Time Slotted Multiple-Hypothesis Interference Tracking in Wireless Networks

Julian Karoliny, *Graduate Student Member, IEEE*, Thomas Blazek, *Member, IEEE*, Fjolla Ademaj, *Member, IEEE*, Andreas Springer, *Member, IEEE*, and Hans-Peter Bernhard, *Senior Member, IEEE*

***Abstract*—Nowadays Internet of Things and Industry 4.0 devices are often connected wirelessly. Current wireless sensor network (WSN) deployments are relying in most cases on the industrial, scientific, and medical (ISM) bands without centralized resource scheduling. Thus, each device is a potential source of interference to other devices, both within its own WSN but also to devices in other collocated WSNs. If the transmission behavior of devices from other WSNs is not random, we are able to find patterns in the time domain in their channel access. This is, for example, possible for periodic channel access, which is quite common for WSNs with demanding low-power and reliability requirements. The main goal of this work is to detect multiple sources of periodic interference in time slotted signal-level measurements and estimate the time windows of future transmissions. This gives a WSN a certain understanding of the radio surrounding and can be used to adapt the transmission behavior to thus avoid collisions. For this, the multihypothesis tracking algorithm is adapted and used together with timeslot-based interference measurements on low-cost sensor nodes. The applicability of the algorithm is shown with extensive simulations and the performance is demonstrated with measurements on a time-division multiple access-based WSN built upon the Bluetooth low-energy physical layer.**

***Index Terms*—Bluetooth low energy (BLE), channel coexistence, interference, multihypothesis tracking (MHT), wireless sensor network (WSN).**

I. Introduction

With the rise of Internet of Things (IoT) and Industry 4.0, the number of wireless sensor networks (WSNs) and devices with wireless transceivers is steadily increasing. Most of these devices rely on the use of the unlicensed industrial, scientific, and medical (ISM) band. This is however limited in capacity and each additional device is a potential source of interference to other devices, both within its own WSN but also to devices in other collocated WSNs. In this context, we consider interference as access to the channel by an device external to our WSN, may it be cross-technology or intertechnology interference. Especially in the 2.4-GHz ISM band, many technologies, such as Bluetooth low energy (BLE), wireless local area network (WLAN), and ZigBee, share the same channels, and thus collisions are unavoidable.

Collisions with other devices can cause packet loss in a WSN and thus result in an increasing number of retransmissions. The collisions can also affect the interfering devices and if they also rely on retransmissions, this again increases the probability of collisions. Due to the increasing number of lost packets and retransmissions, the energy consumption of the devices will increase, which is especially problematic for low-power IoT devices. Collisions can also cause the violation of real-time constraints since the data might not reach the target in time due to a backoff time before resending again. This is a severe problem in safety-critical applications and industrial control systems.

To minimize the number of collisions, it is important that a WSN evaluates the interference behavior of other devices in close proximity and gains a certain awareness of the radio frequency surrounding. A typical approach is to measure the traffic or error rates and avoid highly occupied channels or exclude these channels from hop lists [1]. This only considers current or past interference events but cannot actively prevent future collisions. However, if the interfering device shows a certain pattern, we are able to use this information and predict future collisions. A WSN with the possibility to reschedule the own communication, e.g., certain time-division multiple access (TDMA)-based wireless protocols, can incorporate the future channel access and choose different transmission times to thus avoid collisions. For this, however, the interference has to show a systematic pattern and cannot be purely random. Although random access to the channel is a usual approach for the media access control (MAC) protocol in many WSNs, some MAC protocols show a deterministic, periodic access to the channel, which is often the case for low-power devices and sensor networks.

In this work, we present an approach that allows us to track periodic multisource interference at TDMA timeslot level

Manuscript received 2 March 2022; revised 12 July 2022 and 25 August 2022; accepted 29 August 2022. Date of publication 6 September 2022; date of current version 6 January 2023. This work was supported by the InSecTT Project (https://www.insectt.eu/), which receives from the ECSEL Joint Undertaking (JU) under Grant 876038. The JU receives support from the European Union's Horizon 2020 Research and Innovation Programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, The Netherlands, Turkey. *(Corresponding author: Julian Karoliny.)*

Julian Karoliny and Hans-Peter Bernhard are with the Research Unit Wireless Communications Department, Silicon Austria Labs GmbH, 4040 Linz, Austria, and also with the Institute for Communications Engineering and RF-Systems, Johannes Kepler University Linz, 4040 Linz, Austria (e-mail: julian.karoliny@silicon-austria.com; h.p.bernhard@ieee.org).

Thomas Blazek and Fjolla Ademaj are with the Research Unit Wireless Communications Department, Silicon Austria Labs GmbH, 4040 Linz, Austria (e-mail: thomas.blazek@silicon-austria.com; fjolla.ademaj@silicon-austria.com).

Andreas Springer is with the Institute for Communications Engineering and RF-Systems, Johannes Kepler University Linz, 4040 Linz, Austria (e-mail: andreas.springer@jku.at).

Digital Object Identifier 10.1109/JIOT.2022.3204820

and predict future channel access of devices outside the own WSN. We continuously measure the average signal level in the channel of our WSN for all TDMA timeslots, which allows us to evaluate the signal interference from external devices. The main challenge here is to assign different measured interference signals to different interferer sources and track them over time. For this, we adapted the multihypothesis tracking (MHT) algorithm, originally proposed in 1979 by Reid [2], which is typically used for radar target or visual tracking. It allows a systematic solution to the data association problem, i.e., associating uncertain measurements to known tracks, by considering all combinations of interference observations.

The contributions of this work are as follows.

1) We present a timeslot-based interference measurement solution with low-cost hardware that can be used for channel surveillance in WSNs. Based on these measurements, we build the interference tracking on real hardware.
2) For one example use case, a TDMA-based wireless network protocol, we include the measurement procedure and show the periodic interference prediction for future timeslots.
3) We implement and adapt the well-known MHT algorithm to track periodic multisource interference based on the presented TDMA network.
4) With extensive simulation, we show the limitations of the MHT for interference tracking and verify the performance of the algorithm in the multisource interference case.
5) Finally, we demonstrate the interference tracking capability of our algorithm with an example measurement.

The rest of the work is organized as follows. In Section II, we discuss and present related work in the field of WSN interference, interference estimation, and the MHT algorithm. The system model and our example use case are presented in Section III, including the measurement procedure and used protocol. Additionally, an example measurement is given. Section IV introduces the basic idea of the MHT including the adaptations to interference tracking. The performance of the MHT and its applicability are evaluated in Section V by measurements and simulations. Finally, conclusions are drawn in Section VI.

A. Notation

Scalars are written as x, while vectors and matrices are denoted as lowercase and uppercase boldface, respectively ($\mathbf{x}$ and $\mathbf{X}$). Time indices are indicated with subscript x_k. Conditional parameters are marked with |, e.g., $x_{k|k-1}$ is the parameter x at time k conditioned on the previous timestep $k-1$.

II. Related Work

This work assumes deterministic, or more specific periodic channel access from external devices for the interference tracking. Therefore, we first discuss if this assumption is valid considering the 2.4-GHz ISM band. Then, we present related work in the field of interference measurement and estimation, followed by literature references to the used MHT algorithm.

A. Interference in Wireless Sensor Networks

We consider the access to a wireless channel in a WSN to be either random or deterministic. If the channel access from a device or network is purely random, no communication pattern can be observed and used to predict future message collisions with the own transmissions. This is the case for random channel access within the own network and random access of external devices. To coexist in channels with a high percentage of random access, a common approach is the clear channel assessment (CCA) procedure [3] or simply to avoid these highly occupied channels at all. Natarajan *et al.* [4] and Guo *et al.* [5] studied the type of interference and unwanted channel access that WSNs have to face, especially in the unlicensed 2.4-GHz ISM band.

For low-power applications, devices have to stay in energy-saving mode as long as possible and avoid activating the radio unit to receive packets from another device without guaranteed transmission. A commonly used approach is to perform a part of the communication synchronized, which results in periodic channel access. Communication protocols, such as Bluetooth Mesh, Thread, or WLAN, all include these synchronized low-power amendments in newer releases, which shows a trend toward a more deterministic communication. In the following, we list the periodic behavior of some communication systems and devices in the 2.4-GHz ISM band.

1) *BLE*—connection initializer periodically polls the connected devices in a TDMA fashion [6], where the connection interval (CI) defines the time interval that regularly separates the start times of connection events. Although frequency hopping is used in BLE, for channel selection algorithm #1 the connection events will occur periodically at 37 times the CI [7].
2) *WLAN*—IEEE 802.11 sends beacon frames periodically from an access point to announce its presence and provide the SSID to the devices. They typically show a period of 102.4 ms [8]. The Wi-Fi 6 (IEEE 802.11ax) standard supports the *Target Wake Time* mechanism [9] that allows to define a certain wake-up time for devices which will result in a periodic channel access.
3) *Thread* is a low-power mesh network protocol for IoT products. The Thread 1.2 specification [10] introduced synchronized sleepy end devices (SSEDs) for enhanced low-latency and low-power features. Communication with these devices happens periodically at scheduled intervals.
4) *ZigBee* supports a *beacon-enabled* mode [11] for synchronization with dedicated timeslots for devices connected to the network coordinator or router.
5) *WirelessHART* is an industrial wireless communication protocol. Due to its TDMA structure, it will mostly show periodic channel access where the period depends on the configuration [12].
6) *Microwave oven* radiates a spectrum centered at 2.45 GHz, which can act as a severe source of

interference in the 2.4-GHz ISM band. The output cycle is tied to the 50-Hz ac input cycle [13] and therefore shows a period of 20 ms.

Even if the majority of a communication protocol shows no periodic behavior, the periodic parts may be enough to track the source of interference and apply countermeasures for the own network.

Our proposed method for interference tracking heavily relies on periodic access to the channel and the algorithm fails if the majority of the communication protocol utilizes random channel access. One example of this would be IEEE 802.11 where the access has a strong random nature depending on carrier-sense multiple access (CSMA) and random backoff times. However, even though the access can be considered random, higher network layers of the communication protocol can produce periodic access to a certain extent. Gu *et al.* [14] and Waldmann *et al.* [15] showed that specific applications, such as video streaming on Youtube or Netflix, show a distinct pattern in the channel access. Since the video data is mostly buffered and transmitted with a certain rate in chunks, we can again overall assume periodic access with a small uncertainty due to the CSMA characteristic.

B. Interference Measurement and Estimation

While the different sources of disturbances for WSNs are well studied, the topic of mitigating is mostly limited to channel switching or including the risk of collisions in an occupied channel in the requirements of the wireless link. In recent years, there is a growing need for a better understanding of channel usage, especially in the already very crowded unlicensed ISM bands. Wireless networks cannot only rely on theoretical interference, they have to observe the wireless channel and use this information to optimally adapt to the current circumstances. Ideally, no additional hardware like a spectrum analyzer is needed and the channel surveillance is part of the communication system itself. Homssi *et al.* [16] and Lee and Kim [17] demonstrated methods to improve channel awareness by evaluating the interference in different communication channels in the 2.4-GHz ISM band. A wireless network can use this information to select the best suitable channels for communication while continuously evaluating the situation and adapting to changes.

To further increase awareness, not only the link quality of a channel is important but also a classification of the interference that causes link degradation is needed. Uy *et al.* [18] demonstrated an approach to detect the presence of interference in a channel and additionally classify it as low, medium, or strong. Additionally, they provide estimates for the duration of the interference. Jin *et al.* [19] evaluated especially the interference caused by BLE and WLAN in the 2.4-GHz ISM band. They observed predictable patterns of the interference and were able to identify and distinguish interference based on received signal strength indicator (RSSI) measurements. Grimaldi *et al.* [20] and Zacharias *et al.* [21] studied the possibility to detect and classify sources of interference in WSNs. They proposed methods to measure interference directly on low-cost sensor nodes by continuously measuring the RSSI of the channel without detecting the interfering packets. Our work uses a similar measurement approach, although we focus on the tracking of individual sources of periodic interference over time. This will enable the possibility to predict future channel access. Additionally, the estimated period can provide information about the source of interference, e.g., different communication protocols will show different typical transmission periods.

With the knowledge of future channel access from other devices, a wireless network can adapt its own communication to avoid collisions. Carhacioglu *et al.* [22] showed with the example of BLE and IEEE 802.15.4 that it is possible to minimize the collisions and improve the performance of the network by rescheduling the communication of one device. They assume that the channel access of the devices is known at a central station and can thus calculate timeframes where transmission happens simultaneously on the same channel. With this information, the devices can react and reschedule accordingly. However, this approach can only mitigate devices that are known to a central coordinator and share their information. Our work aims to passively listen to the channel and find periodic interference which can be used similarly to this work for rescheduling. For this, we can not only analyze the current and past interference which is typically done in the presented literature, we have to track the current interference and need to estimate future transmissions.

To track the source of periodic interference, one key element is to estimate the transmission interval. Different sources will show a distinct period or pattern in their transmit behavior and with an accurate period estimation future collisions with the own communication can be predicted. Bernhard *et al.* [23] and Stoica *et al.* [24] presented methods to estimate the period of sparse point processes that can also be used for interference tracking. However, without additions, the presented algorithms are not able to track multiple sources of interference and fail in distinguishing sources with the same interference pattern, e.g., multiple WLAN sources.

C. Multihypothesis Tracking

The main challenge in the interference tracking task is to assign the different interference measurements to a target and distinguish it from other sources and noise. For this, our work uses the MHT algorithm, originally proposed in [2]. The MHT is well studied in the literature and there exist several implementations for various applications, especially in the field of radar and visual tracking [25], [26], [27].

Because of the ability to evaluate multiple possible hypotheses, it provides a systematic solution to the data association problem, which makes it perfect for periodic multi-interference tracking. However, with an increasing number of combinations, the computational complexity of the algorithm gets challenging. Kim *et al.* [27] and Sheng *et al.* [28] discussed improvements to the computational critical parts of the MHT algorithm. To keep the computational manageable, the MHT relies on efficient pruning, i.e., delete unlikely combinations and only keep promising tracks alive. Another approach is to group the different tracks to keep the computational

complexity manageable [29]. This is, for example, needed if we can group interference events which then show a periodic behavior in a larger scale. Ciaparrone *et al.* [30] surveyed various deep learning approaches in video multiobject tracking, including the MHT tracking. In these approaches, parts of the MHT are improved using deep learning.

III. System Model and Implementation

The idea of this work is to track periodic multisource interference in WSNs. The additional information about the radio frequency surrounding can be used to identify other devices, detect intrusion of the network, or for synchronizing to specific interference to monitor the behavior. The example use case on which we focus in this work is the prediction of future interference events in a TDMA-based wireless network protocol. Instead of dealing with current or past interference like CSMA, we are able to predict future transmissions of other devices. This enables the possibility to reschedule the own transmission to actively avoid collisions. The interference tracking and prediction can be performed by a central station without power constraints, however, CSMA has to be performed by each node individually, which might be infeasible for low-power nodes. CSMA proved to be a good mechanism to avoid collisions, however, in highly synchronized applications, it cannot be applied since devices wait for packets at specific times that cannot be delayed by CSMA.

We chose a TDMA network with high reliability and low-power requirements, where we explain how the interference measurement works and how we are able to track periodic interference. The TDMA-based protocol is managed by a central network coordinator, where each node in the network has its own communication timeslot. We show how the needed measurements can be performed directly within the TDMA framework and use the MHT algorithm to perform predictions of future interference. However, neither the following measurement procedure nor the proposed algorithm for interference tracking is limited to the presented communication protocol or use case.

The interference tracking itself consists of two parts: the interference pattern measurement and the tracking with the MHT. In the first phase, the signal level of the channel is measured and the data is collected at a central unit. In the second phase, the measured interference is evaluated to track and distinguish the different sources. Important to mention is that the interference measurement should not be performed by energy constraint low-power sensor nodes in the network, since this would dramatically increase the needed energy for these devices. The interference measurement is performed by separate nodes without energy limitations or the network coordinator. The calculation is done at the network coordinator side where the information can be distributed to the low-power nodes during the regular communication with these devices.

For the tracking, the MHT algorithm described in [27] was implemented. Since the original use case of the MHT is object tracking in visual or radar data, adaptations are made to tackle the special cases that occur in the timeslot-based interference tracking task. New measurements can be evaluated in an online fashion and predictions for future collisions within the own network can be made.

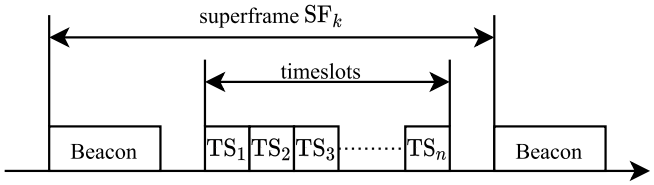


Fig. 1. Format of the EPhESOS superframe, including the beacon for synchronization and the timeslots of the individual nodes.

A. Hardware and Protocol

The TDMA protocol we use in this work is the energy- and power-efficient synchronous sensor network (EPhESOS) protocol [31]. In an EPhESOS network, each node gets a specific timeslot for the communication assigned by the network coordinator. All possible timeslots are collected in a so-called superframe, which repeats periodically with a defined superframe period. This assures a controlled and deterministic access to the channel. It also avoids collisions within the network since nodes are only allowed to communicate during their assigned timeslot. Due to the fixed beacon interval and the timeslots, the communication can be performed highly synchronized, which allows the nodes to stay in energy-saving mode most of the time and the time with an active radio unit is minimized. With the periodic beacons, the interference information can be easily distributed to the nodes and they can react accordingly. Another advantage is the availability of the EPhESOS source code, which allowed us to implement the interference measurement directly within the network protocol. This gives us the possibility to show the interference tracking performance of our algorithm with real measurements. Fig. 1 depicts the frame format of the EPhESOS protocol, including the beacon for synchronization followed by the timeslots for the different nodes.

For the measurements and practical implementations in this work, we consider the EPhESOS protocol on top of the BLE physical (PHY) layer. As a hardware platform, the Nordic NRF52840 [32] controller with an integrated transceiver unit is used.

B. Timeslot-Based Interference Measurement

To perform the required measurements, we introduce sniffer nodes, which are special sensor nodes that measure the signal level of the channel continuously and transmit the data to the network coordinator. Instead of direct message reception, only the signal strength is considered, which allows observing the interference from any communication standard. The channel access in the EPhESOS network is only controlled within the network, the interference of an external device can appear at any timeslot. However, for devices interfering with the target WSN with a periodic transmit behavior, specific patterns can be detected if the interference is observed across multiple superframes. For example, if we assume a superframe period of 100 ms, the interference of an IEEE 802.11 WLAN

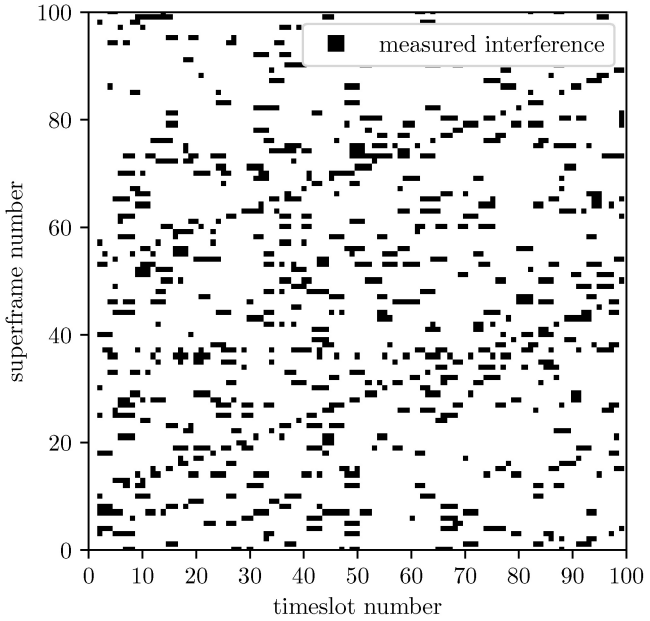


Fig. 2. Example measurement with the NRF52840 nodes in the presence of two periodic interferers. Timeslots with a higher average signal level than −90 dBm are marked black; otherwise, they are left empty.

beacon with a period of 102.4 ms will appear in each subsequent superframe a certain number of timeslots later. If now multiple subsequent superframes are considered, we are able to distinguish periodic interference from random interference. In addition, we can distinguish multiple periodic interferences from each other. We restrict the measurement resolution to the timeslot duration and measure the average signal levels of the timeslots using *energy detection*, a feature in IEEE 802.15.4 [33] and available in many commercial transceivers. It allows an automatic averaging of the signal level of the current channel for a duration of 128 μs, which is used to measure the signal level within all timeslots of the superframes.

C. Example Measurement

To demonstrate the interference measurement and visualize the patterns of periodic interference, experiments with the following setup are conducted. A TDMA network with one network coordinator and one sniffer node is placed in an office environment and the interference is measured. To show the different patterns, two interferers are placed nearby which transmit at a period of 102.4 and 92.4 ms, respectively. Both the measurement system and the interferer, use the BLE 2M PHY layer with a 2-MHz bandwidth and Gaussian frequency shift keying as modulation scheme. This example measurement was performed on BLE channel 22 with a center frequency of 2.45 GHz and a transmit power of +4 dBm.

Fig. 2 depicts the result of one such measurement, showing the interference of the timeslots over the measured subsequent superframes. If in a timeslot the measured signal level is above a −90-dBm threshold, it is marked black; otherwise, it is left empty. The first interferer transmits with a 102.4-ms period, which is larger compared to the superframe duration $t_{\mathrm{SF}} = 100$ ms. Therefore, the observation of the interference shifts to a later timeslot (i.e., higher timeslot number) with every new superframe. This results in the two lines with a positive slope one can recognize among the random pattern of interference which is shown in black rectangles in Fig. 2. The second interferer, which transmits with a period of 92.4 ms has a lower period as compared to the target WSN. Thus, the resulting interference appears several timeslots earlier in every new superframe. This pattern is difficult to detect in the measured pattern from Fig. 2. In Section V, we will see how the MHT is able to find and track also this pattern.

This and other measurements conducted during the work are published as an open-source dataset and can be found on GitHub and Zenodo under InSecTT TDMA Interference Dataset [34].

The challenge now is to find the periodic interference pattern in the measurements and distinguish them from each other and from the random interference. Additionally, we want to be able to track the interference over time and make predictions for the appearance of interference in future superframes. This task is solved with an adaptation to the MHT algorithm.

IV. Multihypothesis Interference Tracking

The general idea of the MHT is to delay the data association decision by keeping multiple hypotheses active until ambiguities are solved. In each superframe, there will be multiple interference observations, which were already shown in Fig. 2. The goal is to find the interference observations from the same periodic source and connect them from one superframe to the other. These connected sequences of observations are referred to as *track hypotheses* which are managed in a tree-like structure, the so-called *track trees*. In each superframe, the existing track hypotheses are updated with new measurements. However, if for a track hypothesis in a track tree multiple observations would fit, the track hypothesis is updated with those multiple measurements in separate branches. This of course leads to ambiguities since now multiple track hypotheses (in one track tree) share the same observations and all but one of these tracks have to be eliminated, which is referred to as *pruning*. For making a pruning decision we will define a scoring to favor the track hypotheses connecting observations from periodic interferers over random interferers. The *global hypothesis* is the best set of track hypotheses (based on the score) that are not in conflict, i.e., that do not share any interference observation at any superframe.

To demonstrate how the MHT algorithm works for interference tracking, Fig. 3 depicts the schematic for a simplified case with one periodic interferer (×) and one random interferer (○). In this example, the periodic interferer has a slightly larger period compared to the superframe duration of the TDMA protocol, hence in each subsequent superframe, the interference occurs one timeslot later.

In the initial superframe SF_1, no track hypotheses are available for an update, hence a new track hypothesis is started for the interference detected in TS_2. In the following superframe SF_2, two interference observations are available for this track hypothesis, so it is updated with both forming a track tree with two separate branches. The subsequent superframes

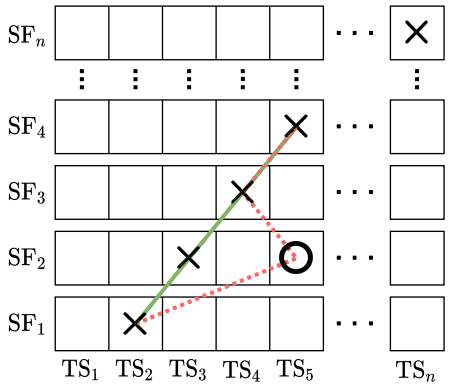


Fig. 3. Schematic of the MHT algorithm for a simplified example with one periodic interferer (×) and one random interferer (○). The two lines mark the track hypotheses that the MHT will consider.

SF$_3$ and SF$_4$ contain only one observation each and both track hypotheses are updated accordingly. The two track hypotheses in the track tree share now the same observations in SF$_1$ and SF$_3$. As mentioned before, this results in a conflict that has to be solved in the following superframes and based on the individual track scores, only one of these track hypotheses will survive. This procedure continues for all superframes.

Track hypotheses do not have to start at the first superframe and therefore the MHT algorithm uses all new detections also as the starting point of new track hypotheses. On the one hand, this ensures that every possible track is considered, but on the other hand, this dramatically increases the number of track hypotheses with every superframe. This is another reason why pruning is essential for the successful application of MHT. Problems with updates occur if there is no suitable detection for a track hypothesis available. This happens, for example, at crossing points, i.e., the observations of multiple interferers overlap in time and are not distinguishable, or if an interference detection is missing due to measurement errors. To mitigate this problem, the track hypotheses are not only updated with detections but also with predictions.

To handle the update and prediction steps of the MHT algorithm, a separate estimator for each track hypothesis is used. For this, we introduce the vector $\mathbf{x}_k$ describing the state of a track hypothesis at time k

$$\mathbf{x}_k = \begin{bmatrix} s_k \\ \dot{s}_k \end{bmatrix} \tag{1}$$

where s_k is the timeslot number occupied at time k and $\dot{s}_k$ is the corresponding velocity. In the context of interferer timeslot estimation, the velocity corresponds to the timeslot shift from one superframe to the next. For example, for a periodic interferer that transmits with the same period as our network, the velocity is zero. For the chosen state vector, the velocity is proportional to the period of the interferer and can be used to estimate the period and the timeslot position in the next superframe.

To provide a better overview of the individual steps, Fig. 4 depicts the flow diagram of the MHT algorithm including the corresponding sections where the individual blocks will be discussed. The changes of the original MHT algorithm for

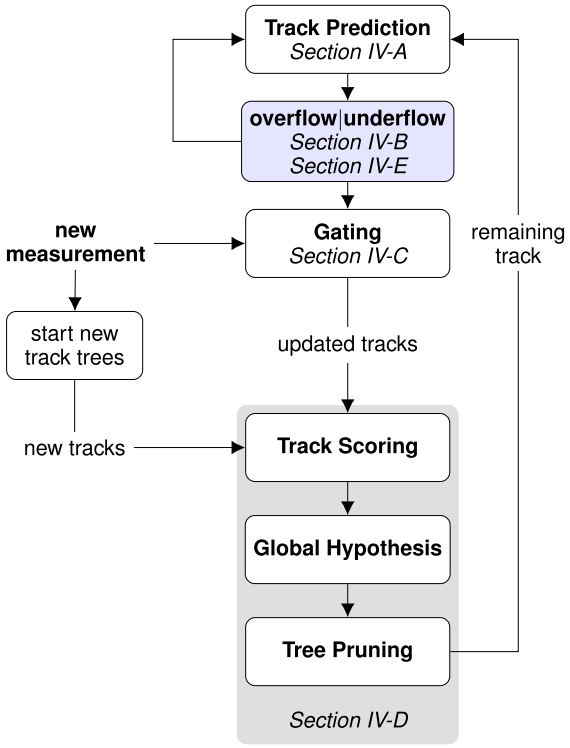


Fig. 4. Flow diagram of the multihypothesis interference tracking algorithm.

interference tracking are mainly within these blocks. One additional block for interference tracking is *overflow|underflow* marked in blue, which will be discussed in Sections IV-B and IV-E. Here, all the special exceptions are handled which will occur in the interference tracking case.

A. Kalman Filter and Model

The state estimates $\mathbf{x}_k$ of the individual track hypotheses are calculated by using a Kalman filter [35]. To represent the behavior of the periodic interference, we propose the following state transition model:

$$\mathbf{x}_k = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}} \mathbf{x}_{k-1} + \mathbf{w} \tag{2}$$

where a constant velocity, i.e., a constant period of the interferer, is assumed. The state noise process $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})$ is modeled as a multivariate normal distribution with zero mean and covariance $\mathbf{Q}$. For the tracking, only the timeslot number s_k is accessible for measurement, therefore, the observation matrix $\mathbf{H}$ is introduced to map $\mathbf{x}$ to an observation state

$$\mathbf{z}_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}_k + \mathbf{v} \tag{3}$$

with the measurement noise $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$. The noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ can be used to tune the characteristics of the Kalman filter, e.g., favor measurements over prediction.

The tracking with the Kalman filter for every new superframe can be divided into two steps, *prediction* and *update*. In the prediction step, the model proposed in (2) is used to

perform a prediction for the state $\hat{\mathbf{x}}_{k|k-1}$ and the covariance matrix $\hat{\mathbf{P}}_{k|k-1}$ of the current timestep k using

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-|1k-1} \tag{4}$$

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{F}\hat{\mathbf{P}}_{k-1|k-1}\mathbf{F}^{\top} + \mathbf{Q}. \tag{5}$$

In the update step, the measurement residual $\mathbf{y}_k$ and residual covariance matrix $\mathbf{S}_k$ are calculated with

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \tag{6}$$

$$\mathbf{S}_k = \mathbf{H}\hat{\mathbf{P}}_{k|k-1}\mathbf{H}^{\top} + \mathbf{R} \tag{7}$$

where $\mathbf{z}_k$ is the current detection. With this, the optimal Kalman gain $\mathbf{K}_k$ is calculated

$$\mathbf{K}_k = \hat{\mathbf{P}}_{k|k-1}\mathbf{H}^{\top}\mathbf{S}_k^{-1}. \tag{8}$$

If in the current superframe a valid interference observation exists for an update, the new state and covariance can be calculated using

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{y}_k \tag{9}$$

$$\hat{\mathbf{P}}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\hat{\mathbf{P}}_{k|k-1}. \tag{10}$$

If no valid observation is available, the current state and covariance are directly computed by the prediction steps (4) and (5).

B. Accounting for the Limited Number of Timeslots

A TDMA superframe consists of n_{TS} timeslots and therefore the proposed linear Kalman model in (1) is only valid for $\{s \in \mathbb{R} : 0 \leq s \leq n_{\mathrm{TS}}\}$. For a periodic interferer with $\dot{s} \neq 0$, the estimation of the Kalman filter will always exceed n_{TS} (or fall short of 0) at some point in time. Due to its periodicity, interference with $\dot{s} > 0$ will simply "wrap around" and appear in a timeslot at the beginning of the superframe. The same happens for $\dot{s} < 0$ in the other direction.

In the following, we define a falling short of the lower bound of s after an update ($s < 0$) as *underflow* and exceeding the upper bound of s after an update ($s > n_{\mathrm{TS}}$) as *overflow*, respectively. To model this behavior, the timeslot number s in (1) is corrected after the prediction step according to

$$s_{\mathrm{cor}} = \begin{cases} s, & \text{for } 0 \leq s \leq n_{\mathrm{TS}} \\ \mathrm{mod}(s, n_{\mathrm{TS}}), & \text{for } s < 0 \quad \text{(underflow)} \\ \mathrm{mod}(s, n_{\mathrm{TS}}), & \text{for } s > n_{\mathrm{TS}} \quad \text{(overflow)} \end{cases} \tag{11}$$

where $\mathrm{mod}(\cdot, \cdot)$ is the modulo operation. This assures that every Kalman prediction results in a timeslot number within the superframe structure, i.e., $0 \leq s \leq n_{\mathrm{TS}}$.

C. Gating

Updating every track hypothesis with every new interference observation leads to a high number of track hypotheses and eventually to problems in finding the global hypothesis. In fact, most interference observations will not fit the existing track hypotheses because it is either random interference or interference from another source. To keep the number of track hypotheses manageable, a simple gating mechanism is applied to decide whether a track hypothesis is updated or not. For this decision, we calculate the Mahalanobis distance d_k [36] of a track hypothesis for new interference observations by

$$d_k^2 = \mathbf{y}_k^{\top}\mathbf{S}_k^{-1}\mathbf{y}_k \tag{12}$$

where $\mathbf{y}_k$ is the measurement residual calculated by (6) and $\mathbf{S}_k$ is the residual covariance matrix calculated by (7). Only if d_k is below a defined gating threshold d_{th}, the track hypothesis will be updated with the new interference observation. The Mahalanobis distance is originally used to whiten different measurements. However, one additional advantage is that the residual covariance matrix $\mathbf{S}_k$ will naturally adapt the gating. At initialization of the Kalman filter, $\mathbf{S}_k$ will be large, allowing a higher residual. With increasing Kalman updates, $\mathbf{S}_k$ will decrease, resulting in a tighter bound and fewer false track updates. If RSSI values of the measurements are available, an additional gating with these values can be applied to improve the distinction of the individual track hypotheses. At no other point of the MHT algorithm, the RSSI values are directly needed for tracking.

D. Managing the Individual Track Hypotheses

Since the MHT algorithm allows to update every track hypothesis with every observation in a superframe, ambiguities are unavoidable. With no countermeasure, the number of track hypotheses in the individual track trees will grow exponentially, since in every new superframe new branches are generated. Therefore, a good scoring of the individual track hypotheses and the pruning of unlikely paths as soon as possible is crucial for the feasibility of the algorithm. To keep the number of track hypotheses at a manageable level, after an update the steps *track scoring*, finding the *global hypothesis*, and *tree pruning* are applied.

1) Track Scoring: The individual track hypotheses are scored using the log-likelihood ratio (LLR) between the target hypothesis and the false alarm hypothesis with the corresponding probabilities $P_{T,K}$ and $P_{F,K}$ at timestep K. The target hypothesis is defined as the marginal likelihood of the observation distribution of the Kalman filter. It is defined as Gaussian distribution with $\mathcal{N}(z_k; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k)$. As we have no information about the false hypothesis, i.e., an uninformative prior, the likelihood is chosen uniformly distributed across the n_{TS} possible timeslots with $P_{F,k} = n_{\mathrm{TS}}^{-1}$. This results in the LLR as

$$\mathrm{LLR}_K = \log\frac{P_{T,K}}{P_{F,K}} = \log\prod_{k=0}^{K}\frac{\frac{1}{\sqrt{2\pi\det\mathbf{S}_k}}\exp\left(-\frac{1}{2}\mathbf{y}_k^{\top}\mathbf{S}_k^{-1}\mathbf{y}_k\right)}{n_{\mathrm{TS}}^{-1}}. \tag{13}$$

To score the track hypotheses in an online fashion, the LLR can be calculated recursively with

$$\mathrm{LLR}_k = \mathrm{LLR}_{k-1} + \Delta\mathrm{LLR}_k \tag{14}$$

$$\Delta\mathrm{LLR}_k = \begin{cases} \log(1 - P_D), & \text{no track update} \\ \log n_{\mathrm{TS}} - \frac{1}{2}\log|2\pi\mathbf{S}_k| - \frac{1}{2}d_k^2, & \text{track update} \end{cases} \tag{15}$$

where P_D is the expected probability of detection, $\mathbf{S}_k$ is the residual covariance from (7), and d_k is the Mahalanobis distance from (12) for timestep k.

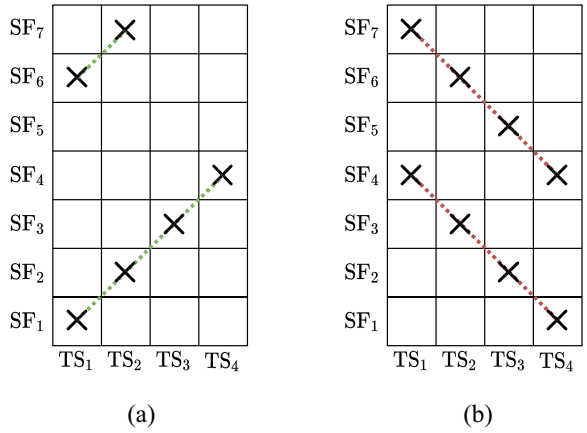


Fig. 5. Simplified example with only four timeslots to demonstrate the two special cases, overflow and underflow, that can occur in the interference tracking. (a) Overflow. (b) Underflow.

2) Global Hypothesis: After the update step and scoring, we now have to find the most likely combination of all track hypotheses given the set of track trees, the global hypothesis. This multidimensional assignment problem can be formulated as a maximum weighted independent set (MWIS) problem as stated in [27] and can be solved exactly with a standard integer linear programming (ILP) solver. This part is the most computational intense and one reason to keep the number of possible track hypotheses low.

3) Tree Pruning: The global hypothesis provides the set of track hypotheses with the highest scores which are not in conflict with each other. Track hypotheses or whole track trees that show a conflict with this solution have to be pruned to solve ambiguities. This ensures that the track trees will not grow uncontrolled and only the most likely track hypotheses survive. For this, the N-scan pruning approach described in [27] is used, which can be summarized in two steps: first, the result of the global hypothesis is used and the corresponding track trees are identified. Second, for all track hypotheses (the branches in the track trees), we go back N steps in the corresponding track trees and delete all tree branches that diverge from the optimal track hypothesis, i.e., the one with the highest score. These steps are repeated for each new superframe.

E. Adaptations to the Original MHT Algorithm

The standard MHT algorithm assumes that in each frame the objects to be tracked are only at exactly one position (one observation per superframe per interferer), though sometimes the corresponding observation is missing due to measurement errors. For the original use case—the tracking of physical objects—this assumption is valid since a real object can neither be in two different positions at the same time nor suddenly disappear. However, this assumption does not hold for the TDMA interference tracking and the two special cases, overflow and underflow, have to be considered separately. Fig. 5 depicts the differences to the initial assumption for an overflow and underflow for a simplified example with $n_{\mathrm{TS}} = 4$.

Overflows occur for periodic interferers with a higher period compared to the superframe duration. Due to this longer period, at some point, the whole superframe will be skipped as depicted for SF_5 in Fig. 5(a). This happens when the interference observation is near the right border (here TS_4 in SF_4) and the next corresponding observation has a difference larger than the superframe duration. As a result, the observation is at the beginning of SF_6 in this example. In this case, no observation of the corresponding interferer will occur in the superframe in-between and the MHT algorithm has to be adapted to skip all updates for the corresponding track hypothesis. Underflows, on the other hand, occur if an interferer has a lower period compared to the superframe duration. As depicted in Fig. 5(b), it is inevitable that at a certain point, two observations of the same periodic interference source occur in one superframe. Here, the MHT algorithm has to accept two updates and consider the possibility that the corresponding observations for one or both of these updates are missing due to measurement errors.

In addition to the updating, these special cases need also to be considered in the global hypothesis and pruning of the track trees.

F. Limitations of the MHT Algorithm

The MHT algorithm was originally proposed for multiobject tracking in video data, therefore, it processes the measurements frame by frame. Additionally, it is assumed that objects will not change the position between the frames much, therefore, the best performance for our use case is for an interferer period close to the superframe period. The frame-by-frame approach of the algorithm brings also a limitation to the trackable interference. The upper bound for the interferer period which can be tracked by our algorithm is two times the superframe period. In the case of a larger period, the interference is only observable in every second superframe and since the algorithm assumes that each measurable object is present in every frame, the MHT fails at this point. One way to overcome this is to run an additional MHT algorithm that considers only every second superframe. Here, the algorithm would simply estimate half the period which can be corrected easily afterward.

The lower bound for the interferer period which can be tracked by our algorithm is not as strictly defined as the upper bound. Below half the superframe period, interference observations will start to occur two times in the same superframe for the same periodic interferer. The frame-to-frame approach will not allow detecting interrelated interference, hence the algorithm will simply use two separate track hypotheses for the same interference. For predicting future interference events, this brings no disadvantage and the separate tracks can be merged by postprocessing, e.g., by combining suitable track hypotheses with nearly the same RSSI level.

Another limitation is the amount of interference that can be processed with the MHT algorithm. If there is too much periodic or random interference per considered superframe, the number of track hypotheses will dramatically increase and hence also the computational complexity of the MWIS problem. At this point, the MHT may still be able to predict future unoccupied timeslots, however, there might not be enough free timeslots left for the nodes to react to blocked

TABLE I
CONFIGURATION FOR SIMULATION AND MEASUREMENTS

BLE Channel	n_{TS}	t_{SF}	t_{TS}
22	100	100 ms	0.9 ms

timeslots. In this case, it may be beneficial to follow the classical approach of switching to a less occupied channel.

Additionally, the time resolution of the measurements has to be considered. Since the measurements are timeslot-based, the duration of one slot will define our measurement resolution. We will not be able to detect interference with a much smaller transmission duration compared to the timeslot duration. However, this will not affect our use case, since these very short interferences may not disturb the functionality of our network. If a higher time resolution is needed, the signal level of fractions of a timeslot can be measured and used for tracking.

V. EVALUATION OF THE INTERFERENCE TRACKING PERFORMANCE

In this section, the interference detection capability of the MHT algorithm is verified with simulations and real-world measurements. For this, the proposed TDMA frame structure for the EPhESOS network with the configuration from Table I is used, where t_{SF} is the superframe duration and t_{TS} is the timeslot length. As depicted in Fig. 1, this will leave a window of 10 ms reserved for the beacon and the additional guard times which is also the duration in each superframe where no measurements are available. These parameters are used for both, the simulations and the real-world measurements with the EPhESOS network.

To assess the performance of the MHT algorithm, we compare the estimated tracks with the observations without noise and random interference. For the simulations, the observations of the periodic interferer are directly accessible. However, for the real-world measurements, we cannot make this separation. Therefore, we performed reference measurements directly at the interferer sources and measured the timing. These additional measurements are only available in our test scenario and are not accessible for the real application since the interference is generally assumed unknown with no possibility to measure it individually.

To measure the performance, the problem is mapped to a binary classification, i.e., does interference appear in a timeslot and is it estimated correctly. As metrics the true positive rate (TPR) and true negative rate (TNR) are used, which are defined as

$$\mathrm{TPR} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{16}$$

$$\mathrm{TNR} = \frac{\text{true negative}}{\text{true negative} + \text{false positive}}. \tag{17}$$

Since we are more interested in the timeslots with detected interference and the data is highly unbalanced, i.e., we have much more free timeslots than timeslots with interference observations, the TPR is the better performance metric.

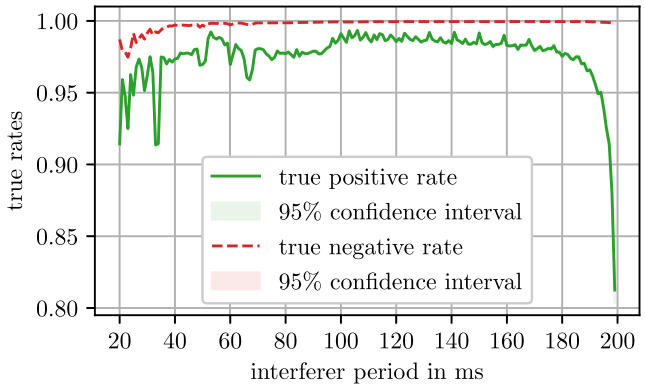


Fig. 6. True rates for all interferer periods in the defined limitations for the algorithm including the 95% confidence interval calculated over the 1000 Monte Carlo realizations.

However, for binary classification, the TPR alone is not sufficient. If, for example, the MHT algorithm always identifies interference in all timeslots, the TPR leads to a perfect result, which is obviously not the case. This is because the TPR does not consider the false positives and therefore, we also included the TNR in our evaluation. The false negative rate (FNR) and false positive rate (FPR) are not considered, since in our mapped binary classification they are just the "one-minus-version" of the true rates.

The interference simulation and tracking with the MHT algorithm are implemented in Python, which gives us the possibility to perform the evaluations on different platforms. Except for the computational effort, the simulation has no limitations like simulation length or the number of simulated interferers. A computational critical part in the MHT algorithm is solving the MWIS problem as mentioned in Section IV-D. To solve this, we used two different ILP solvers, *Gurobi* [37] and *CBC* [38]. For the comparison of the computational effort, the Python library *CProfile* is used which allows for evaluating the code execution time of the individual blocks of the MHT algorithm.

A. Performance for Different Interference Periods

The MHT works best for interference with a period similar to the superframe duration. However, in this section, we evaluate the performance for all trackable interferer periods with simulations. For this, we test the MHT within the theoretical limits defined in Section IV-F. The upper bound, i.e., two times the superframe period, results for the given configuration in 200 ms. As the lower bound is not clearly defined, i.e., tracks with a too low period can be split into multiple tracks with a higher period, we will consider interferer periods down to 20 ms, which will result in 5 track hypotheses for such an interferer. This is also the period with which a microwave oven might interfere in the 2.4-GHz frequency band.

Fig. 6 depicts the TPR and TNR for all interferer periods in the defined range. Additionally to the periodic interference, 5% of the timeslots are occupied by random interference from different sources. For the presented results 1000 Monte Carlo realizations with a random starting point of the periodic interference and noise were conducted for 1000 superframes

each. The curves represent the mean true rates of all 1000 realizations including the 95% confidence interval, which fits the mean quite well and is therefore hardly distinguishable. Within the defined limits, the interference tracking shows good results with an expected performance drop near the upper bound of 200 ms. In this region, the measured distance of interference timeslot numbers is becoming too large until, finally, the interference starts to skip every second superframe, thus making it unable for the MHT algorithm to track it anymore. For lower periods, a performance fluctuation can be observed since here multiple track hypotheses for each periodic interferer are generated, which also increases the possible combinations among the track hypotheses and the random noise. As a result, additional wrong track hypotheses are generated which will lower the TPR. The simulation shows significant performance differences for selected frequencies, especially between 20 and 60 ms. Even with an increasing number of Monte Carlo realizations, there is no averaging effect observable. For interferer periods below half the superframe duration, the MHT algorithm has to use multiple tracks for one interferer (see Section IV-F). For certain periods, multiple track solutions exist and some suboptimal solutions lead to a lower TPR. However, even for these suboptimal solutions, the TPR always stays above 0.9, which we consider a good performance. Overall, the MHT was able to detect over 95% of the timeslots with interference most of the time and at least 90% for periods under 40 ms. The TNRs curve shows that also the number of wrongly identified interference observations was good for all simulations.

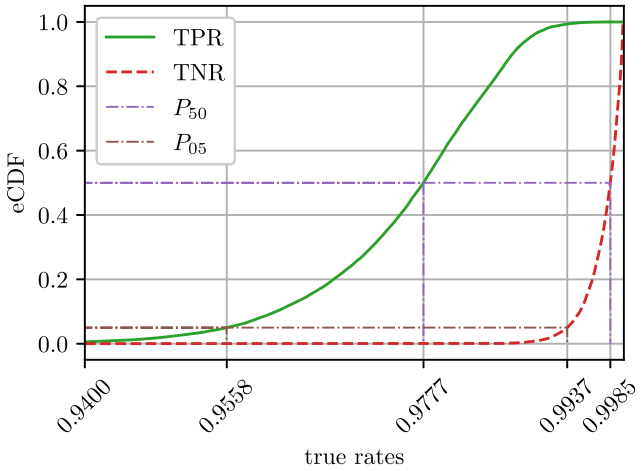


Fig. 7. eCDF of the true rates for 25 000 periodic interference realizations with a random number of interferers between 1 and 5, and random period between 50 and 150 ms.

TABLE II
TPR AND RMSE RESULTS OF THE INDIVIDUAL n_{int} FOR THE P_{50} AND P_{05} PROBABILITY MARK

	TPR		RMSE	
n_{int}	P_{50}	P_{05}	P_{50}	P_{05}
1	0.9840	0.9676	0.1620 ms	0.3595 ms
2	0.9809	0.9621	0.2368 ms	0.4368 ms
3	0.9778	0.9588	0.3060 ms	0.5020 ms
4	0.9741	0.9542	0.3634 ms	0.5680 ms
5	0.9704	0.9489	0.4144 ms	0.6255 ms

B. Performance for Multiple Interferer

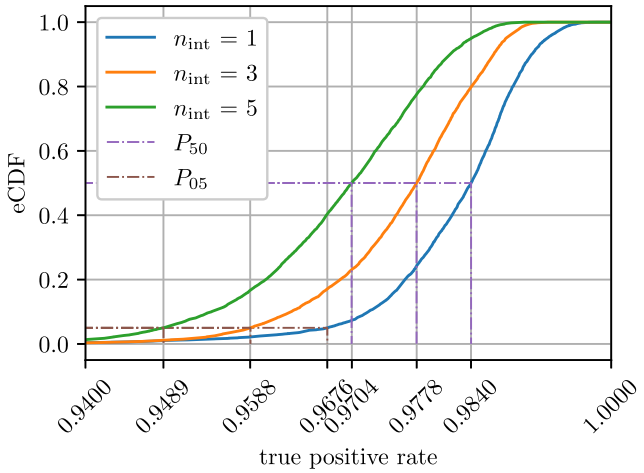
The previous evaluation only covered the single periodic interferer case. However, now we want to show the performance and capability of the MHT for multiple sources of periodic interference. For this, we again performed interference simulations and applied the MHT algorithm. We conducted 25 000 different interference scenarios, where for each we first choose a random number of periodic interferer n_{int} from 1 to 5 and then selected for each a random period from 50 to 150 ms. Like before we mapped the problem to a binary classification and calculated the TPR and TNR for the real interference compared with the tracked one. Fig. 7 shows the empirical cumulative distribution function (eCDF) of the results where we additionally marked the 5% and 50% probability, including the corresponding TPR and TNR. In the multi-interference case, only 5% of the results showed a TPR below 0.9558 and a TNR below 0.9937. This shows that even with multiple periodic interferences the MHT is capable of separating the individual sources and performing predictions.

Fig. 7 considered the combined results for all n_{int}, however, we additionally want to evaluate how the performance of the MHT algorithm depends on the number of interferers. Fig. 8(a) shows the eCDF of the TPR for 1, 3, and 5 interferers. The TNR does not change significantly for the different n_{int} and is not depicted in this figure. With increasing number of interferers the performance of the MHT drops, however, even for $n_{\text{int}} = 5$ the TPR is still 0.9489 for 95% of the simulations. Additionally to the TPR, we also calculate the root mean-squared error (RMSE) between the estimated track of the MHT and the detected interference. Fig. 8(b) depicts the empirical complementary cumulative distribution function (eCCDF) for the RMSE, again for 1, 3, and 5 interferers. Here, we can again see a performance decrease if the MHT has to estimate multiple interferers.

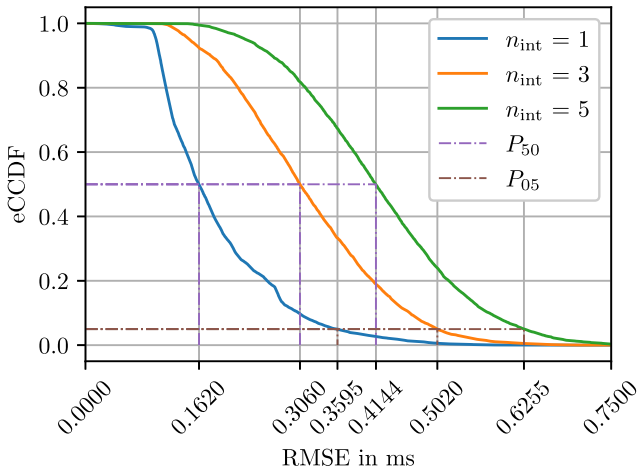
Table II summarizes the main results of Fig. 8 for all simulated n_{int}. The results in the P_{05} columns depict, for example, the performance of the MHT which can be expected for 95% of the simulations. By increasing the number of interferers, the TPR only changes minimal since here we only evaluate if the interference was detected in the right timeslot. The changes in the RMSE are higher, however, for $n_{\text{int}} = 5$ we still could achieve an RMSE of 0.6255 ms. This can be considered a good performance since the simulated measurement resolution is 0.9 ms, which is the duration of one timeslot.

C. Computational Effort

The MHT algorithm solves the data association problem by evaluating every existing track against every new measurement and therefore heavily relies on the ability to keep the number of active tracks low. The computational demanding parts of the MHT algorithm are the prediction steps for all tracks with the Kalman filter and solving the MWIS problem for finding the best set of tracks (see Section IV-D). The other parts of the algorithm are mostly for managing the individual tracks in a tree structure, however, they are still not neglectable due to



(a)



(b)

Fig. 8. Individual simulation results for 1, 3, and 5 interferer with random period between 50 and 150 ms. (a) eCDF of the TPR for the individual number of interferer. (b) eCCDF of the RMSE for the individual number of interferer.

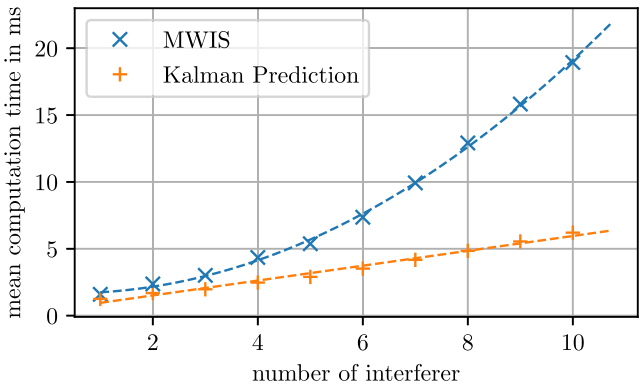


Fig. 9. Mean computational time per superframe for different number of interferers for the two most computational intense tasks in the MHT algorithm.

the large number of tracks. Cong and Hong [39] presented a detailed analysis of the computational complexity of the MHT algorithm. The computation effort depends on the number of active tracks in the last superframe and the new interference measurements in the current frame. They also state that finding the best set of tracks, in our case solving the MWIS problem, has the highest computational complexity.

We assume that the tracking will not be performed by the wireless nodes or the network coordinator, but by an edge computing device with more computational power. To evaluate the feasibility of our MHT implementation we conducted simulations similar to before with 1–10 periodic interferers with random periods. For the evaluation, we used the Python library *cProfile*, which provides the execution time and number of executions of the individual components in the MHT implementation. The absolute measured time heavily relies on the hardware and timer accuracy, however, it gives a good estimation of the computational effort, especially regarding the increasing number of interferers. We conducted the simulations on two hardware platforms, on a Windows computer with an Intel Core i7-8665U 1.90-GHz CPU with 16-GB RAM, and on a Raspberry Pi 4 with 4-GB RAM. We consider the computation feasible if the tracking can be performed faster than the superframe period, which is 100 ms in our simulations. On the Windows platform with the higher computational power, we could easily perform the computations and could satisfy our real-time constraints in most cases, including enough time to transfer the results to the network. Only for more than 8 interferers, the computations could not be performed in sufficient time for some cases. We tried both presented ILP solvers for the MWIS problem and noticed large performance differences. The commercial solver *Gurobi* outperformed the open-source solver *CBC*. The simulations on the Raspberry Pi 4 could for some realizations not satisfy the real-time constraints. Especially for 5 and more interferers, the number of realizations with too high computational time increased. The reason for this is not only the limited computational power of the Raspberry Pi 4 but also because of the not optimized ILP solver *CBC* on ARM controllers.

Our evaluation also confirmed that the two most computationally demanding tasks are the track prediction and finding the best tracks. Fig. 9 depicts the measured computational time for the Kalman predictions and for solving the MWIS problem for increasing number of interferences. Similar to the literature, we observed that the computational time of the prediction step increases linear with the number of interferers, while the computational time for finding the best tracks increases quadratically. This holds for an increasing number of periodic interferers, but also for random interferers. We fitted the depicted curves with the measurements as $0.556n_{\text{int}} + 0.40$ for the Kalman predictions and $0.1888n_{\text{int}}^2 - 0.146n_{\text{int}} + 1.70$ for the MWIS problem. The result showed that while for our simulation the computational time was feasible in most cases, in real applications, further optimizations of our MHT implementation have to be considered.

D. Measurement Results

In addition to the simulations, we prove the applicability of the MHT for interference tracking with measurements.

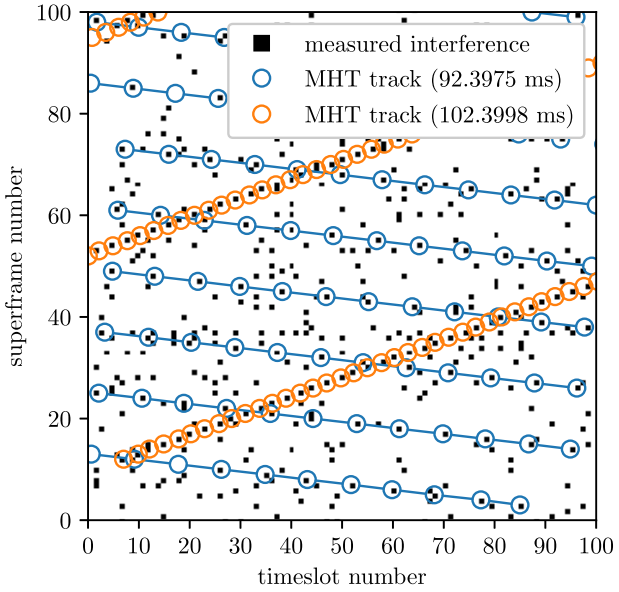


Fig. 10. Example measurement with peak detection and MHT applied. The black rectangles mark the measurements and the connected circles mark the estimated tracks.

For this, we use the measurement setup described in Section III-C and perform the MHT algorithm on the example measurement from Fig. 2. This measurement is available in the InSecTT TDMA Interference Dataset [34] under "dataset/artificial_periodic_interference1."

However, in a real-world scenario, the duration of the interference, e.g., the interference duration, can be larger than one timeslot which results in detecting the same interference over several consecutive timeslots. Aside from that, even if the interference duration is shorter compared to the timeslot duration, it can be detected in two timeslots if the interference occurs at the boundary between the two. Though the MHT can also handle this situation by constructing a separate track hypothesis for each occupied timeslot, it may be beneficial to only use one timeslot for each assumed source of interference. To find the center of the interference we apply a peak detection based on the RSSI values of the individual timeslots. A peak is defined as any timeslot whose two direct neighbors have a lower RSSI value after applying a threshold. If there are multiple peaks with the same RSSI value, the mean timeslot number is used. The number of consecutive timeslots can be saved in addition to the center for reconstructing the interference duration after applying the MHT algorithm.

Fig. 10 depicts the results of the peak detection with the subsequently applied MHT algorithm. The raw measurements after the peak detection are marked with black squares and the estimated periodic interferer positions with circles. The MHT algorithm is clearly able to track both periodic interferers (with periods of 102.4 and 92.4 ms) present. In the beginning, the algorithm needs a few superframes to find the individual track hypotheses, though once synchronized the MHT is able to perfectly follow the interference, even in the overflow and underflow cases discussed in Section IV-E. The time it takes to find the track hypotheses strongly depends on the timeslot position at the start of the algorithm. For example, track hypotheses that start too close to the superframe borders will take more time to be recognized. This effect can also be observed in Fig. 10, as the 102.4-ms interference appears at high timeslot numbers in the first few superframes considered by the MHTs and thus too close to the immeasurable area. The MHT algorithm is also able to track the 92.4-ms interference which is quite hard to distinguish from random noise in Fig. 2 by optical inspection only. Also crossing points of both interferences are no problem and the algorithm can easily separate both tracks. It can be observed that for some estimations there is no corresponding raw measurement. This shows that even if the physical detection of the sniffer nodes is missing, the MHT algorithm is able to reconstruct the tracks due to the prediction of the Kalman filter.

Comparing the MHT estimation with the reference data results in TPR $=$ 0.890 and TNR $=$ 0.998. These results show a lower TPR compared to the simulations. However, by closer inspection we observed that most errors are due to an off-by-one timeslot prediction. The output of the Kalman filter in the MHT algorithm is a float number. However, after rounding to integer numbers for the TPR calculation, the estimation appears in some cases one timeslot before or after. By counting off-by-one errors as correct, we scored TPR $=$ 0.999, which shows that the lower TPR is due to the described integer rounding. If the MHT is used for interference avoidance, this is only a minor problem, since here in addition to the estimated occupied timeslot number, a guard time has to be added to guarantee no collisions.

To give an idea of how precisely we are able to estimate the timing of the interference observations, we additionally calculated the RMSE between the MHT estimation and the corresponding measurement. This resulted in an RMSE of 0.146 ms, which shows that the MHT is able to estimate the position of the interference with a higher resolution than the 0.9-ms timeslot duration of the measurement system.

Due to the Kalman filter and the chosen model (1) and (2), in addition to the timeslot number of the interference, also an estimation of the velocity $\dot{s}$ is available. With this, we can directly calculate the corresponding interference period t_i in ms using

$$t_i = t_{\mathrm{TS}}\,\dot{s} + t_{\mathrm{SF}} \tag{18}$$

with the superframe period $t_{\mathrm{SF}} = 100$ ms and the timeslot duration $t_{\mathrm{TS}} = 0.9$ ms. Fig. 11 shows the error of the period estimation for both track hypotheses over the superframe number. Here, the estimations of the track hypotheses do not start right at the beginning and show a large error for the first estimations. This is due to the fact that the Kalman filter is initialized with $\dot{s} = 0$. However, once the interference is identified the error rapidly drops in the following superframes and reaches a steady state after a few superframes. In the steady-state, this measurement shows and RMSE of 0.024 ms which is a good result since the measurement resolution, i.e., the timeslot length, is 0.9 ms.

VI. Conclusion

In this work, we showed the applicability of the MHT algorithm for interference tracking in WSNs. With the proposed

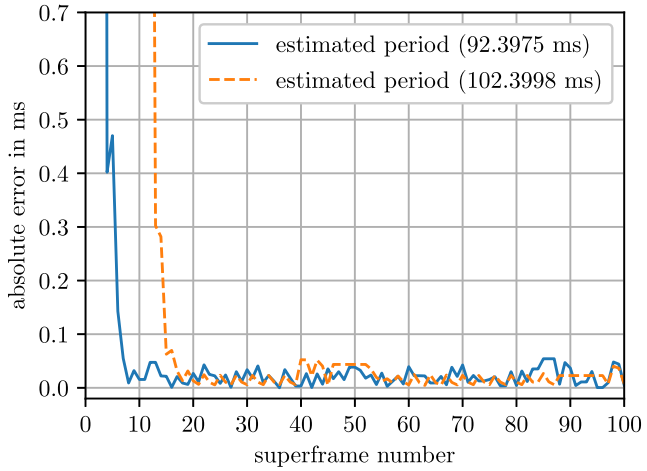


Fig. 11. Absolute error of the period estimation over the measured superframes for both tracks. The tracking of the interference starts at superframes 3 and 12, respectively.

algorithm we were able to track periodic interference and distinguish it from random interference. The estimated period can also provide additional knowledge of the interfering device since different communication protocols show different behavior on the channel access. We demonstrated the approach with measurements from a TDMA-based WSN. However, our approach is not limited to these kinds of networks. As long as continuous measurements of the channel signal level are available, the proposed algorithm can be applied. We are able to synchronize to periodic sources and are able to provide predictions about the time at which the interference will appear in future times.

If the interference prediction is combined with a TDMA-based network protocol, a central coordinator is able to reschedule the timeslots of the nodes to avoid sending at predicted interference. This will reduce the collisions with devices external to the WSNs and thus improve the coexistence in the wireless channel. Moreover, it will not only reduce the number of retransmissions in the own WSN but also for other devices.

The performance of our approach is shown with extensive simulations and real-world measurements. In both scenarios, the MHT algorithm was able to separate the periodic interference from the random interference and to score a TPR of over 0.9 and an RMSE of 0.146 ms for the measurement set. The period estimation allows to distinguish the sources of interference and the tracking allows to apply countermeasures, e.g., predicting future occupied timeslots and avoiding collisions.

ACKNOWLEDGMENT

The document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] P. Popovski, H. Yomo, and R. Prasad, "Strategies for adaptive frequency hopping in the unlicensed bands," *IEEE Wireless Commun.*, vol. 13, no. 6, pp. 60–67, Dec. 2006.

[2] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[3] W. Yuan, J.-P. M. G. Linnartz, and I. G. M. M. Niemegeers, "Adaptive CCA for IEEE 802.15.4 wireless sensor networks to mitigate interference," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2010, pp. 1–5.

[4] 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 *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, 2016, pp. 6025–6032.

[5] W. Guo, W. M. Healy, and M. Zhou, "Impacts of 2.4-GHz ISM band interference on IEEE 802.15.4 wireless sensor network reliability in buildings," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2533–2544, Sep. 2012.

[6] L. Leonardi, G. Patti, and L. Lo Bello, "Multi-hop real-time communications over Bluetooth low energy industrial wireless mesh networks," *IEEE Access*, vol. 6, pp. 26505–26519, 2018.

[7] S. Sarkar, J. Liu, and E. Jovanov, "A robust algorithm for sniffing BLE long-lived connections in real-time," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[8] S. Zacharias, T. Newe, S. O'Keeffe, and E. Lewis, "2.4 GHz IEEE 802.15.4 channel interference classification algorithm running live on a sensor node," in *Proc. IEEE SENSORS*, 2012, pp. 1–4.

[9] M. Nurchis and B. Bellalta, "Target wake time: Scheduled access in IEEE 802.11ax WLANs," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 142–150, Apr. 2019.

[10] B. Johnson, A. Lazar, and C. Woodward, "Thread 1.2 base features," London, U.K., Thread Group, White Paper, 2019. [Online]. Available: https://www.threadgroup.org/Portals/0/documents/support/Thread%201.2%20Base%20Features.pdf

[11] "ZigBee specification," ZigBee Alliance, Davis, CA, USA, ZigBee document 05-3474-21, 2015.

[12] J. Song *et al.*, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, 2008, pp. 377–386.

[13] T. Rondeau, M. D'Souza, and D. Sweeney, "Residential microwave oven interference on Bluetooth data performance," *IEEE Trans. Consum. Electron.*, vol. 50, no. 3, pp. 856–863, Aug. 2004.

[14] J. Gu, J. Wang, Z. Yu, and K. Shen, "Traffic-based side-channel attack in video streaming," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 972–985, Jun. 2019.

[15] S. Waldmann, K. Miller, and A. Wolisz, "Traffic model for HTTP-based adaptive streaming," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 683–688.

[16] B. A. Homssi, A. Al-Hourani, Z. Krusevac, and W. S. T. Rowe, "Machine learning framework for sensing and modeling interference in IoT frequency bands," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4461–4471, Mar. 2021.

[17] W. Lee and H. Kim, "Channel quality estimation for improving awareness of communication situation in the 2.4 GHz ISM band," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2002–2013, Sep. 2018.

[18] C. H. Uy, C. Bernier, and S. Charbonnier, "Design of a low complexity interference detector for LPWA networks," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, 2019, pp. 1–6.

[19] M. Jin *et al.*, "Exploiting interference fingerprints for predictable wireless concurrency," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2354–2366, Jul. 2021.

[20] S. Grimaldi, A. Mahmood, and M. Gidlund, "Real-time interference identification via supervised learning: Embedding coexistence awareness in IoT devices," *IEEE Access*, vol. 7, pp. 835–850, 2019.

[21] S. Zacharias, T. Newe, S. O'Keeffe, and E. Lewis, "A lightweight classification algorithm for external sources of interference in IEEE 802.15.4-based wireless sensor networks operating at the 2.4 GHz," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 9, 2014, Art. no. 265286.

[22] O. Carhacioglu, P. Zand, and M. Nabi, "Cooperative coexistence of BLE and time slotted channel hopping networks," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2018, pp. 1–7.

[23] H.-P. Bernhard, B. Etzlinger, and A. Springer, "Period estimation with linear complexity of sparse time varying point processes," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 767–771.

[24] P. Stoica, J. Li, and H. He, "Spectral analysis of nonuniformly sampled data: A new approach versus the periodogram," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 843–858, Mar. 2009.

[25] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.

[26] K. Yoon, Y.-M. Song, and M. Jeon, "Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views," *IET Image Process.*, vol. 12, no. 7, pp. 1175–1184, 2018.
[27] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4696–4704.
[28] H. Sheng, J. Chen, Y. Zhang, W. Ke, Z. Xiong, and J. Yu, "Iterative multiple hypothesis tracking with Tracklet-level association," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3660–3672, Dec. 2019.
[29] S. P. Coraluppi, C. Rago, and C. A. Carthel, "Multiple-hypothesis group tracking," in *Proc. IEEE Radar Conf. (RadarConf)*, 2020, pp. 1–6.
[30] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020.
[31] H. Bernhard, A. Springer, A. Berger, and P. Priller, "Life cycle of wireless sensor nodes in industrial environments," in *Proc. IEEE 13th Int. Workshop Factory Commun. Syst. (WFCS)*, 2017, pp. 1–9.
[32] *NRF52840 Product Specification v1.1*, Nordic Semicond., Trondheim, Norway, Feb. 2019.
[33] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2020 (Revision IEEE Std 802.15.4-2015), 2020.
[34] J. Karoliny, T. Blazek, F. Ademaj, H.-P. Bernhard, and A. Springer, "InSecTT TDMA interference Dataset." Distributed by Zenodo. Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6306288
[35] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," in *Proc. IEEE Int. Conf. Inf. Autom.*, 2010, pp. 1862–1866.
[36] P. C. Mahalanobis, "On the generalized distance in statistics," *Proc. Nat. Inst. Sci.*, vol. 2, no. 1, pp. 49–55, 1936.
[37] (Gurobi Optim., Beaverton, OR, USA). *Gurobi Optimizer Reference Manual*. 2022. [Online]. Available: https://www.gurobi.com
[38] J. Forrest *et al.* "coin-or/Cbc: Release releases/2.10.8." May 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6522795
[39] S. Cong and L. Hong, "Computational complexity analysis for multiple hypothesis tracking," *Math. Comput. Model.*, vol. 29, no. 9, pp. 1–16, 1999.

Julian Karoliny (Graduate Student Member, IEEE) received the B.Sc. and Dipl.Ing. (M.Sc. equivalent) degrees in mechatronic from Johannes Kepler University Linz, Linz, Austria, in 2018 and 2020, respectively, where he is currently pursuing the Dr.Tech. degree (Ph.D. equivalent) in cooperation with Silicon Austria Labs Doctoral College, under the supervision of Prof. A. Springer.

Since 2020, he has been with Silicon Austria Labs GmbH, Linz, working as a Junior Scientist. His research interests include wireless sensor networks, signal processing, and machine learning.

Thomas Blazek (Member, IEEE) received the B.Sc. degree in electrical engineering, the Dipl.Ing. degree (M.Sc. equivalent) in telecommunications, and the Dr.Tech. degree (Ph.D. equivalent) in telecommunications from the Technical University of Vienna, Vienna, Austria, in 2013, 2015, and 2019, respectively.

His Dr.Tech. dissertation was on the essential aspects of reliable vehicular communications. He is employed with Silicon Austria Labs GmbH, Linz, Austria. His research interests include vehicular network topologies and radio frequency machine learning solutions.

Fjolla Ademaj (Member, IEEE) received the M.Sc. degree (Hons.) in electrical engineering from the Faculty of Electrical and Computer Engineering, University of Prishtina, Pristina, Kosovo, in 2014, and the Dr.Techn. degree (Ph.D. equivalent) (Hons.) in telecommunications engineering from Technische Universitaet (TU) Wien, Vienna, Austria, in 2019.

From 2014 to 2019, she was a Project Assistant with the Institute of Telecommunications, TU Wien, where she co-developed the Vienna LTE-A and 5G system-level simulators. Since 2019, she has been with Silicon Austria Labs GmbH, Linz, Austria, research center working as a Postdoctoral Researcher. Her research interests include wireless communications, system-level modeling and simulations, channel modeling, and signal processing.

Andreas Springer (Member, IEEE) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Vienna, Vienna, Austria, in 1991, and the Dr.Techn. (Ph.D.) degree and the Univ.-Doz. (Habilitation) degree from Johannes Kepler University Linz (JKU), Linz, Austria, in 1996 and 2001, respectively.

From 1991 to 1996, he was with the Microelectronics Institute, JKU. In 1997, he joined the Institute for Communications and Information Engineering, JKU, where he became a Full Professor in 2005. Since July 2002, he has also been the Head of the Institute for Communications Engineering and RF-Systems (formerly, Institute for Communications and Information Engineering), JKU. In the Austrian K2 Center for Symbiotic Mechatronics, he serves as a Research Area Coordinator. Since 2017, he has been the Co-Leader of the Christian Doppler Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications. His current research interests are focused on wireless communication systems, architectures and algorithms for multiband/multimode transceivers, wireless sensor networks, and recently molecular communications. In these fields, he has published more than 280 papers in journals and at international conferences, one book, and two book chapters.

Dr. Springer was a member of the editorial board of the *International Journal of Electronics and Communications* from 2012 to 2019, and he serves as a reviewer for a number of international journals and conferences. In 2006, he was a co-recipient of the Science Prize of the German Aerospace Center. He is a member of the IEEE Microwave Theory and Techniques, the Communications, and the Vehicular Technology societies, OVE, and VDI. From 2002 to 2012, he served as the Chair of the IEEE Austrian Joint COM/MTT Chapter.

Hans-Peter Bernhard (Senior Member, IEEE) received the master's degree in electrical engineering and the Ph.D. degree in technical sciences from the Technical University of Vienna (TU Vienna), Vienna, Austria, in 1991 and 1997, respectively.

He is a Principal Scientist and the Head of Research Unit Wireless Communications, Silicon Austria Labs GmbH, Linz, Austria, and a Senior Scientist with the Institute of Communications and RF Systems, Johannes Kepler University Linz, Linz. He was an Assistant Professor with TU Vienna until 1998 and joined JKU as a Lecturer in 1999. In 2014, he started as a Senior Scientist with JKU and Silicon Austria Labs GmbH in 2018. He was a Guest Researcher with the Prague Academy of Science, Prague, Czechia, and the University of Cambridge, Cambridge, U.K.

Dr. Bernhard has organized/co-organized several special sessions at ETFA, WFCS, NOMS, and WF-IoT and serves as the General Chair for WFCS2021 and the Organizing Chair for EWSN2022. He is an Active Member of the IEEE P1451 Standard Technical Committee and a member of IEEE IES TC Industrial Informatics, Industrial Internet of Things, Industrial Cloud and Edge Computing, and IEEE IES TC Factory Automation.