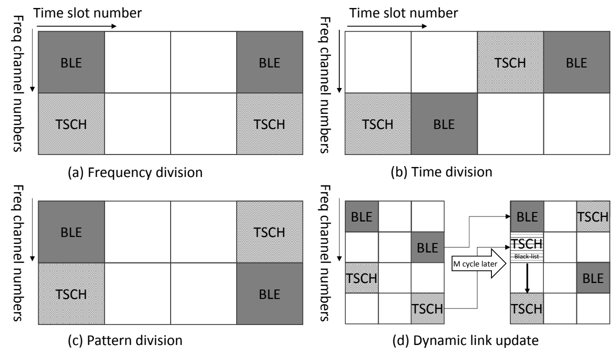


Fig. 4. Frequency (a), time (b), pattern (c) division solutions and dynamic link update solution (d) for different coexistence examples

frequency channels. This solution differs from the first one, since it does not limit the number of available channels for each protocol. Instead, this method uses the division of the hopping patterns. Fig. 4(c) depicts this solution.

d) Dynamic Link Update Solution: To apply this method, the orchestrator utilizes the scheduling matrix to continually predict upcoming communication and upcoming collisions. Accordingly, the scheduler either constructs new links for TSCH, or defines new parameters for the BLE network. The time and frequency domain characteristics of dynamic solutions are shown in Fig. 4(d). As is shown in this figure, the scheduler predicts the upcoming collision after m cycles and it reschedules the TSCH communication to avoid the interfering cell.

In more complex real-life scenarios, the number of channels can be higher and there might be some constraints regarding the period of the update messages. These limitations are caused by processing delays and response speed. To tackle these issues, we introduced an update window. An update window is the minimum time interval between two consecutive update messages. The orchestrator exploits the scheduling matrix to construct all the upcoming communications and calculates the number of expected collisions for the existing links. Following this, it either chooses the link that has the lowest chance of collision or it defines a new link that will have the lowest chance of collision. Afterwards, the scheduler sends a link update command to delete or modify the existing links or to add new ones. This update message can only be sent at the beginning of an update window.

The performance of the dynamic solution depends on the available links and the update window size. If the window size is smaller than the minimum time between two consecutive collisions, this solution can fully achieve PER due to the collisions. However, even in experiments with larger window sizes, a significant improvement in the PER is observed.

V. Performance Evaluation

In this section, we first describe the setup that we used for experiments. Then the different scenarios and their results are discussed.

A. Test Setup

In our experiments, we have one smart multi-radio gateway, one TSCH end device and one BLE peripheral. The smart gateway consists of one Raspberry Pi (i.e., orchestrator), and two co-located radio modules: one TSCH coordinator module and one BLE controller module. All these four radio devices use Texas Instruments Sensortag CC2650 as hardware. This network is depicted in Fig. 2.

On the TSCH side, the uplink data packets are transmitted from the end device to the gateway. The TSCH MAC layer re-transmissions are deactivated to ensure that each packet gets transmitted only once, allowing us to directly translate the packet error into a packet loss. On the BLE side, each connection event involves the transmission of five packets and their ACKs.

In the experiments, a large number of data packets is transmitted on the TSCH connection, while the PER is measured based on the error ratio of these packets. In other words, BLE is used as an interfering network while TSCH is used as the affected network. For the dynamic experiments, the update window size is 700 ms. Table I shows the test parameters and the PER results.

Among the static solutions, the frequency division solution requires BLE to adapt by changing its own schedule based on TSCH transmissions. However, in time division and pattern division solutions, TSCH adapts itself on the basis of BLE transmissions. Therefore, in the frequency division solution, BLE is used as the adapting network. In the time division and pattern division solutions, the adapting network is TSCH.

B. Static Case Experiments

The static experiments require either hard-coded schedules or only a one link update at run-time. This is because the newly chosen link never interferes with the BLE transmissions in the static cases.

a) Frequency Division Solution: This solution proposes the separation of transmission frequencies of BLE and TSCH. This solution is tested in Sc. 1 of the Table I showing a significant improvement in PER. In this scenario, the TSCH connection suffers from CTI because BLE and TSCH channels are not disjoint. After rescheduling, Channel 32 and Channel 36 are activated for BLE instead of Channels 3 and 22. Therefore, the TSCH and BLE channels become disjoint, which decreases the PER from 46.4% to 7.4%.

b) Time Division Solution: This solution proposes the separation of transmissions in the time domain, and it is tested in a set-up that reflects Sc. 2 in Table I. In the default setup of this scenario, the TSCH connection has a high PER: TSCH and BLE have the same transmission period and they always transmit at the same time. Approximately, half of their transmissions fail, because Channel 3 of BLE and Channel 12 of TSCH use the same frequency. However, when the time division solution is applied, TSCH transmissions are shifted 50 ms. As a result, the PER decreases from 46.4 % to 8.1%.

c) Pattern Division Solution: This solution provides an improvement in PER by separating the hopping patterns of BLE and TSCH. In other words, using this solution provides disjoint hopping patterns to BLE and TSCH. The main advantage of the pattern division solution over time and frequency division solutions is that this method does not put any limitations on time and frequency resources. The solution can be applied in two ways i) by choosing a CTI-free pattern from given options, or ii) by constructing a new CTI-free pattern.

The first option is tested in Sc. 3, where the PER is improved by choosing a CTI-free pattern. This pattern ensures that TSCH does not use Channel 12, while BLE uses the interfering Channel 3. The second option is tested in Sc. 4, where the list of the disjoint TSCH channel hopping sequence is constructed. Firstly, the BLE channel hopping sequence is calculated. Following this, a disjoint TSCH channel hopping sequence, which does not overlap with the BLE channel hopping, is calculated. Finally, the network uses the disjoint TSCH channel hopping sequence.

C. Dynamic Case Experiments

a) Dynamic Link Update discussions: This section discusses the dynamic case experiments. In these experiments, static solutions do not work, in the absence of a unique link that provides CTI-free communication. In view of these constraints, we propose a dynamic link update solution in order to minimize the PER.

Given the characteristics of channel distribution algorithms, the PER of scenarios with only one interfering channel can be calculated using Eq. 1.

$$\begin{aligned}
&Ratio_{time} = \frac{TSCHperiod}{LCM(CI, TSCHperiod)}, \forall k \in \mathcal{N}, \\
&num_{ActiveTSCH} = f((TSCHperiod \times k)\%num_{CH}) \\
&Ratio_{freq} = \Big(num_{BleCH} \times num_{ActiveTSCH}\Big)^{-1} \\
&PER = Ratio_{time} \times Ratio_{freq}
\end{aligned} \tag{1}$$

where $TSCHperiod$ is the slotframe size in TSCH. The f function calculates the number of possible values that the modular operation can take for all integer k values. $num_{ActiveTSCH}$ is the number of TSCH active channels that are used during radio active times, num_{BleCH} is the number of BLE channels and, $Ratio_{time}$ is equal to the TSCH period divided by the Least Common Multiple (LCM) of the BLE interval and TSCH period. $Ratio_{time}$ represents the ratio of

TABLE I
TEST SCENARIOS AND THE RESULTANT PERS

	Case	Solution	BLE channels	TSCH channels	BLE interval (ms)	TSCH period (ms)	PER without solution	PER with solution
Sc. 1	Static	Freq. Division	3,22 →32,36	26,12	100	100	46.4 %	7.4 %
Sc. 2	Static	Time Division	3,22	26,12	100	100	46.4 %	8.1 %
Sc. 3	Static	Pattern Division	3,6,18,22	26,12	100	70	2.7 %	0.2 %
Sc. 4	Static	Pattern Division	3,22	pattern introduced	100	100	46.4 %	0.8 %
Sc. 5	Dynamic	Link Update	3,6,18,22	26,12	100	70	2.7 %	1.2 %
Sc. 6	Dynamic	Link Update	3,22	26,12	100	70	5.9 %	2.5 %
Sc. 7	Dynamic	Link Update	3,22	12	100	70	5.1 %	1.3 %
Sc. 8	Dynamic	Link Update	3,6,18,22	23,16,26,12	100	70	2.2 %	1.1 %
Sc. 9	Dynamic	Link Update	3,8,17,22	20,18,14,12	100	70	5.4 %	1.5 %
Sc. 10	Dynamic	Link Update	3,6,18,22	26,12	100	130	3.2 %	1.7 %

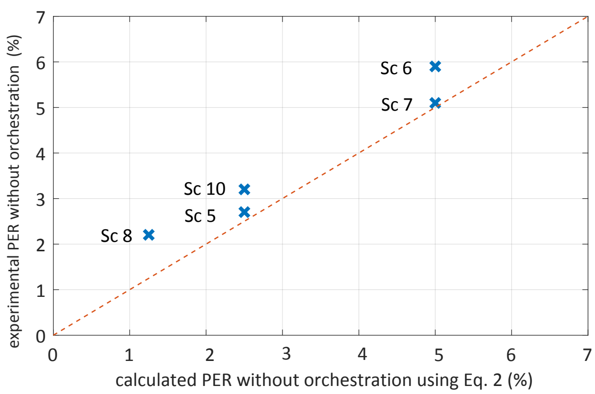


Fig. 5. Calculated PER is compared with the experimental results of dynamic experiments. Ideally, all the experimental results are expected to cross the dashed line, but the noise increases the experimental PER

colliding transmissions to total transmissions in the time domain, while $Ratio_{freq}$ represents the same ratio in the frequency domain. Using Eq. 1, Figure 5 is created. In this figure, the calculated PER values are compared with actual PER values of Sc. 5, 6, 7, 8, and 10, without any orchestration. This figure shows that while the PER calculation is accurate, the experimental PER can be 1% higher than the calculated one. This difference can be explained by noise (Sec. V-E) in the experiments.

b) Dynamic Link Update solution: In order to test the dynamic-link-update solution, six different experiments have been conducted. These experiments are Sc. 5, 6, 7, 8, 9, and 10 in Table I. Among these experiments, Sc. 5, 6, 7, and 8 present the effect of the dynamic-link-update solution, when the CI and TSCH period are 100 ms and 70 ms respectively, and only one channel interferes. Sc. 9 presents the performance of the dynamic-link-update solution when more channels interfere. In this scenario, all BLE and TSCH channels interfere with each other. Sc. 10 shows that the dynamic-link-update solution also works in other (CI, TSCH period) combinations as well.

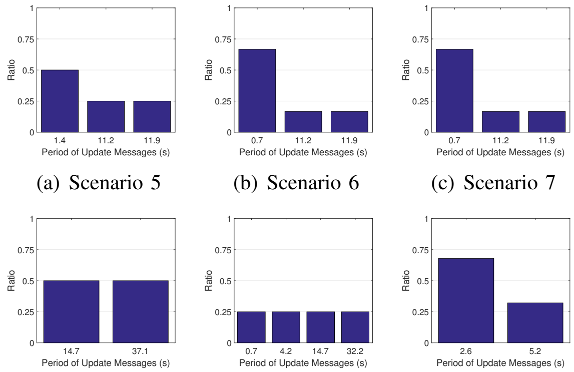
In the dynamic-link-update solution, the link update commands introduce additional message traffic to the network. Therefore, this solution can dynamically monitor and update the network performance at the cost of extra power consumption. In this sense, the dynamic-link-update solution differs from the static solutions, which do not create message overhead. This cost is evaluated by considering the ratio of overhead messages to the total number of messages.

TABLE II
THE OVERHEAD CREATED BY LINK UPDATE MESSAGES

Scenario	PER improvement	average update period (s)	overhead (%)
Sc. 5	2.7 → 1.2	6.475	1.07
Sc. 6	5.9 → 2.5	4.32	1.59
Sc. 7	5.1 → 1.3	4.32	1.59
Sc. 8	2.2 → 1.1	25.9	0.27
Sc. 9	5.4 → 1.5	12.95	0.54
Sc. 10	3.2 → 1.7	3.44	3.64

During the experiments, it was observed that the link update messages are not generated within a constant period. Instead, each time, the period of link update messages changes between a couple of possible values. This change is not probabilistic, but deterministic with a repeating sequence. Also, the possible values are multiples of an update window, i.e. 700 ms. For example, in Sc. 5, the time difference between each link update is given in the following set: {11.9, 11.2, 1.4, 1.4, 11.9, 11.2, 1.4, 1.4 ...}. Therefore, in Sc. 5, the distribution of the link update messages is 11.9 with 25%, 11.2 with 25%, and 1.4 with 50%. Using these values, the average link update period is calculated as 6.475. Table II and Figure 6 display the period of link update messages, based on the same deterministic method. The overhead of the link update messages is calculated as the ratio of the update messages to the total number of messages.

Table II shows that PER is improved in all of the dynamic scenarios when using the dynamic-link-update solution. However, this solution did not achieve 0% PER. There are two reasons for this. Firstly, PER can be affected by noise, which is explained in more depth in Section V-E. Secondly, if both old and new links collide within the length of the update window, a possible collision cannot be avoided.



(a) Scenario 5 (b) Scenario 6 (c) Scenario 7

(d) Scenario 8 (e) Scenario 9 (f) Scenario 10

Fig. 6. Link update message period for each dynamic scenario

From the scenarios with constant periods and one interfering channel, i.e. Sc. 5, 6, 7, 8, the effect of the number of channels on the overhead can be observed. When the number of BLE channels or $num_{ActiveTSCH}$ increases, the overhead will decreases. Also, Sc. 8 and 9 show that as the ratio of interfering channels grows, so does the overhead. Lastly, in Sc. 5 and 10, an increase in (TSCH period/CI) leads to more overhead.

D. Discussion on PER for large-scale network

In a real deployment of a dense network, with 10 to 100 nodes installed in the same transmission range, the data generation period of a single node cannot be as small as 100 ms, due to the limitations of the network bandwidth. In such a deployment, the entire interfering wireless network functions as a single interfering node with a higher data rate (10 to 100 times more than a single node). This set up decreases the reliability of the affected wireless network.

E. Discussion on PER uncertainty

The PER values discussed in the experiments are measured using the data packets of TSCH. TSCH devices not only transmit data packets, they also periodically broadcast beacons to advertise the presence of the TSCH network. These beacon packets play a critical role in initiating and preserving the connection.

If both a beacon and a data packet are scheduled for a specific time-slot, then the beacon packet would be transmitted while the data packet is ignored. This is because the 802.15.4 standard does not support simultaneous multi-channel transmission. Consequently, the transmission of beacon packets increases the PER, although it has no relation with CTI. In order to compensate for this effect, a different and prime number period is assigned as the beacon period. However, there remains some noise on PER calculation.

In Sc. 2 and 4, the beacon packets are transmitted more frequently, because these scenarios have a high PER before the relevant solution is applied at run-time. If enough beacon packets are not transmitted, these scenarios suffer from disconnections. In these scenarios, up to 8% of the PER is caused by beacons. In the other scenarios, the beacon period is set higher and the effect of beacon noise is less than 1%.

VI. Conclusion

In this paper, we have proposed a joint time-slot and channel-hopping synchronization for collocated BLE and TSCH networks. The proposed solution uses a scheduling matrix to model the resource usage by the networks. In addition, four cooperative coexistence solutions have been proposed. These solutions improve the reliability of cooperative and collocated TSCH and BLE networks by reducing inter-network collisions.

These solutions were tested using real-world experiments. In static scenarios, up to 45.6% improvement in PER was observed, while in the dynamic scenarios it was up to 3.9 %. Also, the effect of the number of channels and the period lengths on PER were discussed. While, these solutions were first and foremost proposed to improve PER, however, similar solutions can be applied to other objectives, such as minimizing latency or lowering the number of re-transmissions.

Acknowledgment

The joint time-slot and channel-hopping orchestration solution that we have described in this paper has been filed for patent. This work has been performed partially within the SCOTT project (SCOTT (www.scott-project.eu) has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway).

References

[1] L.-C. Ko, H.-K. Hsu, C.-C. Lee, I.-K. Fu, and J.-J. Su, "Synchronized activity bitmap generation method for co-located coexistence (clc) devices," Aug. 16 2016, uS Patent 9,420,599.

[2] L.-C. Ko, H. Ye, S.-J. Hsu, and I.-K. Fu, "Bluetooth low energy and lte coexistence enhancements," Jul. 7 2015, uS Patent 9,078,275.

[3] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, "Enhanced time-slotted channel hopping in wsns using non-intrusive channel-quality estimation," in *2015 IEEE 12th MASS*, Oct 2015.

[4] ——, "Dependable interference-aware time-slotted channel hopping for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 14, no. 1, Jan. 2018.

[5] O. Carhacioglu, P. Zand, and M. Nabi, "Time-domain cooperative coexistence of ble and ieee 802.15.4 networks," in *IEEE 28th PIMRC*, Oct 2017.

[6] M. D. Sanctis, M. Monti, M. Ruggieri, and R. Prasad, "A collaborative coexistence mechanism for IEEE 802.15.3 and 802.15.4 WPANs," in *IEEE 20th PIMRC*, 2009.

[7] P. D. M. Truong, M. E. Russell, and I. S. Sen, "Method and apparatus for multi-radio coexistence," Jul. 22 2014, uS Patent 8,787,468.