

Whitelisting Without Collisions for Centralized Scheduling in Wireless Industrial Networks

Vasileios Kotsiou, Georgios Papadopoulos, Periklis Chatzimisios, Fabrice Theoleyre

▶ **To cite this version:**

Vasileios Kotsiou, Georgios Papadopoulos, Periklis Chatzimisios, Fabrice Theoleyre. Whitelisting Without Collisions for Centralized Scheduling in Wireless Industrial Networks. IEEE Internet of Things Journal, 2019, 6 (3), pp.5713-5721. 10.1109/JIOT.2019.2905217 . hal-02323131

HAL Id: hal-02323131
https://hal.science/hal-02323131

Submitted on 7 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Whitelisting without Collisions for Centralized Scheduling in Wireless Industrial Networks

Vasileios Kotsiou, Georgios Z. Papadopoulos, *Member, IEEE* Periklis Chatzimisios, *Senior Member, IEEE* Fabrice Theoleyre, *Senior Member, IEEE*

***Abstract*—Industrial applications require more and more low-power operation and high-reliability (close to 100%). Since traditional low-power radio technologies are sensitive to external interference, many recent standards implement frequency hopping schemes. For instance, IEEE 802.15.4-2015 Time Slotted Channel Hopping (TSCH) relies on a deterministic schedule of data transmissions combined with a pseudo-random frequency hopping scheme to improve the reliability. Unfortunately, specific radio channels keep on increasing the average number of retransmissions. Using a subset of the best radio channels (whitelisting) helps to improve the reliability, but may create collisions when used improperly. We here investigate the most accurate techniques to use only the best radio channels while still providing deterministic performance. We propose to group the links per timeslot, allocating them either to the same whitelist or even appropriately re-ordering them to avoid collisions. Finally, we evaluate the performance of the different whitelisting schemes using an experimental dataset from FIT IoT-LAB platform, proving the relevance of such approach to improve the reliability.**

***Index Terms*—Industrial Internet of Things; IIoT; IEEE 802.15.4-TSCH; TSCH; 6TiSCH; Centralized Scheduling; Whitelisting; Blacklisting; Collisions; Deterministic.**

I. Introduction

Industry 4.0 designates the novel industrial revolution that originally initiated in Germany [1]. This novel concept aims to improve the efficiency, with a digitalization, optimization and adaptation of the manufacturing process. Thus, the Cyber Physical Systems and the Internet of Things (IoT) [2] represent the two cornerstones of the Industry 4.0. The different agents (i.e., sensors and actuators) have to be controlled remotely in real-time, and integrated in a smart analytic workflow to enable a continuous optimization of the manufacturing process.

In this context, smart factories [3] need wireless industrial networks where small devices usually operate on battery power [4]. Since wireless transmissions are known to be lossy (particularly in complex industrial environments with multipath fading, potential interference and obstacles [5]), intelligent protocols have to be designed to improve the reliability and robustness of the network. Unfortunately, industrial automation networks require network reliability above 99.9% and guaranteed maximum delay and jitter [6].

The IEEE 802.15.4-2015 Time Slotted Channel Hopping (TSCH) standard was published in 2016 [7] to provide bounded delay and reliable communication in industrial wireless networks. It relies on a schedule of the transmissions, where two interfering transmitters are scheduled in different timeslots or over different frequency. By constructing accurately the schedule, provisioning some cells for retransmissions, and allocating consecutive cells along a path, IEEE 802.15.4 TSCH is able to provide strict guarantees even in harsh radio environments.

In order to also combat external interference, IEEE 802.15.4-TSCH employs a slow frequency hopping scheme. The 2.4 GHz sub-band is divided into smaller frequency portions, i.e., radio channels. The pseudo-randomly spread of the load over these different radio channels improves globally the reliability [8]. In particular, a packet and its retransmissions use different frequencies, making the packet losses less repetitive.

Furthermore, frequency hopping solutions often support whitelisting techniques, where only the good radio channels (i.e., high reliability, low variability) are employed. For instance, in WirelessHART, a list of bad radio channels is distributed to the network to remove these radio channels from the frequency hopping sequence [9].

Unfortunately, the radio quality is highly location sensitive. In particular, the list of the best radio channels is very link dependent [10]. Thus, a set of good radio channels (i.e., whitelist) has to be assigned per link to reduce globally the packet drops due to external interference.

MABO-TSCH [11], representing a pioneering piece of work, actually allocates multiple channel offsets to each radio link. By decoupling the scheduling process and the whitelist exploitation, MABO-TSCH keeps on maintaining a deterministic behavior. However, this approach is suboptimal when exploiting small whitelists, i.e., when many radio channels are subjected locally to external interference.

In this paper, we propose to efficiently exploit a whitelist when considering a centralized scheduling algorithm. When the controller constructs the schedule, it can both modify the whitelists and the cells to use to avoid collisions. The contributions of this paper are as follows:

1) We detail how probabilistic collisions may occur when exploiting link-dependent whitelists. In particular, two

This work was supported by the French National Research Agency (ANR) project Nano-Net under contract ANR-18-CE25-0003.

V. Kotsiou and F. Theoleyre are with the ICube Laboratory, CNRS / University of Strasbourg, 67412 Illkirch, France, (e-mail: lastname@unistra.fr).

G. Z. Papadopoulos is with the IRISA, IMT Atlantique, UBL, 35510 Cesson-Sévigné, France, (e-mail: georgios.papadopoulos@imt-atlantique.fr).

P. Chatzimisios is with the CSSN Research Lab, Department of Informatics, Alexander TEI of Thessaloniki (ATEITHE), 57400 Thessaloniki, Greece and with the Department of Computing & Informatics, Bournemouth University, BH12 5BB, United Kingdom (e-mail: peris@it.teithe.gr).

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

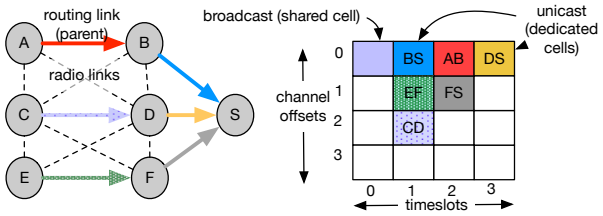


Fig. 1: TSCH schedule for a 7 nodes topology.

transmitters may create collisions for *some* of their packets, even if they use a different channel offset, because they exploit different whitelists;

2) We propose two different schemes to avoid collisions:
   a) We force all the links scheduled in the same timeslot to use the same whitelist. These whitelists are suboptimal, but no collision is created;
   b) We apply our algorithm to reorder the whitelists to forbid any possible collision.
3) We evaluate the performance of these whitelisting techniques using an experimental dataset obtained in an indoor environment i.e., FIT IoT-LAB platform.

The paper is organized as follows. In Section II, we expose the related work around TSCH and whitelisting techniques. Section III formalizes the collision problem with whitelisting. Section IV describes the two approaches we propose to avoid collisions, constructing either a common whitelist per timeslot, or re-ordering the whitelist to remove all the collisions. Section V evaluates the performance of our whitelisting scheme using an experimental dataset. Finally, Section VI concludes the paper and provides certain future directions.

II. Related Work

We present here the main concepts of IEEE 802.15.4-TSCH and how it implements a frequency hopping sequence. We also explain the different existing techniques to employ priority in the selection of the most efficient radio channels.

A. IEEE 802.15.4-TSCH

IEEE802.15.4-2015 has proposed the TSCH mode, which relies on a strict schedule of the transmissions [7]. The *slotframe* contains a fixed number of timeslots, during which at most one frame and its acknowledgment are transmitted. Each timeslot is labelled with an Absolute Sequence Number (ASN) that counts the number of timeslots since the network started. Based on the schedule, a node can decide its role (transmitter/receiver/sleeping mode) at the beginning of each timeslot.

IEEE 802.15.4-2015 TSCH implements a frequency hopping approach to combat external interference and, thus, to achieve high reliability [8]. At the beginning of a timeslot, a device verifies in the schedule if it has to stay awake. If the cell is allocated to the device, the frequency to use for transmission/reception is derived from the ASN of the timeslot and the channel offset assigned to this cell:

$$\texttt{Frequency} = F\left[(\texttt{ASN} + \texttt{Offset}) \mod \texttt{nFreq}\right] \quad (1)$$

where `nFreq` is the number of available radio channels (i.e., 16 when employing IEEE 802.15.4-compliant radios at 2.4 GHz with all channels in use), $(\mod k)$ denotes the modulo operator, and $F[]$ is a function mapping an integer to a physical frequency.

To avoid confusion, we make in this paper the following distinction:

Radio channel (or physical channel): Small portion of the radio spectrum used to transmit a packet (PHY layer).

Channel offset: An integer variable allocated to a radio link by the scheduler, translated into a radio *frequency* at the runtime right before the actual transmission (IEEE 802.15.4-TSCH layer).

Finally, IEEE 802.15.4-TSCH supports both distributed and centralized scheduling algorithms and it has received much attention [12].

B. Whitelisting

Some approaches propose to identify the radio channels which perform poorly in the whole network [9]. The nodes have to modify their pseudo-random hopping sequence with only the *good radio channels* (i.e., whitelist). In a confined environment, such as an aircraft, this strategy presents good performance [13].

The terms *blacklisting* and *whitelisting* may be used interchangeably since they correspond to dual concepts. Blacklisting consists in constructing the list of radio channels which cannot be used to transmit; oppositely, whitelisting references all the radio channels that are part of the frequency hopping sequence. While they correspond to two complementary sets, they lead exactly to the same behavior. Let $\mathcal{C}$ denote the set of all radio channels, we have:

$$Blacklist \cup Whitelist = \mathcal{C} \quad (2)$$

However, the performance may be very link-dependent in large-scale topologies [10]: the same radio channel may be *good* for some links, and *bad* for the other ones. Shi *et al.* [14] provide mechanisms to change on-the-fly the pseudo-random hopping sequence without regenerating from scratch the whole sequence each time the whitelist changes.

LABeL [15] exploits a link-based whitelisting. A pair of transmitter and receiver decides the whitelist to use autonomously. Then, at the beginning of the timeslot, each device computes the radio channel to use. More precisely, it re-computes the radio channel to use with Equation 1 inserting a pseudo-random variable derived from the ID of the transmitter, until the resulting radio channel is in the whitelist. Unfortunately, this pseudo-random approach may create collisions among links with different channel offsets, as we will highlight in Section III-A. Furthermore, detecting collisions for links with the same channel offset is also more challenging due to the pseudo-random approach.

MABO-TSCH [11] relies on a centralized scheduling algorithm that is based on a coloring scheme (one color corresponds to a channel offset). To enable local whitelists, the scheduling algorithm assigns an ordered set of channel offsets to each receiver. At the beginning of a cell assigned to a link, the receiver and the transmitter test iteratively each channel offset, to verify it corresponds to a whitelisted radio channel, with Eq. 1. The process stops when a whitelisted radio channel is computed, or the last channel offset is tested. To avoid deafness, both nodes must use the same whitelists. Thus, MABO-TSCH relies on piggybacking and sequence numbers to reliably exchange the whitelists.

Determining which radio channels to whitelist represents a key challenge. Active measurements with probes tend to provide very accurate estimations, but with a very large overhead, or additional nodes can be inserted, dedicated to monitoring [16], but with a non-negligible financial cost.

On the contrary, passive measurements can only use the data packets, while the modified pseudo-random frequency hopping creates a bias. LABeL [15] whitelists all the radio channels which do not exhibit a Packet Delivery Ratio (PDR) significantly below the k best radio channels. To continuously monitor the PDR for all the radio channels, LABeL modifies the pseudo-random hopping sequence to send data packets infrequently through the *bad* radio channels, to update the PDR value. Monitoring the PDR of broadcast packets such as Enhanced Beacons may help to infer the channel quality, but with long delays [17]. ATSCH [18] reserves two specific timeslots to measure the level of external interference, through an Energy Detection mechanism. ETSCH [19] rather exploits the idle time at the beginning of each cell to detect external interference.

A node has to determine which whitelisted radio channel it will use to transmit the packets, and how these radio channels have to be organized in the frequency hopping sequence.

III. Problem Formulation

To improve the reliability, only the best radio channels (whitelist) should be used. Improving the reliability means also less retransmissions and, thus, lower energy consumption; an idle timeslot consumes much less energy than both the reception and transmission [20]. Thus, we need to assess the quality provided by a radio channel with a certain link metric (e.g., RSSI, Packet Delivery Ratio), measured independently for each radio channel. Because the radio channel to use is derived pseudo-randomly from the channel offset, we need to wait for a sufficiently long time to obtain accurate measurements.

Whitelisting consists in identifying the best radio channels which should be used when transmitting the packets to optimize reliability. Unfortunately, whitelists are often link dependent [10]; the signal strength, and the location of the source of interference impact both the size of the whitelist as well as the set of the radio channels to include.

A. Collisions with whitelists

A collision can occur only if two transmitters use the same radio channel to send their packet during the same timeslot.

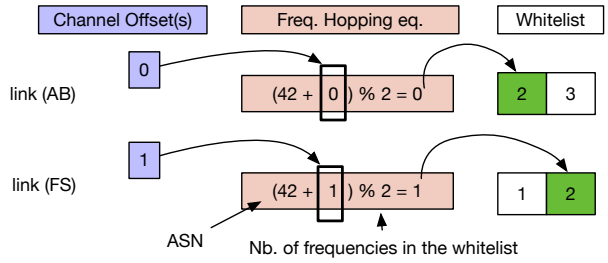


Fig. 2: Radio channel computation with whitelists (ASN=42).

TABLE I: Notation

Variable	Meaning
$\mathcal{S}$	Slotframe length (in timeslots)
$\mathcal{W}_{AB}$	whitelist of the link (A,B), of size $\lvert\mathcal{W}_{AB}\rvert$
pos_{AB}	position in $\mathcal{W}_{AB}$ of the common whitelisted radio channel
$choff_{AB}$	channel offset assigned to the link (A,B)
$GCD(x, y)$	Least Common Divisor between x and y
$LCM(x, y)$	Greatest Common Multiplier between x and y
lcm	Least Common Multiplier between $\lvert\mathcal{W}_{AB}\rvert$ and $\lvert\mathcal{W}_{FS}\rvert$
ts	timeslot number (ASN) where the collision occurs

Thus, the scheduling algorithm does not create collisions when allocating different timeslots to the different links. Allocating two different channel offsets for two transmitters during the same timeslot is also safe if we do not use whitelists. Both transmitters will derive the radio channel to use with equation 1, leading thus to different values (radio channels).

However, collisions may occur when considering whitelist, when two transmitters use the same timeslot. Indeed, the Equation 1 may use a different modulo operator. The pseudo-random sequences may overlap (i.e., same radio channel at the same position for the two sequences), and create collisions.

Let us consider the example depicted in Fig. 1, with a schedule for a topology of 7 nodes. The links (A, B) and (F, S) have been scheduled in the same timeslot, but over a different channel offset (0 and 1, respectively).

Fig. 2 illustrates the frequency mapping process for a specific timeslot (ASN=42), using Equation 1. Let us assume that (A, B) and (F, S) have the whitelists $\{2, 3\}$ and $\{1, 2\}$, respectively. The link (A, B) has to use its first whitelisted radio channel (i.e., 2) while the link (F, S) has to use the second one (also radio channel 2). In conclusion, a collision will be created in the cell with the ASN 42.

B. Formalization

Let us calculate the sequence of ASNs where collisions occur (see the example illustrated in Fig. 2). A collision is generated when two transmitters use different channel offsets during the same timeslot, but which leads to the same radio channel. Indeed, the radio channel is derived from the channel offset, according to equation 1. Thus, the equation can lead to the same value, even with two different channel offsets, when using different whitelists (different modulo operator, or different positions of each radio channel in the whitelist, cf. section III-A.)

Let us assume that the whitelists of the two radio links (A, B) and (F, S) are $\mathcal{W}_{AB}$ and $\mathcal{W}_{FS}$, respectively. A collision can occur only if both whitelists have at least one common radio channel. We denote by pos_{AB} (resp. pos_{FS}) the position of the common radio channel in $\mathcal{W}_{AB}$ (resp. $\mathcal{W}_{FS}$). A collision occurs if Equation 1 results in the same radio channel for the links (A, B) and (F, S):

$$\begin{aligned}&(ASN + choff_{AB}) \pmod{|\mathcal{W}_{AB}|} = pos_{AB}\\ \wedge\quad &(ASN + choff_{FS}) \pmod{|\mathcal{W}_{FS}|} = pos_{FS}\end{aligned}$$

$$\begin{aligned}\Leftrightarrow \exists (x,y) \in \mathbb{Z}, &\; ASN + choff_{AB} = |\mathcal{W}_{AB}| \cdot x + pos_{AB}\\ \wedge\quad &ASN + choff_{FS} = |\mathcal{W}_{FS}| \cdot y + pos_{FS}\end{aligned} \tag{3}$$

$$\Leftrightarrow \exists (x,y) \in \mathbb{Z}, \quad |\mathcal{W}_{AB}| \cdot x - |\mathcal{W}_{FS}| \cdot y = c_1 \tag{4}$$

$$\text{where:} \quad c_1 = (choff_{AB} - pos_{AB}) - (choff_{FS} - pos_{FS})$$

Equation 4 is a linear Diophantine equation [21] of the general form $ax+by=c \quad x,y \in \mathbb{Z}$. It has an infinite number of solutions if the greatest common divisor (GCD) of a and b divides c ($GCD(a,b) \mid c$). Moreover, if (x_o, y_o) is a solution, then the other solutions have the following form:

$$x = x_o + \frac{b}{d}n, \quad y = y_o - \frac{a}{d}n \quad n \in \mathbb{Z} \tag{5}$$

$$\text{where:} \quad d = GCD(a,b)$$

Consequently, Equation 4 has an infinite number of solutions if $d = GCD(|\mathcal{W}_{AB}|, -|\mathcal{W}_{FS}|)$ divides c_1. Let's assume that (x_1, y_1) are possible solution of Equation 4. All the integer solutions are then:

$$\forall n \in \mathbb{Z}, \quad x = x_1 + \frac{-|\mathcal{W}_{FS}|}{d}n, y = y_1 - \frac{|\mathcal{W}_{AB}|}{d}n$$

Let lcm be the least common multiple of $|\mathcal{W}_{AB}|$ and $|\mathcal{W}_{FS}|$. We have also $|\mathcal{W}_{AB}| \cdot |\mathcal{W}_{FS}| = LCM(|\mathcal{W}_{AB}|, |\mathcal{W}_{FS}|) \cdot GCD(|\mathcal{W}_{AB}|, |\mathcal{W}_{FS}|)$. Thus, according to Equation 3:

$$\begin{aligned}&ASN = |\mathcal{W}_{AB}| \left(x_1 - \frac{|\mathcal{W}_{FS}|}{d}n\right) + pos_{AB} - choff_{AB}\\ \Rightarrow\; &ASN = -lcm \cdot n + |\mathcal{W}_{AB}| \cdot x_1 + \\ &\qquad pos_{AB} - choff_{AB} \quad n \in \mathbb{Z}\end{aligned} \tag{6}$$

We denote by $\mathcal{S}$ the slotframe length. Since the two links use the same timeslot ts, a collision occurs if:

$$\begin{aligned}&ASN = -lcm \cdot x + |\mathcal{W}_{AB}| \cdot x_1 + pos_{AB} - choff_{AB} \quad \wedge\\ &\qquad ASN = \mathcal{S} \cdot y + ts \quad x,y \in \mathbb{Z}\end{aligned} \tag{7}$$

$$\begin{aligned}&\Rightarrow lcm \cdot x + \mathcal{S} \cdot y = c_2 \quad where\\ &c_2 = |\mathcal{W}_{AB}| \cdot x_1 + pos_{AB} - choff_{AB} - ts \quad x,y \in \mathbb{Z}\end{aligned} \tag{8}$$

Equation 8 is still a linear Diophantine equation so it has solutions iif $GCD(lcm, \mathcal{S}) \mid c_2$.So if Equation 8 has a solution (x_2, y_2) and $d = GCD(lcm, S)$, according to Equation 5, we have a collision for every timeslot with the following ASN:

$$\begin{aligned}&\mathcal{S}\left(y_2 - \frac{lcm}{d}n\right) + ts =\\ &\qquad \mathcal{S} \cdot y_2 + ts - \frac{lcm \cdot \mathcal{S}}{d}n \quad n \in \mathbb{Z}\end{aligned} \tag{9}$$

Since $ASN \geqslant 0$ we can rewrite Equation 9 as follows:

$$ASN = \mathcal{S} \cdot y_2 + ts + \frac{lcm \cdot \mathcal{S}}{d}n \quad n \in \mathbb{N}^* \tag{10}$$

Summarizing, a collision occurs every $\frac{lcm}{d}$ slotframes, therefore, the ratio of collisions is $\frac{1}{\frac{lcm}{d}} = \frac{d}{lcm}$.

In this paper, we propose mechanisms to exploit whitelists without creating these collisions. More precisely, we rearrange the whitelists when we detect collisions to avoid any inconsistent configurations.

IV. Avoiding collisions when using whitelists

Existing per-link whitelists, independent of the schedule, often generate collisions pseudo-randomly. In particular, LABeL [15] adopts a pseudo-random approach to use whitelists. However, it may generate collisions even among links which use different channel offsets. Here, we adopt rather a deterministic approach, where no collision is generated in the network. We rely on a centralized scheduling algorithm to assign the cells (timeslot and channel offset), and then to resolve collisions due to overlapping whitelists.

A. Common Whitelist per Timeslot

Since radio characteristics are variable across the wireless network, imposing the same whitelist for all radio links is suboptimal. This constraint causes an overall downgrading of network performance. Inversely, a per link whitelist may create collisions, as exposed previously. Thus, we propose here to group the radio links so that all the links in a group share the same whitelist, but different groups may have different whitelists.

Our *common whitelist per timeslot* consists in forcing all the nodes which share the same timeslot to use the same whitelist. The controller collects the statistics about the quality of each radio channel for all the nodes toward their parent. Then, it groups the nodes per timeslot, and constructs a common whitelist for each group.

When all nodes use the same whitelist, it is impossible for a collision to take place. Indeed, the whitelist has to be large enough to support all transmissions: the scheduler assigns as many channel offsets as the whitelist size for each timeslot.

Let us consider the TSCH schedule depicted in Fig. 1. Typically, the links *CD*, *EF* and *BS* must use the same whitelist to avoid collisions. Moreover, the whitelist has to contain at least 3 radio channels to support multiplexed transmissions, through 3 different channel offsets.

We construct a common whitelist as following:

1) All the nodes measure the PDR toward their parent (one measurement per radio channel). Then, this PDR is reported to the controller;

2) The controller allocates a set of timeslots for each radio link, depending on their traffic;
3) The controller then constructs a common whitelist for all radio links scheduled in the same timeslot. More precisely, it selects the k best radio channels with the highest average rank since we have to consider the fairness.
4) The controller sends then the schedule (timeslot/channel offset/whitelist to all the nodes).

B. Whitelist re-Ordering

If two links exhibit very different conditions, we would select the radio channels that perform *on average* the best. In other words, we select medium quality radio channels, leveling down the whole network performance.

We propose to enhance the previous solution by rearranging the whitelists to remove the collisions. Let us consider the example depicted in Fig. 2. If the whitelists of (A, B) and (F, S) are re-arranged into $\{3, 2\}$ and $\{1, 2\}$ respectively, a collision cannot anymore happen. Indeed, a collision occurs only if the Equation 1 results in the same integer value. However, this will never occur since the two links have different channel offsets.

In fact, the scheduler has the full knowledge of the radio topology and the link qualities. When allocating two or more radio links in the same timeslot, it has to verify that the two whitelists cannot lead to collisions.

This problem is closely related to the *University Course Timetabling Problem* (UCTP) [22], where a set of lectures have to be scheduled for a set of students, during the same timeslots. Similarly, the links correspond to the students, the channel offsets to the timeslots and the radio channels to the lectures. Furthermore, a lecture should be given in a single timeslot, for all students who have to attend it. Similarly, a radio channel must be located at the same place in the different frequency hopping sequences.

The above problem is NP-complete and the research community has proposed many algorithms such as genetic, hybrid, tabu [23] approaches. In our case, the frequency hopping sequence length and the number of links during the same timeslot are reasonably small. Thus, a greedy approach seems acceptable to produce an efficient common whitelist.

Basically, the input of our algorithm is a matrix where each row corresponds to the whitelist of a link scheduled in a given timeslot (Fig. 3). The output of the algorithm is the same matrix, with the re-ordered whitelists to avoid collisions.

For instance, let us assume three links have the whitelists depicted in Fig. 3a. These whitelists have to be re-arranged, else, the radio channel 12 which is used by different links may create collisions. Thus, we first construct the list of links associated with each radio channel (Fig. 3b). Our greedy algorithm then picks the radio channels: it first picks the radio channel 12, placed at the beginning of all whitelists. Then, the radio channel 7 is selected, used only by two links. Since the first link does not have the radio channel 7 in its whitelist, it has to search for a radio channel only whitelisted by this link (i.e., radio channel 13). Finally, the algorithm finds a re-ordered solution, without collision, as illustrated in Fig. 3c.

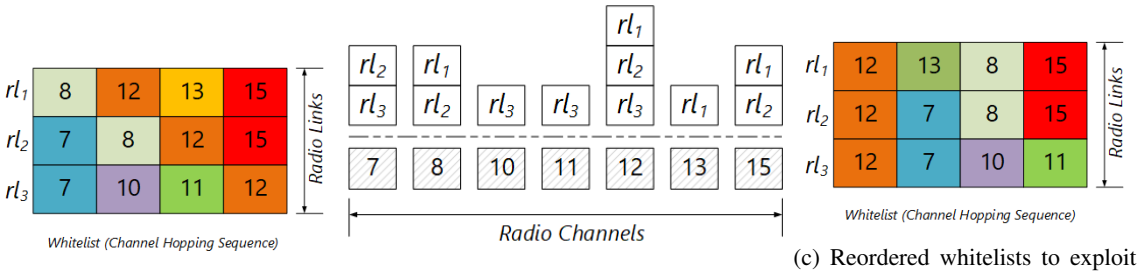
Algorithm 1: Whitelists Re-Ordering

Data:
$\mathcal{L}inks$: list of links scheduled during the considered timeslot
nch: number of radio channels in a whitelist
$\mathcal{WL}[|\mathcal{L}inks|][nch]$: matrix of the whitelists for each link
$ListofLinksPerChannel[16]$: array of lists
$Cols[nch]$: array of lists - keep track of $\mathcal{WL}_{reord}$ empty cells
$ChRank[|\mathcal{L}inks|][16]$: list of channels of each link arranged in descending order according to their quality
Result:
$\mathcal{WL}_{reord}[|\mathcal{L}inks|][nch]$: ReOrdered whitelists, initially all cells are empty($\emptyset$)

```
   // for each radio channel and each link
1  for l ∈ Links and ch ∈ [0, nch − 1] do
2      if ch ∈ WL[l] then
3          ListofLinksPerChannel[ch].add(l)
4      end
5  end
   // each column contains all links since WL_reord
      is empty
6  for l ∈ Links and ch ∈ [0, nch − 1] do
7      Cols[ch].add(l)
8  end
   // selects greedily the radio channels
9  for k ∈ [0, nch − 1] do
10     repeat
11         ch ← maxSubset(ListofLinksPerChannel, Cols[k]);
           // constructs the whitelists
12         for l ∈ Links do
               // If the link has this radio channel
                  in its whitelist, let's use it
13             if ch ∈ WL[l] then
14                 WL_reord[l][k] ← ch;
15                 Cols[k] = Cols[k] − ListofLinksPerChannel[ch];
16                 ListofLinksPerChannel[k] ← ∅;
17             end
18         end
19     until Cols[k] = ∅ or ch = −1;
20 end
   // completes the whitelists by the isolated
      radio channels
21 for l ∈ Links and k ∈ [0, nch − 1] do
       // identifies all empty cells of WL_reord
22     if WL_reord[l][k] = ∅ then
23         for j ∈ [nch, 15] do
24             ch ← ChRank[l][j];
               // if the channel has not been used
                  so far or used by another link at
                  the same column let's use it
25             if ch ∉ WL_reord or ch ∈ WL_reord[∗][k] then
26                 WL_reord[l][k] ← ch;
27             end
28         end
29     end
30 end
   // we found a valid allocation, returns the
      reordered whitelists
31 return WL_reord;
```

Our re-ordering algorithm works in the following way (algo. 1):

1) We list the radio links which contain a given radio channel in their whitelist (fig. 3b) (lines 1-5).
2) We initialize the output matrix, the reordered whitelists (lines 6-8).
3) For each column (position in the whitelists) of the output matrix (lines 9-20) :
   a) We look for a radio channel whose corresponding



(a) Whitelists of the three radio links in the same timeslot

(b) Stacks of the links which have each radio channel in their whitelist

(c) Reordered whitelists to exploit consistent (i.e. non colliding) frequency hopping sequences

Fig. 3: Reordering process of the whitelists for a group of links scheduled during the same timeslot.

list is the largest subset of the column's list (line 11). It represents the most widely whitelisted radio channel;

b) We report this radio channel in the final re-ordered whitelists, and delete the common elements of the two lists (lines 12-18).

c) We repeat the above steps, i.e., **a)**, and **b)**, until the column is full, or there is no channel that meets the criterion of step **a)** (lines 10-19).

4) If some cells are still empty in the output matrix, it means that some radio channels are missing in the whitelists. Thus, we adopt here a best-effort approach, filling with suboptimal radio channels as follows:

a) We identify the incomplete whitelists (line 22);

b) We replace the empty cells by radio channels, not used in other whitelists or used by other whitelists at the same position (lines 23-28). This way, collisions are impossible, at the price of a reduced reliability for these radio links.

V. Evaluation Setup

We assess here the performance of our common whitelisting technique with rearrangement. To obtain realistic results, we emulate a network of 60 nodes, using an experimental dataset.

A. Experimental Dataset

We rely on a dataset obtained from the FIT IoT-LAB[1] platform. This large-scale testbed mimics well an indoor, complex, environment. Other Wi-Fi or IEEE802.15.4 networks are also deployed in the building, generating external interference. More precisely, we select a large set of radio links, which forward 1 data packet every 3 seconds. The radio links are scheduled in different timeslots without collision. We store the transmission failure/success for each data packet, for 90 min[2].

We emulate a 60 nodes topology, plus a border router which collects the data packets that were randomly positioned in an area of 200 X 200 m^2. The radio transmission range of each device is 50 m. We then map each *emulated* link to a *real* link in the testbed. We must consider both the correlation among links which are geographically close, and the strength of the links (i.e., longer links tend to be statistically weaker). We proceed in this way:

1) We map the sink to a device randomly selected in the testbed;

2) We map each link, considering both the distance between the transmitter and the receiver, and the distance of the different links. When emulating a path $a \rightarrow b \rightarrow c$, we have to map these two *emulated* links ($a \rightarrow b$ and $b \rightarrow c$) to real links ($a' \rightarrow b'$ and $b'' \rightarrow c'$) in the testbed.

We select the two *real* links so that the euclidean distance of the *emulated* and *real* links are similar, and to minimize the distance between the devices b' and b''.

We then use the success/loss event of each real link for the emulated links, while preserving the correlations for geographically close links.

B. Scheduling and whitelisting algorithm

In this study, we employed the Traffic Aware Scheduling Algorithm (TASA) [24] to construct the schedule. At the beginning of each slotframe, each node generates a random number of data packets per slotframe in the range $[1, 5]$. We consider a slotframe size of 293 timeslots with 16 channel offsets, to be able to forward all data packets. Because we focus on the efficiency of the whitelisting mechanism, and not on the scheduling process, we do not provision additional cells for the retransmissions. We repeat each experiment for twenty different random network topologies.

Each whitelisting algorithm selects the k best radio channels to be included in the whitelist. We compare the following approaches:

Default (No Whitelisting): The whitelist contains all 16 available radio channels;

Global Whitelisting: Each link ranks its radio channels according to their PDR, the rank being its position in the list. Then, the global whitelist selects the k best radio channels i.e. highest average rank for all links;

MABO-TSCH: The controller assigns a fixed number of channel offset per link [11] .

LABeL: Distributed per-link independent whitelists are implemented, where collisions may arise pseudo-randomly among interfering transmitters [15].

Common Whitelist per Timeslot: The scheduler assigns the same (common) whitelist for all links scheduled in a

[1] https://www.iot-lab.info/

[2] The dataset is freely available for the research community at https://github.com/vkotsiou/grenoble-multichannel-dataset

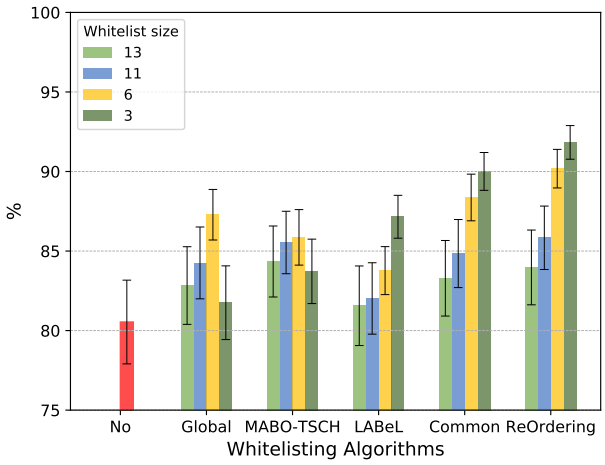


Fig. 4: Link-level Packet Delivery Ratio.

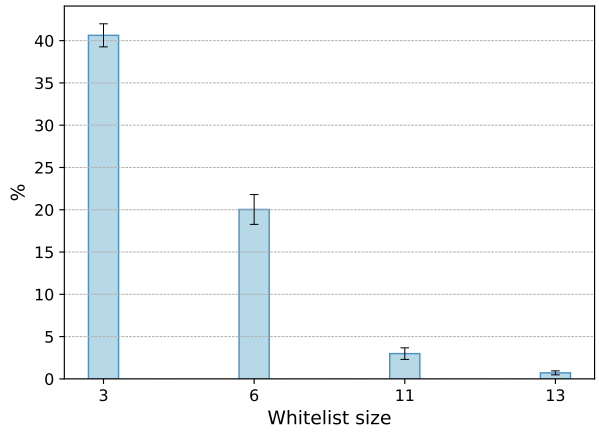


Fig. 5: Percentage of transmissions which use a non-whitelisted radio channel with MABO-TSCH.

given timeslot (Section IV-A), selecting the k radio channels which exhibit the highest average PDR;

Reordered Whitelists: We apply our re-ordering whitelist algorithm (Section IV-B);

C. Reliability

We first measure the link-level PDR, i.e., the ratio of the number of data packets delivered by the receiver and the total number of data packets transmitted by the transmitter (Fig. 4). We also measure the impact of the whitelist size: a small whitelist means that only the best radio channels are selected, to reduce the number of retransmissions. TSCH without whitelisting achieves the worst reliability: many packets use possibly radio channels with a poor PDR, which negatively impacts the global reliability. Using whatever whitelisting algorithm improves the reliability. Moreover, smaller whitelists mean that only the best radio channels are used, reducing the number of retransmissions. This reliability improvement comes with a decrease of the network capacity: the load has to be spread across a smaller number of radio channels. MABO-TSCH seems less scalable: it does not handle small whitelists: an insufficient number of channel offset is assigned, and the nodes have to use also bad radio channels (Fig. 5). A global blacklist improves slightly the reliability by removing the worst radio channels. However, some of the whitelisted channels keep on providing a lower PDR for *some* links.

We also measured for MABO-TSCH the percentage of transmissions where the assigned channels doesn't give a whitelisted channel (Fig. 5). A small whitelist increases the proportion of transmissions over non whitelisted channels ($\approx$ 40% for 3 whitelisted radio channels). This inevitably impacts negatively the reliability, the non whitelisted channels exhibiting the worst PDR. On the other hand, large whitelists abolish the use of non whitelisted channels ($\approx$ 1% for 13 whitelisted radio channels), but integrate also channels with a lower reliability.

Oppositely, our reordering strategy helps to handle small and heterogeneous whitelists. Radio links with different characteristics may even exploit different whitelists during the same timeslot without creating collisions.

Exploiting whitelists is only relevant if the radio channels condition do not change too frequently. Here, the same whitelist is used for the whole experiment, and we keep on improving the reliability. Thus, applying a centralized scheme when the environment is sufficiently stable seems reasonable, changing the schedule infrequently. Besides, distributed whitelisting schemes such as LABeL keep on generating collisions, which makes the network non-deterministic, and less suitable for critical applications.

D. Identification of Packet Drop reasons

We also identified the main reasons for packet losses:

- **Whitelisted:** The packet has been dropped, even if a whitelisted radio channel was used.
- **Collision**: The same cell has been used by an interfering radio link.
- **Probe:** The packet has been dropped because the link used a non-whitelisted radio channel (for probing).
- **Non-Whitelisted:** The radio link had to use a bad radio channel because no whitelisted radio channel was available.

LABeL adopts a non-deterministic approach, leading possibly to collisions (Fig. 6). Indeed, this strategy selects pseudo-randomly a good radio channel if the equation leads to a bad one. If this radio channel is used by an interfering link, a collision is created. Some packets are also dropped because of probes through the bad radio channels. In the same way, using a small size of global whitelist has a negative impact on the network capacity (Fig. 6). Not enough cells are available: the scheduler cannot allocate all transmissions to different cells, leading to parallel transmissions, and thus, collisions.

With MABO-TSCH, a small number of packets (1.5%) are also dropped because they act as probes (Fig. 7). However, half of the packet drops are due to the bad radio channels. The number of channel offsets assigned to MABO-TSCH becomes often insufficient when the whitelist is too small. In

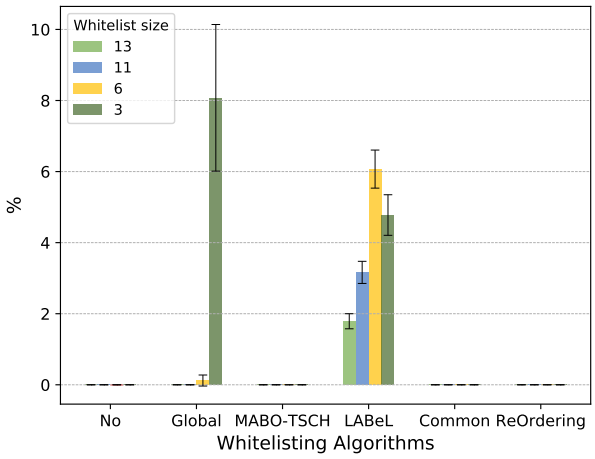


Fig. 6: Percentage of Collisions (same radio channel with different channel offsets).

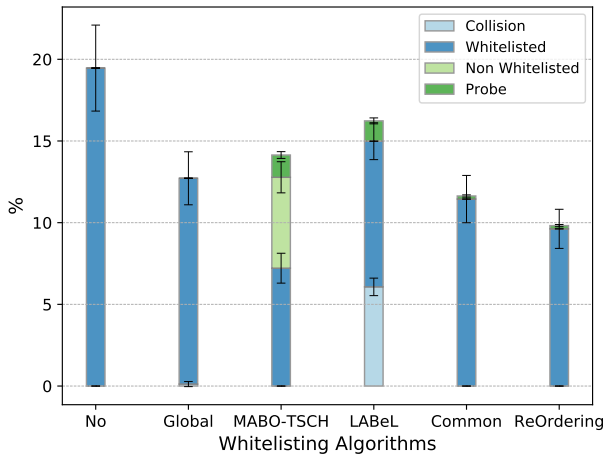


Fig. 7: Classification of the packet drops, in ratio of the total number of data packet generated in the network (whitelist size of 6 radio channels).

that cases, the packets have to be transmitted through the bad radio channels, with a negative impact on the reliability.

Forcing all the links in the same timeslot to have the same whitelist seems more efficient. With a global blacklist, all the transmissions use whitelisted channels, but some of them perform poorly for *some* links. Finally, reordering allows the different links to have different whitelists, adapted to the links characteristics. Thus, this strategy is fully deterministic, avoids collisions, and improves the per-link reliability. The remaining drops are due to the residual Packet Error Rate of the wireless links, since the transmissions use the *good* radio channels.

E. Efficiency

We aim here to quantify the gain of our whitelisting algorithms compared with a no whitelisting approach (Fig. 8). We measured here for each radio link its *gain*, i.e. ratio of the PDR provided without and with whitelisting. Formulated

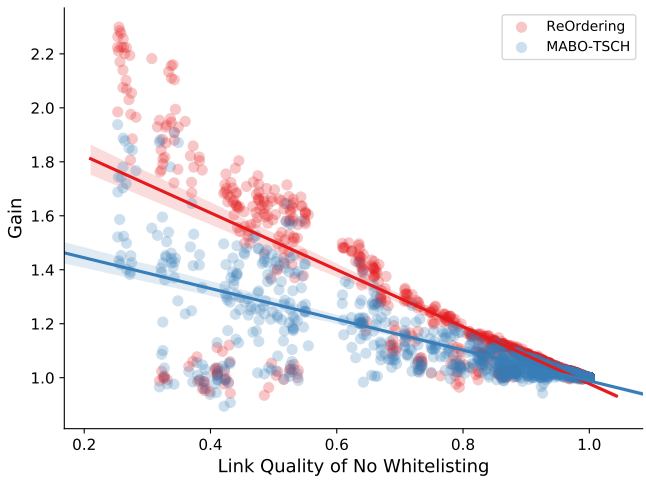


Fig. 8: Comparing per link improvement of ReOrdering and MABO-TSCH against No Whitelisting (whitelist size of 6 radio channels).

differently, the gain estimates the improvement of PDR when using a given whitelisting approach compared with the default TSCH behavior.

MABO-TSCH improves the reliability for all radio links. Since the gain is always superior than 1, this means that all links have a better Packet Delivery Ratio with MABO-TSCH. Obviously, whitelisting does not improve the reliability for perfect links: the gain is much larger for radio links with a medium or bad link quality. However, we can also notice that our reordering algorithm improves significantly the PDR compared with both TSCH without whitelisting and MABO-TSCH. More precisely, almost all the links with the reordering approach exhibit a gain larger than any link which uses MABO-TSCH. MABO-TSCH tends to use bad radio channels when the whitelist is too small, a fallback strategy which is not adopted with our reordering whitelists.

VI. Conclusions and Future Work

We here proposed a whitelisting technique able to construct a collision-free centralized schedule, even using per-link whitelists. A common whitelist for all the links scheduled in the same timeslots allows the network to reduce the usage of bad radio channels. We also proposed an algorithm to reorder the whitelists so that even two links with different whitelists can be scheduled in the same timeslot without creating collisions. Our simulation results, which used a real experimental dataset, demonstrated the relevance of this approach to reduce the packet drops.

As future work, we expect to design a centralized scheduling algorithm able to adapt the schedule according to the whitelist. Indeed, it corresponds to an optimization problem where we have to maximize the minimum reliability, by both carefully selecting which radio channels to whitelist and how to reorder them. We also plan to evaluate the accuracy of the whitelisting techniques under different environments (outdoor, different technologies or radio bands), and the ability of this technique to accommodate dynamic conditions.

REFERENCES

[1] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1 – 10, 2017.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, March 2017.

[3] Y. Luo, Y. Duan, W. Li, P. Pace, and G. Fortino, "A novel mobile and hierarchical data transmission architecture for smart factories," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3534–3546, 2018.

[4] Y. Duan, Y. Luo, W. Li, P. Pace, G. Aloi, and G. Fortino, "A collaborative task-oriented scheduling driven routing approach for industrial iot based on mobile devices," *Ad Hoc Networks*, vol. 81, pp. 86–99, 2018.

[5] L. Tang, K. C. Wang, Y. Huang, and F. Gu, "Channel characterization and link quality assessment of ieee 802.15.4-compliant radio for factory environments," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 99–110, May 2007.

[6] R. A. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, and N. Montavont, "From Best-Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4468–4480, 2018.

[7] IEEE Standard for Low-Rate Wireless Networks, "IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)," April 2016.

[8] T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *PE-WASUN*. ACM, 2009, pp. 116–123.

[9] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "Wirelesshart: Applying wireless technology in real-time industrial process control," in *IEEE RTAS*, April 2008, pp. 377–386.

[10] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "Is Local Blacklisting Relevant in Slow Channel Hopping Low-Power Wireless Networks?" in *International Conference on Communications (ICC)*. IEEE, 2017.

[11] P. H. Gomes, T. Watteyne, and B. Krishnamachari, "MABO-TSCH: Multi-hop And Blacklist-based Optimized Time Synchronized Channel Hopping," *Transactions on Emerging Telecommunications Technologies*, vol. e3223, pp. 1–20, 2017.

[12] R. Teles Hermeto, A. Gallais, and F. Theoleyre, "Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks," *Computer Communications*, pp. 84–105, Dec. 2017.

[13] M. Gürsu, M. Vilgelm, S. Zoppi, and W. Kellerer, "Reliable co-existence of 802.15.4e tsch-based wsn and wi-fi in an aircraft cabin," in *International Conference on Communications Workshops (ICC)*. IEEE, May 2016, pp. 663–668.

[14] C. F. Shih, A. E. Xhafa, and J. Zhou, "Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks," in *International Conference on Communications (ICC)*. IEEE, 2015.

[15] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LABeL: Link-based Adaptive BLacklisting Technique for 6TiSCH Wireless Industrial Networks," in *MSWiM*. ACM, 2017.

[16] R. D. Gomes, D. V. Queiroz, A. C. L. Filho, I. E. Fonseca, and M. S. Alencar, "Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes," *Ad Hoc Networks*, vol. 59, pp. 116 – 133, 2017.

[17] R. T. Hermeto, A. Gallais, K. V. Laerhoven, and F. Tholeyre, "Passive link quality estimation for accurate and stable parent selection in dense 6tisch networks," in *EWSN*. ACM, 2018, pp. 114–125.

[18] P. Du and G. Roussos, "Adaptive channel hopping for wireless sensor networks," in *International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, Oct 2011, pp. 19–23.

[19] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, "Dependable interference-aware time-slotted channel hopping for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 14, pp. 3:1–3:35, Jan. 2018.

[20] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, "A realistic energy consumption model for tsch networks," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482–489, Feb 2014.

[21] S. Johnson, "A linear diophantine problem," *Canadian Journal of Mathematics*, vol. 12, no. 3, pp. 390–398, 1960.

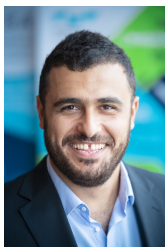
[22] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, pp. 43 – 59, 2015.

[23] A. Nanda, M. P. Pai, and A. Gole, "An algorithm to automatically generate schedule for school lectures using a heuristic approach," *International Journal of Machine Learning and Computing*, vol. 2, 2012.

[24] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, Oct 2013.



Vasileios Kotsiou is a Ph.D. student at the ICube laboratory, University of Strasbourg (France). Previously, he received his M.Sc. in Engineering of Pervasive Computing Systems from Hellenic Open University in 2014 and his B.Sc. in Computer Science from University of Crete (Heraklion) in 1996. He has already co-authored several peer-reviewed papers in international conferences and journals (e.g., IEEE SENSORS, IEEE WF-IoT). His research interests span primarily in the area of Wireless Sensor Networks (WSNs), Internet of Things (IoT), Ubiquitous Computing and Mobile Computing.



Georgios Z. Papadopoulos (MIEEE) serves as an Associate Professor at the IMT Atlantique in Rennes, France. Previously, he was a Postdoctoral Researcher at the University of Bristol. He received his Ph.D. from University of Strasbourg (France), in 2015 with honors. Dr. Papadopoulos has been involved in the organization committee of many international events (AdHoc-Now'18, IEEE CSCN'18, GIIS'18, IEEE ISCC'17). Moreover, he has been serving as Editor for Wireless Networks journal and Internet Technology Letters. He is author of more than 40 peer-reviewed publications in the area of computer communications, networks and security. His research interests include Industrial IoT, 6TiSCH, LPWAN, Wireless BMS, Smart Grid and Moving Target Defense.



Periklis Chatzimisios (MIEEE) received his B.Sc. from Alexander TEI of Thessaloniki (ATEITHE) (Greece) in 2000 and his Ph.D. from Bournemouth University (U.K.) in 2005. He serves as an Associate Professor and the Director of the Computing Systems, Security and Networks (CSSN) Research Lab in the Department of Informatics, ATEITHE. He is involved in several standardization and IEEE activities serving as a member of the Standards Program Development and the Education Services Boards for the IEEE Communication Society (ComSoc) as well as the Vice Chair of the IEEE ComSoc Technical Committees on Big Data (TCBD) and Information Infrastructure and Networking (TCIIN).



Fabrice Théoleyre is a researcher at the CNRS. After having spent 2 years in the Grenoble Informatics Laboratory (France), he is part of the ICube lab (Strasbourg, France) since 2009. He received his Ph.D. in computer science from INSA, Lyon (France) in 2006. He was a visiting scholar at the University of Waterloo (Canada) in 2006, and a visiting researcher at INRIA Sophia Antipolis (France) in 2005. He serves in about ten TPC per year, and is area editor for Ad Hoc Networks since 2018. He has been associate editor for IEEE Communications Letters and guest editor for Computer Communications and Eurasip JWCN. His research interests mainly concern distributed algorithms and experimental design for the Internet of Things.