SmarTiSCH: An Interference-Aware Engine for IEEE 802.15.4e-based Networks

Zihao Yu[†], Xin Na[†], Carlo Alberto Boano[‡], Yuan He[†], Xiuzhen Guo[†], Pengyu Li[†], and Meng Jin[§]
[†]School of Software & BNRist, Tsinghua University, China
[‡]Institute of Technical Informatics, Graz University of Technology, Austria
[§]Shanghai Jiao Tong University, Shanghai, China

{zh-yu,nx20,guoxz16,lpy18}@mails.tsinghua.edu.cn ; cboano@tugraz.at ; heyuan@tsinghua.edu.cn ; jinm@sjtu.edu.cn

ABSTRACT

Time-Slotted Channel Hopping (TSCH) is a popular link-layer protocol defined in the IEEE 802.15.4e standard that improves the reliability and throughput of wireless sensor networks by exploiting diversity in both time and frequency. Despite the body of literature proposing several scheduling schemes for TSCH, a gap yet to be filled is the design of an effective way to deal with internal and external interference, which are both known to strongly affect communication performance. In fact, existing works either make use of a fixed schedule (and hence cannot cope with interference), or require extra control traffic (and hence increase energy consumption). In this paper, we present SmarTiSCH, an interference-aware engine for IEEE 802.15.4e-based networks that retains the simplicity and energy-efficiency of autonomous scheduling, while increasing the awareness as well as robustness to both internal and external interference. With SmarTiSCH, the nodes in the network infer the presence of interference and react to it without the need of extra control traffic. Specifically, SmarTiSCH enables each node to infer the interference by passively observing existing data exchanges. It then lets a pair of nodes exchange information and mutually agree on a proper strategy to mitigate interference without the need of extra transmissions. We implement SmarTiSCH in Contiki-NG and evaluate its performance on a testbed of 20 off-the-shelf IEEE 802.15.4 devices based on the nRF52840. Our results show that SmarTiSCH increases the reliability of transmissions by up to 2.9 times compared to state-of-the-art approaches in the presence of interference, while even lowering the duty cycle by 54.3%.

KEYWORDS

Autonomous scheduling, Efficiency, IEEE 802.15.4e, Orchestra, Low-power wireless communication, nRF52840, Performance evaluation, Reliability, RF Interference, Time-slotted channel hopping.

1 INTRODUCTION

Time-Slotted Channel Hopping (TSCH) [14] as defined in the IEEE 802.15.4e standard is a TDMA-based MAC protocol combined with channel hopping that is explicitly designed to satisfy the requirements of industrial and mission-critical IoT applications. The design of the TSCH scheduler allocating timeslots and channels to the various links is not specified by the standard, and is left up to the system designer. Despite decades of research on how to increase the reliability of wireless communications, the scheduling of timeslots and channels in TSCH under interference still presents unique challenges: the packet delivery rate needs to be maximized for a dependable communication performance, whilst the transmission overhead in building and maintaining the schedule should be reduced to a bare minimum, so as to maximize the network lifetime.

Interference is known to strongly affect the network performance [17, 55] as it causes transmission failures between nodes and extra energy expenditure due to re-transmissions. It can be either *internal* (i.e., caused by other sensor nodes operating in the same IEEE 802.15.4 network) or *external* (i.e., caused by nearby wireless appliances employing other radio technologies that utilize the same frequencies). The robustness of communications to interference is hence an important consideration when designing schedulers.

Existing works propose many schedulers to either maintain information about interference with extra cost, or to fix the scheduling rules despite the unpredictability of interference. None of them allow to achieve awareness of interference with energy efficiency. To minimize interference, centralized or distributed schedulers [34, 46, 52] first obtain either a global view of interference at the coordinator or a local picture of interference at every node, and then adjust the timeslots or channels accordingly, which requires extra control traffic. This increases the energy expenditure of nodes and may nullify the benefits introduced to mitigate interference.

Autonomous TSCH schedulers [9, 24] achieve a more energy-efficient scheduling by interacting with the routing layer and utilizing a simple hash function to autonomously derive which timeslots to use. These schedulers randomly allocate timeslots, which inevitably produces an inefficient allocation on the links whose transmissions interfere with each other, and rely on blindly channel hopping to re-transmit packets corrupted by nearby wireless appliances (e.g., Wi-Fi devices). As a consequence, in the presence of internal or external interference, the packet delivery rate drops and the energy expenditure of nodes may drastically increases due to the necessary re-transmissions.

To increase the robustness to interference, our key design objective is to enrich the nodes with awareness of interference with minimum energy cost. This seems a *catch-22* dilemma, as autonomous scheduling schemes lack the information exchange to gain understanding of interference, whereas other approaches to gather information about interference require extra control traffic.

We solve this problem by observing that the existing data exchanges between nodes in a TSCH-based network already provide enough clues that can be accumulated and exploited to construct the understanding of interference, forming the basis for subsequent reaction. We hence propose and design SmarTiSCH, an interference-aware engine for schedulers in IEEE 802.15.4-based networks that allows the various nodes in the network to effectively deal with interference using only locally-available information. After letting each node autonomously infer the presence of internal and/or external interference, SmarTiSCH allows an efficient exchange of information such that two nodes can mutually agree on a strategy (timeslot or channel adaption) to deal with interference. It does so by embedding the interference mitigation strategy in the timing of acknowledgement packets (ACKs), and by triggering a new rendezvous in a customized control channel should the nodes need to re-transmit the packets corrupted by interference.

In summary, this paper makes the following contributions:

(1) We present the design of SmarTiSCH, an interference-aware engine for IEEE 802.15.4-based networks that enables each node to passively observe the presence of interference utilizing fine-grained information conceived in existing data exchanges. Based on this information, SmarTiSCH locally infers the presence and severity of interference.
(2) SmarTiSCH utilizes a simple but effective mechanism to allow every node pair to react to interference with no extra control traffic. It does so by re-transmitting data corrupted by interference in a customized control channel, and by enabling the information exchange by shifting the timing of ACK packets. This allows the node pair to ultimately agree on the mitigation strategy and effectively handle interference.
(3) We implement SmarTiSCH on the popular Contiki-NG operating system and evaluate its performance experimentally on a public testbed. Our results show that SmarTiSCH outperforms state-of-the-art schemes such as Orchestra in the presence of both internal and external interference.

This paper proceeds as follows. Sect. 2 provides some background on TSCH and Sect. 3 summarizes related work. We present the basics of SmarTiSCH in Sect. 4. Sect. 5 provides a high-level overview of SmarTiSCH and its main functional blocks. Sect. 6, 7, and 8 describe the inner working of these functional blocks, which enable, respectively, a passive observation of interference, the inference of its presence and severity, as well as the reaction to it. After discussing practical design issues in Sect. 9, we illustrate the results of our experimental evaluation in Sect. 10 and conclude in Sect. 11.

2 BACKGROUND

The idea behind the design of TSCH was applied to Low-power and lossy networks (LLNs) and standardized as WirelessHART [7], ISA 100.11a [44], and IEEE 802.15.4e [14]. In this section, we first provide some background on the operations of TSCH, and we then describe the two types of interference that may affect its performance.

In an IEEE 802.15.4 network employing TSCH [14], low-power nodes form a globally synchronized mesh network. Time is cut into timeslots, which are typically 10 ms long and enough for each link to complete a packet transmission with acknowledgement. Timeslots are grouped into slotframes. A slotframe is a two-dimensional time-frequency schedule table that indicates how every link in the network uses a given timeslot and channel combination to transmit or receive information. For brevity, we call the timeslot and channel combination as cell in the rest of this paper. Fig. 1 shows an exemplary slotframe of a simple network using TSCH: there are 24 cells combined by 6 timeslots (0-5) and 4 channels (0-3). The slotframe iterates in the time domain and each link is allocated a cell, indicating when to transmit/receive and which channel to use.

Source and characteristic of interference. Internal interference is an inherent problem of TSCH. In each slotframe, the links that use the same cell suffer from internal interference if they are sufficiently close to each other. In Fig. 1, the links $E \rightarrow B$ and $C \rightarrow A$ both use the cell of timeslot 3 and channel 3. Here we name the cell with at least two links allocated as **shared cell**. Assuming that the nodes of the links in shared cell are in each other's communication range, they may face packet collisions when sending data. This may either result in a complete loss of information at all nodes, or only on some of the nodes, e.g., when capture effect occurs and the stronger signal can be successfully demodulated [39]. Therefore, to handle internal interference, the allocation of cells in a slotframe to the various links in a network needs to be carefully scheduled. Though IEEE 802.15.4e standard [14] provides a back-off mechanism for nodes to re-transmit lost packets after a random number of slotframes, the links in shared cell face high link loss under high traffic load, as they contend the channel resources of the same cell.

External interference comes from the traffic of co-located wireless devices that do not belong to the IEEE 802.15.4 network, but operate in the same frequencies, e.g., Wi-Fi devices. Fig. 1 shows a scenario where external traffic exists in channel 1. In this exemplary slotframe, the links communicating in the cells shadowed in grey ($A \rightarrow C$, $B \rightarrow A$, and $B \rightarrow D$) may suffer from external interference in an **interfered channel**. TSCH uses channel-hopping to deal with external interference, i.e., it selects a different frequency over time. This allows two nodes to eventually escape interference and successfully exchange data, although at the price of a longer latency when hopping "blindly". The performance of channel hopping can be increased by first getting an estimation of the quality of all channels and by then selecting a subset of good channels to be used for communication. To this end, one can either use only good channels by means of channel whitelisting [10], or avoid the use or poorly-performing channels by means of channel blacklisting [8].

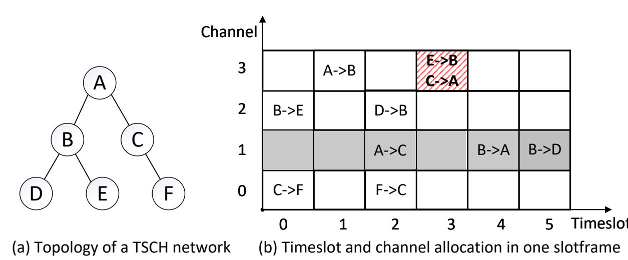

Figure 1: Exemplary slotframe built by TSCH and impact of internal and external interference. The links in the shared cells with red diagonal lines suffer from internal interference due to an overlapping cell allocation. The links using the grey cells make use of a channel that overlaps with the transmissions of a co-located Wi-Fi device and may hence suffer from external interference.

3 RELATED WORK

Several works have studied the design of low-power MAC protocols [6, 30, 42] and the problem of interference mitigation in LLNs [4, 16, 27]. We focus next on the body of works to mitigate interference on top of standardized TSCH, which can be divided into centralized, distributed, and autonomous approaches.

Centralized approaches. One approach for interference mitigation consists in collecting information at the network coordinator, who then derives a corresponding mitigation strategy [7, 31, 32, 34, 40, 44, 47, 49, 53]. To handle internal interference, the network coordinator (e.g. the root node of the RPL tree in a LLN) first collects connectivity information about all nodes in the network. It then uses an algorithm to derive a slotframe schedule without shared cells, and disseminates it back to all nodes. To mitigate external interference, these works use channel blacklisting on top of channel hopping that proves valid in industrial environments [48]. The coordinator performs channel quality estimation based on channel detection or the historical communication performance of each link. It then builds a blacklist of low-quality channels and then broadcasts it to all nodes, so that they can avoid using interfered channels [7, 44]. Unfortunately, the control traffic overhead to derive an effective timeslot allocation and channel selection for all nodes in the network in centralized approaches is extremely high, which results in a high energy expenditure.

Distributed approaches. Another approach consists in allowing nodes to locally exchange control packets with their neighbors. This allows to handle internal interference by gathering local information about network traffic and by negotiating a new cell on-the-fly [21, 28, 29, 33, 41, 52]. In a similar way, neighboring nodes can locally analyze the quality of the channels used to communicate and agree on which ones to blacklist, so to deal with external interference [13]. To enable the detection of interference on a blacklisted channel, some works also propose adaptive channel selection techniques using a combination of centralized whitelisting and distributed blacklisting [8, 10, 18, 45, 46]. Whilst all these works provide nodes with the ability to detect and mitigate interference, they incur extra energy costs, due to the additional communication overhead. Although such an overhead is lower than that of centralized approaches, it is still far from being negligible.

Autonomous approaches. The idea behind autonomous scheduling is to let each node resolve its slotframe by hashing the node addresses rather than by exchanging control packets [9, 11, 24, 25]. Autonomous approaches hence reduce the energy cost of scheduling to zero, but cannot give guarantees on the reliability of transmissions in the presence of internal or external interference. In fact, because nodes do not exchange information, they are never aware about the presence of interference at neighboring nodes. As a result, one relies exclusively on fixed schedules to allocate cells to the various links in the network, despite the unpredictability of interference. To mitigate external interference, these schedulers only make use of channels that are expected to be less interfered by Wi-Fi devices (e.g., IEEE 802.15.4 channels 15, 20, 25 and 26). Note that we cannot simply add blacklisting to autonomous scheduling to handle external interference. Even though the transmitter and the receiver can detect external interference and blacklist the channel based on local observation, they cannot blacklist the same channel

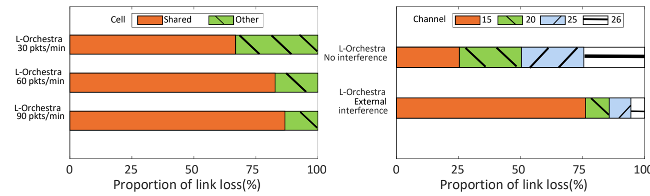

Figure 2: Link loss distribution of a state-of-the-art autonomous scheduler under interference. The lost packets are mainly transmitted in the shared cell under internal interference (left), or in a highly-congested channel in the presence of external interference (right).

at the same time due to the asymmetry of their observations. To better emphasize the limitations of state-of-the-art autonomous schedulers, we implement L-Orchestra (link-based Orchestra) [24] in a dense network of 20 nodes with interference, and find that the packet delivery rate decreases significantly, i.e., the PDR of L-Orchestra decreases by 43.0% and 44.9% under harsh internal and external interference, respectively. We track all the packets lost and show the distribution of link loss in Fig. 2. As shown in Fig. 2 (left), under high traffic load the lost packets are usually transmitted in a shared cell (in 86.7% of the cases when sending 90 packets/min). In practice, at least two links will be allocated shared cells with a probability of 97.7%. When Wi-Fi traffic is present (in Wi-Fi channel 2), the link loss in the interfered ZigBee channel (IEEE 802.15.4 channel 15) is much higher than that in other channels. As shown in Fig. 2 (right), the proportion of link loss in channel 15 is 76.19%. More details about the employed experimental setup and obtained results can be found in Sect. 10.

Summary. The main obstacle of existing schedulers to mitigate interference is to gain awareness of interference with minimum energy expenditure. Centralized or distributed schedulers collect information about interference based on extra channel detection and packet exchange, while autonomous schedulers resolve the scheduling with minimum cost, but are exposed to unpredictable interference in the network. Experimental evidence shows that the reliability of a state-of-the-art autonomous scheduler sharply declines under interference, with the link loss being mainly caused by transmissions in the same shared cell or in an interfered channel. This scheduler hence requires a method to smartly adapt the schedule by changing the cell or by avoiding interfered channels.

4 SmarTiSCH: BASICS

We first present the challenge to obtain and utilize awareness of interference for schedule adaption, then show the basics of SmarTiSCH that handle this challenge and form the basis for the engine to mitigate interference.

Challenges. While all the nodes are able to observe the interference by recording link loss, it is hard to make a node pair utilize the awareness for autonomous schedule adaption (to change the cell or to blacklist interfered channels). On the one hand, the node pair gets asymmetric picture of interference and needs to find a track to exchange information without extra packets. On the other hand, this track should be very reliable for consistent adaption even under interference. If not, the asynchronous schedule adaption of the node

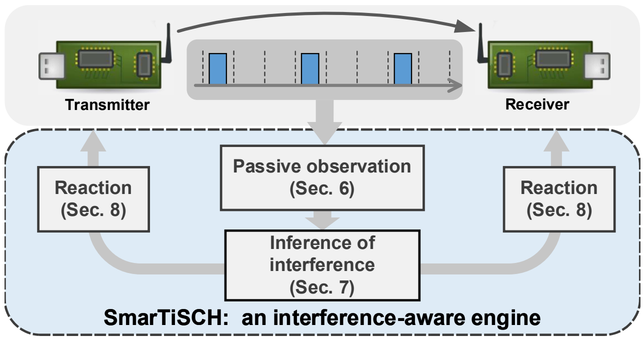

Figure 3: Design overview of SmarTiSCH. SmarTiSCH is an interference-aware engine which helps each link in the network to collect information about interference from existing data exchanges and then adapts the data schedule to mitigate the presence of interference.

pair will have dire consequences. For example, if the transmitter changes the cell but the receiver does not, then the transmissions of the node pair always fail due to cell mismatch.

Channel splitting. SmarTiSCH logically splits all available channels into data and control channels, and the latter provide a reliable way to utilize the awareness of interference. Data channels are used for normal communications, whereas control channels are used for network formation (e.g., to broadcast enhanced beacons) and re-transmission of lost packets (packets with no ACK), where the information about interference is smartly embedded. In control channels, both internal and external interference are minimized. Nodes do not use control channel to directly communicate, so the packets in control channels are much fewer. SmarTiSCH also configures the control channel to reduce external interference. In one of the configurations used to evaluate SmarTiSCH in Sect. 10, for example, we select channels 15, 20, 25, and 26 (i.e., those that do not overlap with the main Wi-Fi channels) as control channels. We detail the channel selection in Sect. 9.2.

Schedule initialization. SmarTiSCH also reduces the interference in control channels when initializing the schedule during the network formation. SmarTiSCH uses RPL for network formation[1] and lets each node maintain a link with its parent and all its children. To initialize the scheduling for each link, SmarTiSCH first derives the cell allocation following the principles of autonomous scheduling schemes. For each directional link in the network, we denote the transmitter's address and receiver's address as $ID(S)$ and $ID(R)$, respectively. Each link is given a data cell, i.e., a timeslot (TS) and a frequency offset (FO) in the slotframe with SL timeslots and N_D data channels. SmarTiSCH also allocates each link a control cell in N_C control channels denoted by control timeslot (CTS) and control frequency offset (CFO). The allocation of these cells for each link follows these equations:

$$\begin{aligned} TS &= mod(Hash(ID(S) + \alpha_1 ID(R)), SL) \\ FO &= mod(Hash(\alpha_1 ID(R)), N_D) \\ CTS &= mod(Hash(ID(S) + \alpha_2 ID(R)), SL) \\ CFO &= mod(Hash(\alpha_2 ID(R)), N_C) \end{aligned}$$

Here, the coefficient α_1 and α_2 is used to differentiate data and control cells ($\alpha_1 = 256$ and $\alpha_2 = 255$ in our implementation). When two links are allocated the same data cells, it is very likely that they will be allocated different control cells. This allows SmarTiSCH to effectively mitigate internal interference in the control channel, as one link will set up a new rendezvous in the control cell and promptly deal with the packet loss caused by simultaneous transmissions in the data cell shared by another link.

5 SmarTiSCH: DESIGN OVERVIEW

SmarTiSCH is an interference-aware engine for IEEE 802.15.4e networks based on TSCH that retains the simplicity and energy efficiency of autonomous scheduling, while increasing the awareness as well as robustness to both internal and external interference. Specifically, SmarTiSCH utilizes the existing data exchanges between the nodes to observe and react to interference.

Fig. 3 sketches the key modules of SmarTiSCH and illustrates how they operate on a link-level. After forming the network and initializing the schedule of each link, SmarTiSCH allows each node to infer the presence of interference and adapt the schedule accordingly at runtime, so to increase the reliability of communications. This is achieved in three steps, each of which is linked to a separate module in SmarTiSCH. First, each node efficiently collects fine-grained information about interference based on the exchanged data thanks to the *passive observation* module, which is detailed in Sect. 6. This module provides information to the *inference of interference* module, which then analyzes whether the link experiences internal or external interference and keeps track of the severity of such interference, as detailed in Sect. 7. In case interference is present on a link and affects its performance severely, this module triggers the *reaction to interference* module, in which the link decides a proper strategy to mitigate interference as detailed in Sect. 8. Specifically, the receiver re-transmits data in the control channel, and embeds information in the timing of the ACKs to inform the transmitter about the chosen strategy to handle interference.

6 PASSIVE OBSERVATION

We describe next the *passive observation* module of SmarTiSCH, which makes each node locally collect information about interference out of the existing data exchanges. Our key observation is that each node can implicitly obtain knowledge about the presence of interference by recording the outcome (status) of ongoing transmissions over time. For example, nodes receiving a valid packet, but destined to another recipient, can immediately notice the presence of internal interference. In SmarTiSCH we hence let every node passively observe interference by recording the status of each transmission, where the status is related to the presence or absence of internal and external interference. Note that the status is logged individually by the transmitting and receiving node, and it strongly depends on whether the transmitter has data to send in the timeslot.

[1] RPL is the IPv6 routing protocol for LLNs standardized in 2012 [51]. For more information on RPL, please refer to [22].

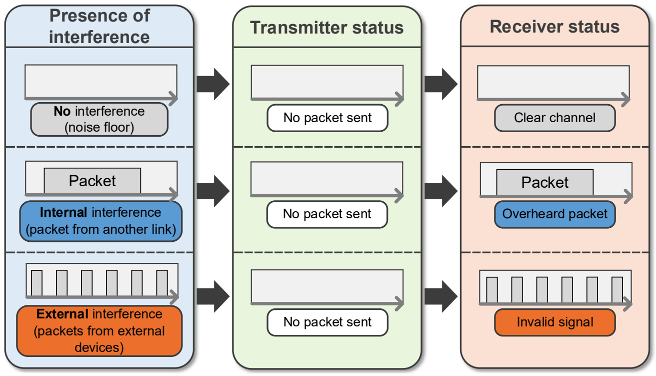

Figure 4: Status in the absence of data traffic. When the transmitter has no data to send, it sleeps in the allocated timeslot. The receiver, instead, listens to the channel and collects information about the presence of interference.

Status in the absence of data traffic. Fig. 4 shows the possible status of transmitter and receiver in case the former has no data to send. In TSCH, a transmitter with no data to send sleeps for the entire duration of the allocated timeslot, whereas the receiver still keep its radio on to check for packets. Therefore, the transmitter obtains no information about the presence of interference. The receiver, instead, can get a different status depending on the interference conditions. Specifically, the receiver senses a clear channel in the absence of interference; it senses an invalid signal (e.g., by detecting a high energy but failing to decode a packet) under external interference, and overhears a packet destined to another receiver in the presence of internal interference.

Status in the presence of data traffic. Fig. 5 shows the possible status of transmitter and receiver in case the former has data to send[2]. We distinguish between three conditions: (i) absence of interference, (ii) presence of external interference, and (iii) presence of internal interference.

Absence of interference. If interference does not occur during a transmission, the receiver should correctly receive a packet, and the transmitter should correctly receive the corresponding ACK.

Internal interference. If internal interference is present, packet reception is impaired due to the simultaneous transmissions of other nodes in the network in the same cell. In case the simultaneous transmissions are received by a node with a similar signal power, this results in a collision at the receiver (invalid signal), which will therefore not transmit an ACK. In case one of the packets is received with a much higher signal power (e.g., with a difference higher than the radio's co-channel rejection ratio, which is typically about 3 dB), instead, capture effect occurs [39]. This means that the receiver can successfully demodulate the stronger of the two signals, correctly receiving the packet. Depending on whether the received packet is from the expected node or not, a receiver will either experience no interference (and transmit the ACK back), or will overhear a packet destined to another recipient (and not transmit any ACK).

External interference. If external interference is present, RF activities may occupy the channel at any time, although they typically occur

[2]Note that Fig. 5 does not embed all the possible combinations described in the text.

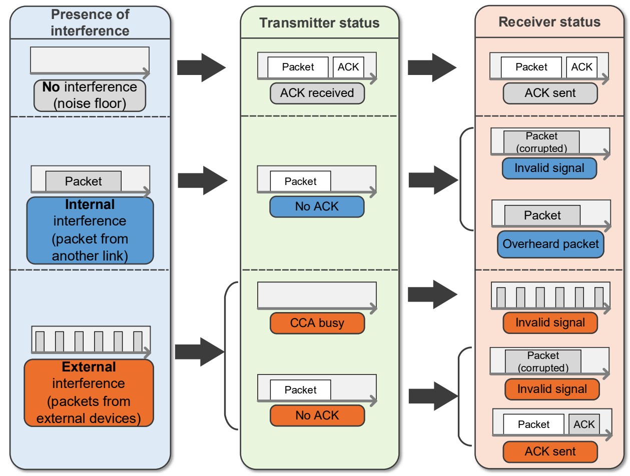

Figure 5: Status in the presence of data traffic. The presence of internal and external interference may affect the transmission and reception of the data packet and its acknowledgement in different ways.

in bursts [3]. In such a case, the transmitter may sense a busy channel when performing a CCA check prior to the transmission of a packet. Alternatively, the transmitter detects a clear channel and transmits a packet, which may not be received correctly at the receiver. Consequently, the receiver does not acknowledge the packet reception, as it senses activity in the channel but cannot successfully parse a packet. It may also happen that the transmitter transmits a packet, which is correctly decoded by its intended receiver, but the ACK packet responded by the receiver is lost.

Collection of additional information. In addition to the status of transmitter and receiver at each timeslot, SmarTiSCH also makes sure that a receiver node records duplicated packets (i.e., whether a message has already been received before), as this is a hint that the transmitter did not receive an earlier acknowledgement. Furthermore, SmarTiSCH also lets each receiver proactively sense the channel by monitoring the received signal strength (RSS) [15, 19, 43, 54]. As dedicating an entire timeslot for RSS sampling would drain the device battery, SmarTiSCH lets instead a receiver device seamlessly collect RSS samples (up to eight in our implementation) in the channel during reception[3]. That is, when the receiver fails to detect a packet, it still collects RSS information that can be used to infer the presence of interference, as discussed in the next section.

7 INFERENCE OF INTERFERENCE

We now describe the *inference of interference* module of SmarTiSCH, which allows to infer the presence of interference based on the information collected by the passive observation module.

7.1 Inference from Short-term Observations

SmarTiSCH makes use of a receiver-dominant design principle to infer the presence of interference at each transmission, i.e., the receiver is in charge of short-term observations. This principle is

[3]Note that the collection of RSS samples is orthogonal to the packet reception and does not incur an additional energy expenditure.

Table 1: Relationship between the transmission/reception status and interference

		Transmitter status			
		No traffic	CCA busy	ACK received	No ACK
Receiver status	Clear channel	No internal interference No external interference	N/A	N/A	N/A
	Overheard packets	Internal interference No external interference	Internal interference and External interference	N/A	N/A
	Packet received & ACK sent	N/A	N/A	No internal interference No external interference	No internal interference External interference
	Invalid signals	(Internal interference) External interference	(Internal interference) External interference	N/A	Internal interference or External interference

motivated by two observations from passive observation module. First, in the absence of data traffic, the transmitter sleeps and only the receiver wakes up and obtain information about interference. Second, in the presence of data traffic, the information collected by the receiver covers that of the transmitter, which will be analyzed later. Therefore, the receiver has an advantage with respect to the collection of interference-related information.

Table 1 summarizes the absence or presence of interference as a function of both the transmitter status and the receiver status. Note that an entry "N/A" in the table means that the status combination cannot occur, whereas "(Internal interference)" means that internal interference may or may not occur during the data transmission on top of external interference. In practice, both transmitter and receiver only have a local perspective on the current status: the transmitter/receiver only have knowledge about the corresponding column/row, respectively, as they are not aware of the status of the other device. We enumerate and discuss next that the receiver can resolve any ambiguity without knowing the transmitter status.

Clear channel. If a receiver detects a clear channel, then it can unambiguously infer the absence of any interference.

Overheard packets. If a receiver has overheard a packet destined to another node, it can unambiguously infer the presence of internal interference. Note that, in principle, also external interference may be affecting the link, as the transmitter may have experienced a CCA check failure. This case, however, is rather unlikely, and could be inferred by the receiver through an analysis of the RSS samples.

Packet received & ACK sent. In case the receiver has successfully received a packet and sent the corresponding ACK, it cannot know whether external interference affects the link, i.e., whether the transmitter receives its ACK correctly or not. However, the receiver can still infer the status of the transmitter by keeping track of its future transmissions. In fact, the transmitter will re-transmit the packet if it has not been acknowledged: this will result in the reception of a duplicate packet in the next slotframe.

Invalid signal. In case the receiver detects an invalid signal, a node can infer the presence of external interference when the average value of all the collected RSS samples exceeds a pre-defined threshold (−60 dBm in our implementation), as external interference continues in the channel and is typically generated by devices operating at a higher power than IEEE 802.15.4 nodes.

7.2 Inference from Long-term Observations

Using the aforementioned method, the receiver in SmarTiSCH is able to transform local information including (i) the recorded status, (ii) an analysis of RSS samples, and (iii) the existence of duplicate packets to the presence or absence of interference at a given timeslot. To properly react to interference, each node should have a long-term observation about the severity of a link's interference.

Considering that the nodes use very low power (so to achieve a long lifetime), the method to infer the intensity of interference should have minimal overhead. SmarTiSCH uses the interference probability to capture the intensity of internal and external interference, respectively. To efficiently monitor the channel quality, SmarTiSCH uses the Exponentially Weighted Moving Average (EWMA) because of its quick computation and stability. Let P_i be the receiver's estimated probability of a certain type of interference at the i^{th} timeslot: the receiver can update P_i as follows.

$$P_i = \begin{cases} (1-\lambda)P_{i-1} + \lambda, & \text{Interference} \\ (1-\lambda)P_{i-1}, & \text{No interference} \\ P_{i-1}, & \text{Not clear} \end{cases} \tag{1}$$

where $0 < \lambda \leq 1$ is a weight for a new sample. Note that a large λ yields a fast decay of older information and thus leads to a faster detection of interference (but to a lower stability). We set λ according to the slotframe length in our implementation.

To capture the intensity of the internal interference quicker, SmarTiSCH can set the initial value of P_i based on the traffic load of the network. To increase the confidence when inferring the presence of external interference, a node that serves as parent of multiple children can also merge several observations, as external interference is likely to affect all the nodes in the same area.

When the detected interference is severe, i.e., when the interference probability exceeds a predefined threshold (0.3 in our implementation), the receiver will select a mitigation strategy based on the *reaction to interference* module, which is described next.

8 REACTION TO INTERFERENCE

In this section, we detail the design of the *reaction to interference* module of SmarTiSCH. Now that the receiver of the link infers the interference based on status, it needs to share the inference with the transmitter so that they take consistent reactions. For this purpose, the transmitter and the receiver autonomously enter the control channel and use the allocated cell in control channel to re-transmit the interfered data packet. The receiver embeds information about the chosen mitigation strategy in the ACK packet that is sent back to the transmitter: it does so by tuning the instant of time at which ACK packets are sent. The rest of this section will elaborate on the design of data re-transmissions, on the tuning of the ACK timing, and on the strategies to mitigate interference.

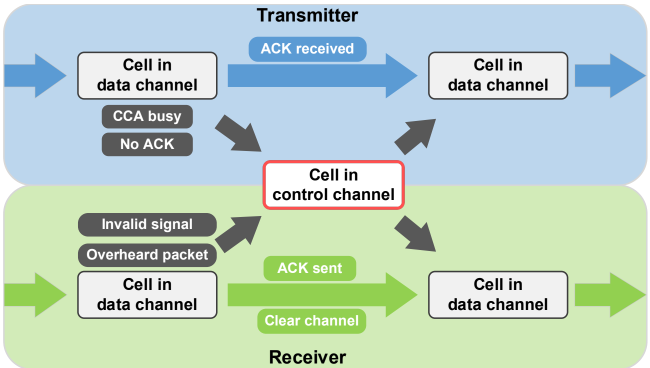

Figure 6: Mapping between transmission status and next selected timeslot. SmarTiSCH drives the transmitter and the receiver to autonomously use the control channel for data re-transmission when presence of interference is observed.

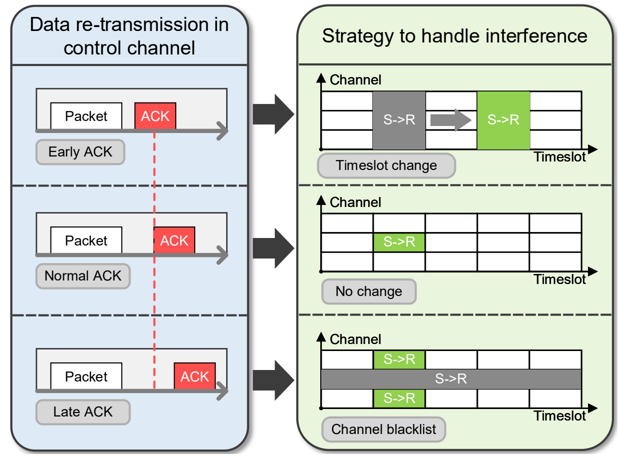

Figure 7: In SmarTiSCH, the receiver embeds information about the chosen interference mitigation strategy in the transmission timing of the ACK. Specifically, an early ACK denotes internal interference, whilst a late ACK denotes external interference. This mechanisms ensures that transmitter and receiver agree on the interference mitigation strategy.

8.1 Ensuring Consensus

Data re-transmission in the control channel. After the transmitter and the receiver have separately observed the presence of interference, they will autonomously enter the cell in control channel, in order to attempt a data re-transmission. Note that nodes do not make use of the control channel in case they have inferred the absence of interference. This is important, as it minimizes the utilization of the control channel.

The decision logic of both the transmitter and the receiver is shown in Fig. 6. When internal interference corrupts the data packet, the transmitter senses no ACK and the receiver cannot parse a packet during reception (invalid signal due to packet collision or overheard packet due to capture effect). Both nodes will hence enter the cell in the control channel for data re-transmission. Similarly, when external interference is present, the transmitter senses a busy channel when performing a CCA check prior to the data transmission or sends a packet but receives no ACK. The receiver also senses an invalid signal when listening to the channel for the possible packet from the transmitter. They both decide to use the cell in control channel for data re-transmission. Note that SmarTiSCH is also able to handle external interference that is not detectable by the sender. When this situation occurs, the receiver detects invalid signals and listens in the slot operating on the control channel: hence, the transmitter receives no ACK in this slot and also re-transmits the packets in the control channel. Therefore, both transmitter and receiver use the slot in the control channel for data re-transmission.

Tuning the ACK timing. Other than allowing an attempt for data re-transmisison, the cell in the control channel also provides the opportunity for the receiver to inform the transmitter of the strategy it selects. Basically, there are three strategies for reaction, which correspond to the three possible conditions, i.e., no interference, internal interference, and external interference. SmarTiSCH adapts the schedule autonomously, so this information cannot be transferred by making use of extra control packets, which would go against the principles of autonomous scheduling. In SmarTiSCH, we let the receiver embed information about the chosen interference mitigation strategy in the timing of the ACK packet sent back to the transmitter, as shown in Fig. 7. This operation is inspired by time modulation in cross-technology communication [23]. The idea behind the modulation of the ACK to communicate a strategy allows to keep SmarTiSCH fully autonomous while also standard-compliant: in fact, SmarTiSCH does not require any changes in the packet format. This operation is also reliable despite the synchronization error between the transmitter and the receiver. This is because the transmission and reception of a packet builds a unified synchronization event between the transmitter and the receiver.

Specifically, the transmitter begins listening to the ACK after 600 μs, while the receiver transmits the ACK after 1080 μs by default. Therefore, the timing of ACK packets can be used to represent different symbols. Specifically, SmarTiSCH lets the receiver anticipate the ACK by 320 μs to represent the strategy handling internal interference, whereas it delays the ACK by 320 μs to represent the strategy handling external interference. The transmitter will correspondingly identify the strategy by checking the receiving time of the ACK. If the ACK has not been shifted, then transmitter and receiver will remain in the current cell for data transmissions. In this way, both transmitter and receiver can agree on the strategy chosen to mitigate the presence of interference.

8.2 Strategies to Handle Interference

SmarTiSCH utilizes timeslot adaption to handle internal interference and channel blacklisting to handle external interference. To handle internal interference, timeslot adaption is simple but effective, as the parameters about frequency depend on the receiver. To handle external interference that persists in one channel, we utilize the channel blacklisting technique.

Since the information embedded in the ACK timing is very limited, how to ensure that the strategy is consistently and appropriately implemented by either side needs further consideration. Our design of this part includes three different cases:

(1) *No change*. In this case, the receiver informs the transmitter to keep the normal data transmission as it is. This case usually happens when a data transmission fails due to poor link quality, but the receiver deems interference as not severe.
(2) *Synchronous timeslot change*. When the transmitter and the receiver agree on the timeslot change strategy, they both use an algorithm to map the current timeslot index to a new timeslot index. In our implementation, we use a linear congruential generator, i.e., an algorithm yielding a sequence of pseudo-randomized numbers calculated with a discontinuous segmented linear equation [26].
(3) *Channel blacklisting*. The transmitter and the receiver both blacklist the last used data channel, as external interference is detected. The link can blacklist the channel for a certain number of slotframes. The number of slotframes for blacklisting may be decided by taking into account the number of available data channels. When there are sufficient channel resources, the link may blacklist a channel for a relatively long period as it does not hurt the network throughput. Otherwise, the blacklisting should be kept short, so as to release channel resource in time for data transmissions. When the external interference changes, the blacklisted channel will be revoked afterwards (the link sets a counter for each blacklisted channel and re-uses these channels after the counter reaches zero); SmarTiSCH then detects and blacklists the channels currently affected by external interference.

SmarTiSCH enables the nodes to handle internal and external interference separately. The links observe internal and external interference based on two estimated probability values, respectively, and then adopt the corresponding strategy for reaction when either of the interference is severe (i.e., when one observed value exceeds the defined threshold). When both values exceed the threshold at the same time, we let the links first change the timeslot to solve internal interference, and then use channel blacklisting to handle external interference.

9 PRACTICAL ISSUES

We discuss next a few practical implementation aspects.

9.1 Slotframe Settings

In our implementation of SmarTiSCH, we define three types of slotframe with different lengths to support network formation and to exchange data between nodes. The first type of slotframe is used to exchange EBs, and has a length of 397 timeslots (same setting as in Orchestra). The second type of slotframe is used to exchange unicast packets in the data channels and to carry out their re-transmission in the control channels. The length of this slotframe is adjustable, and is commonly selected as a prime number. In our implementation we select lengths between 17 and 71 timeslots. The third type of slotframe has a length of 31 timeslots and is used to send RPL control packets for network formation.

9.2 Channel Selection

The selection of which channels to use in SmarTiSCH is flexible depending on the spectrum availability in the scenario at hand. A simple channel selection consists in using the four IEEE 802.15.4 channels in the 2.4 GHz band that are the least interfered from Wi-Fi (i.e., 15, 20, 25, 26) as done in [9]. In this setting, SmarTiSCH uses channel 26 as control channel and channels 15, 20, 25 as data channels. This choice is driven by the fact that channel 26 does not overlap with any of the IEEE 802.11 channels operating at 2.4 GHz, and it is therefore unlikely to suffer from heavy Wi-Fi interference.

When all IEEE 802.15.4 channels are available, there will be three channel sets in SmarTiSCH. The control channel set including channels 15, 20, 25 and 26 will be used for sending control packets like EBs and data re-transmissions. Two data channel sets are used for normal unicast packets: one of them uses the channel hopping sequence *11, 16, 21, 13, 18, 23*; the other one uses the channel hopping sequence *12, 17, 22, 14, 19, 24*. The use of frequency sets retains the advantage of frequency hopping [48] while decreasing internal interference, since two links using different frequency sets will not affect each other's communications. Note that when there are multiple data channel sets, SmarTiSCH also allocates the frequency set (FS) during the schedule initialization as follows:

$$FS = mod(Hash(ID(R)), N_{DS})$$

where N_{DS} is the number of data channel sets, and $ID(R)$ is the address of the receiver.

9.3 Minimizing the energy cost

SmarTiSCH minimizes the energy cost caused by idle listening for interference mitigation by carefully designing the decision logic at the receiver side: the idea is that a node only listens in the control channel in the presence of interference. As shown in Figure 6, when no interference exists, the receiver detects a clear channel or the sender's transmission, and – after sending an ACK – then skips listening, i.e., it sleeps in the slots assigned in the control channel. As a result, when there is no interference or after the interference is handled by SmarTiSCH, the receiver listens and collects information only in the slots assigned in data channel, i.e., the slots that are commonly used for data transmission in a TSCH network.

10 EVALUATION

We evaluate the performance of SmarTiSCH experimentally and compare it to the state-of-the-art, focusing on receiver-based Orchestra (Orchestra) [9] and link-based Orchestra (L-Orchestra) [24][4]. Note that SmarTiSCH here refers to the link-based Orchestra with our engine. Unless differently specified, these three protocols all use four channels (15, 20, 25 and 26). We also evaluate the protocols using all 16 ZigBee channels in 2.4GHz, and make "X-16" denote protocol X with 16 channels. For example, "SmarTiSCH-16" refers to SmarTiSCH using all 16 channels with configuration detailed in Sect. 9.2. Note that we do not compare SmarTiSCH with A3 [25] because they target different problems and are complementary to each other to improve the performance of a TSCH-based network. A3 proposes to fulfill high traffic demands at the links near the root node by allocating more slots for those links, while SmarTiSCH enables the nodes to detect and handle interference. As they both

[4]L-Orchestra implements the core idea behind the link-based scheduling of ALICE, and further improves performance by relating the frequency scheduling to the receiver only [11], which enables the receiver to simultaneously receive packets from multiple senders. L-Orchestra does not implement the cell change function in each slotframe. As demonstrated by [11], L-Orchestra has a similar average performance as ALICE.

provide an engine for link-based autonomous scheduling, it is possible to use either method to improve network performance. We will investigate a combination of these two schemes in future work.

After describing our experimental setup (Sect. 10.1), we first evaluate the performance of SmarTiSCH in the presence of internal interference (Sect. 10.2), and we then artificially generate Wi-Fi interference to evaluate the performance of SmarTiSCH under external interference (Sect. 10.3). We also evaluate the efficiency of SmarTiSCH in terms of control channel utilization and in terms of successful strategy exchanges (Sect. **??**).

10.1 Experimental Setup

We implement Orchestra, L-Orchestra and SmarTiSCH on Contiki-NG, and use RPL with storing mode on top. We evaluate the performance of SmarTiSCH on a dense multi-hop network consisting of 20 nRF52840 nodes deployed in a $50m^2$ area [35, 36, 38], showing that internal interference can cause severe issues across the links in the network. We then generate reproducible Wi-Fi interference using JamLab-NG [37] and show how external interference also negatively affects network performance. We emulate a data collection application at the root note (convergecast), i.e., each node generates packets with a predefined traffic load, and transfers the packet with a payload length of 50bytes to the root node along the upward links. Our implementation supports LLNs with dense deployment and heavy traffic under interference, which have been investigated in diversified use cases [20, 24, 55]. Each experiment lasts for ten minutes, and SmarTiSCH always works as an engine to detect and react to the interference. Unless otherwise specified, we set λ as 0.35 and 0.3 when producing the internal and external interference, respectively. When detecting external interference, the nodes blacklist the channel for 100 slotframes.

10.2 Performance under Internal Interference

We start by evaluating the performance of SmarTiSCH under internal interference against the state-of-the-art. Then, we analyze the link loss by tracking every lost packet and show the effectiveness of SmarTiSCH to handle internal interference.

Setup. We set the traffic load of each node as 30, 60 and 90 packets/min to generate different levels of internal interference. Note that the traffic load is much higher than the usual set (e.g., 2 packets/min) in a large sparse network, so the nodes in our set can also be regarded as the nodes with heavier traffic (e.g., parent node with a subtree of tens of nodes) in a large network. These nodes are often the bottleneck of the whole network [24, 25]. We then fix the length of the slotframe used to exchange unicast data to 13 timeslots to support such heavy data traffic, and at least two links will be allocated a shared cell with a probability of 97.7%. The traffic rates used for evaluation are similar to the ones chosen by ALICE [24], which supports LLNs with dense deployment and heavy traffic. This resembles real-world applications such as electronic shelf label systems with thousands of nodes and heavy traffic as well as industrial IoT systems monitoring environmental quantities, where sensor nodes require frequent data reporting for data analytics [20, 50, 55]. We evaluate the performance of SmarTiSCH and compare it with Orchestra and L-Orchestra in terms of: (i) average packet delivery ratio (PDR), (ii) average end-to-end latency from node to coordinator, (iii) average duty cycle of all nodes in the network, as well as (iv) link loss rate, i.e., the percentage of packets that were transmitted, but not ACKed. We also compare SmarTiSCH to SmarTiSCH-16 to observe the effect of the number of used channels. In each experiment, 8600 to 26300 packets are transmitted to the sink for of ten minutes. Each experiment is repeated ten times.

PDR. Fig. 8(a) shows the average packet delivery ratio of the three protocols at different traffic loads. SmarTiSCH sustains a PDR that is 32.4% higher than that of Orchestra, as well as 10.5% higher than that of L-Orchestra. Note that the gap between SmarTiSCH and L-Orchestra is larger when the traffic load is higher. At a traffic load of 90 packets/min, the PDR of SmarTiSCH is larger than L-Orchestra by about 20%. When the traffic load is higher, the links that use shared cell face higher link loss as the packets prone to collide due to dense traffic. SmarTiSCH enables the nodes in the network to change the timeslot when several links use a shared cell, which solves the problem caused by an inefficient cell allocation. Fig. 9(a) shows that the PDR of SmarTiSCH is only lower than SmarTiSCH-16 by about 2%. This also shows the ability of SmarTiSCH to adapt the schedule under internal interference as SmarTiSCH uses fewer available cells than SmarTiSCH-16 and are prone to have more shared cells during schedule initialization.

End-to-end latency. As shown in Fig. 8(b), the end-to-end latency of SmarTiSCH is lower than that of Orchestra by 87% and than that of L-Orchestra by 44%. We also measure the end-to-end latency of SmarTiSCH-16 and it is lower than SmarTiSCH by about 4%. The end-to-end latency of Orchestra is very high as the nodes with the same parent node suffer from collisions due to severe internal interference. These nodes have to re-transmit the packets in the latter slotframes with the binary exponential backoff algorithm. Under higher traffic load, the nodes have to re-transmit the packets several times after many slotframes. As we calculate the end-to-end latency, the latency at each link accumulates to a very high value; also L-Orchestra suffers from a similar problem. The latency of SmarTiSCH is obviously lower than Orchestra or L-Orchestra. Because when nodes face link loss in the shared cell, the nodes immediately re-transmit the packets in the control channel: thanks to the control cell allocation shown in Sect. 4, the packets will not collide again. This proves the potential of SmarTiSCH in increasing the performance in dense networks with high traffic load.

Duty cycle. Fig. 8(c) show the average duty cycle of all nodes in the network. As expected, Orchestra exhibits a lower duty cycle than L-Orchestra and SmarTiSCH (each node use only a single timeslot to receive packets from all its children). The difference between SmarTiSCH and L-Orchestra is not that large (roughly 0.1%) with a low traffic load. However, the performance of L-Orchestra significantly decreases when the traffic load increases. Due to higher link loss caused by internal interference, the nodes in L-Orchestra have to re-transmit collided packets. At a traffic load of 90 packets/min, the duty cycle of SmarTiSCH is lower than L-Orchestra by 1%. Note that the duty cycle of SmarTiSCH-16 is similar to SmarTiSCH with a difference of less than 0.2%.

Link loss. Generally, as shown by Fig. 8(d) and Fig. 9(b), the link loss of Orchestra and L-Orchestra increases with higher traffic load, while the link loss of SmarTiSCH and SmarTiSCH-16 are well

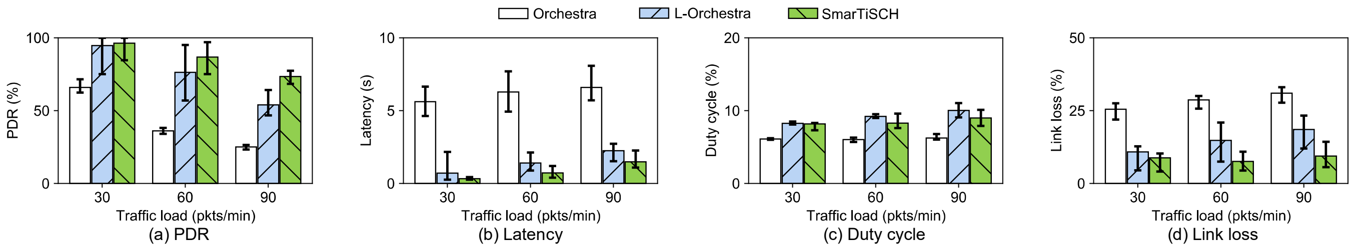

Figure 8: Comparison of SmarTiSCH to the state-of-the-art approaches under internal interference.

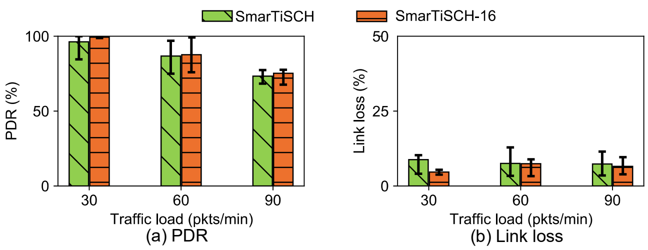

Figure 9: Comparison of SmarTiSCH and SmarTiSCH-16 under internal interference.

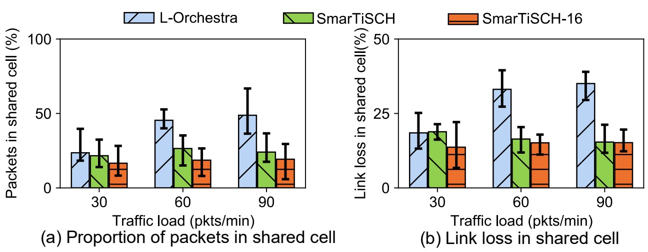

Figure 10: A detailed analysis of link loss of SmarTiSCH and the state-of-the-art approaches under internal interference.

under control. In Orchestra, as the nodes with the same parent node are allocated the same cell ($B \longrightarrow A$ and $C \longrightarrow A$ in Figure.1 for example), so nearly all the packets are transmitted in the shared cell and suffer from high link loss. So we mainly consider L-Orchestra, SmarTiSCH and SmarTiSCH-16 for a deeper analysis. At the traffic load of 90 packets/min, the link loss of SmarTiSCH and SmarTiSCH-16 is lower than L-Orchestra by 60.2% and 64.5%, respectively. The cause mainly lies in two aspects: the proportion of packets in shared cell and the link loss in shared cell.

Fig. 10(a) shows that in L-Orchestra, nearly 50% of packets are transmitted in the shared cell when the traffic load is high. While the proportion of packet in shared cell are 24% and 18% in SmarTiSCH and SmarTiSCH-16, respectively. In SmarTiSCH, the links are able to change the cell after detection of use of a shared cell, which reduces the amount of packets transmitted in the shared cell. In L-Orchestra, the links use a fixed cell allocation so the links using shared cell suffer from high link loss and keep re-transmitting the lost packets.

We also measure the link loss in shared cell and show it in Fig. 10(b). At the traffic load of 90 packets/min, SmarTiSCH reduces the link loss in the shared cell than L-Orchestra by 56.4% by average. This shows that SmarTiSCH is able to decrease the link loss in the shared cell to an acceptable level and improve the performance of the whole network.

10.3 Performance under External Interference

We evaluate next the performance of SmarTiSCH in the presence of the external Wi-Fi interference generated using JamLab-NG. Please refer to [3] for more details on the interference generation process.

Setup. To observe only the impact of external interference, we minimize internal interference by fixing the length of the unicast data slotframe as 71 timeslots (all the links will not use shared cells with the probability of 49.7%). We also decrease the traffic load of each node to 6 packets/min, as the nodes communicate less frequently with a longer slotframe. In this setting L-Orchestra and SmarTiSCH both achieve a PDR of over 98% when there is no external interference. We generate 4 types of external interference by letting Wi-Fi devices in the scenario generate traffic in Wi-Fi channel 2 and 7 using median or high intensity. We use the combination of channel and intensity to notify the interference type. For example, "2-m" means that the interference is generated by Wi-Fi devices using channel 2 and medium intensity.

To have a visual view of the impact of different types of external interference, we show the link loss of all 16 channels of ZigBee under 4 types of interference in Fig. 13. The center frequency of Wi-Fi channel 2 is 2.417GHz, between ZigBee channel 13 (2.415GHz) and 14 (2.420GHz), and Wi-Fi channel 7 (2.442GHz) is between ZigBee channel 18 (2.44GHz) and 19 (2.445GHz). When Wi-Fi device uses a high intensity, more ZigBee channels around its center frequency are interfered. Note that under interference "2-m" and "2-h", ZigBee channel 15 is badly interfered but channel 20, 25 and 26 are free from interference, so we show the performance of protocols using four channels under them together and note the type as "2-m/h". We then evaluate the performance of Orchestra, L-Orchestra, L-Orchestra-16, SmarTiSCH and SmarTiSCH-16, using the metric described in the Sect. 10.2.

PDR. Fig. 11(a) shows the average PDR of the three protocols using four channels under external interference, and Fig. 12(a) shows that of L-Orchestra-16 and SmarTiSCH-16. In general, the PDR of Orchestra, L-Orchestra and L-Orchestra-16 declines under interference "2-m", "7-m", and declines more sharply under interference "7-h", while the PDR decline of SmarTiSCH is less, and the PDR of SmarTiSCH-16 stays stable above 80%. The performance of protocols directly depends on the number of interfered channels. For protocols using four channels, "2-m/h" interferes channel 15, "7-m" interferes channel 20, and "7-h" interferes both of them. As a result, the PDR of L-Orchestra decreases by 32.4% with one interfered channel by 44.9% with two interfered channels. For SmarTiSCH, the decrease is 14.4% and 30.6%, respectively. The PDR decrease of SmarTiSCH under "7-h" is because that two interfered channels

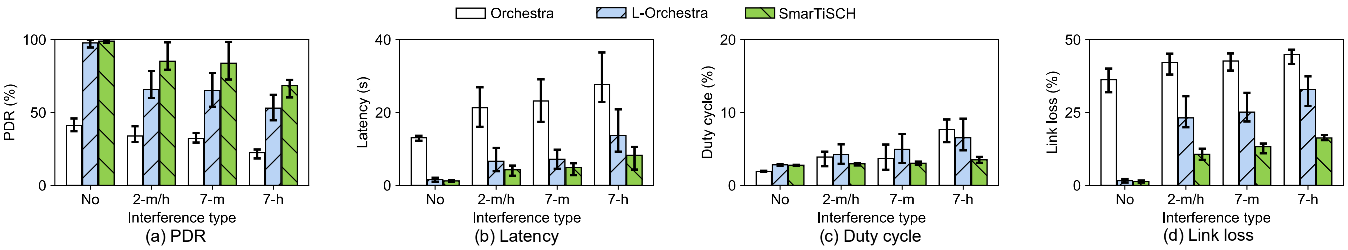

Figure 11: Comparison of SmarTiSCH to the state-of-the-art approaches using 4 channels under external interference.

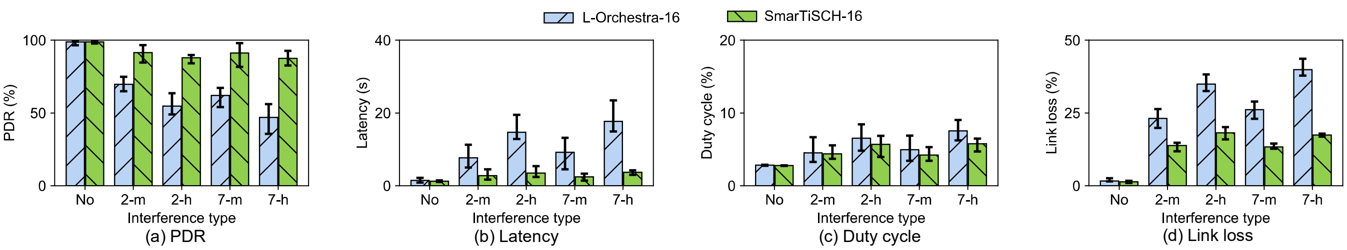

Figure 12: Comparison of SmarTiSCH to the state-of-the-art approaches using 16 channels under external interference.

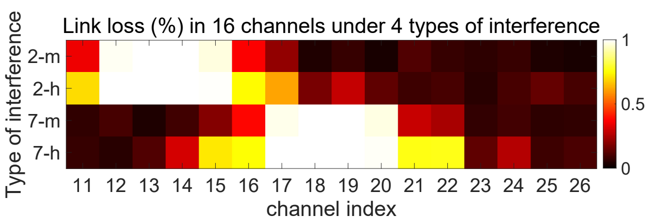

Figure 13: Link loss in all channels under 6 types of external interference.

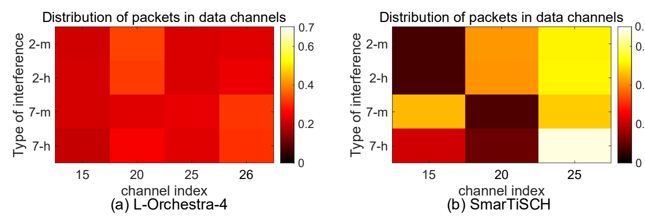

Figure 14: L-Orchestra uses all channels with no difference while SmarTiSCH blacklists interfered channels.

of three data channels are blacklisted, which reduce the channel resources to use. The PDR of L-Orchestra-16 decrease by 32.9% in average under "2-m", "7-m", and by 47.0% under "2-h" and "7-h". For SmarTiSCH-16, the PDR only decreases by 9.2% and 12.4% in the same settings thanks to the blacklisting techniques shown in Sect. 8.2. Under interference "7-h", the PDR of SmarTiSCH-16 larger by Orchestra by 2.9 times and L-Orchestra-16 by 0.87 times.

End-to-end latency. As shown in Fig. 11(b) and Fig. 12(b), the end-to-end latency of SmarTiSCH and SmarTiSCH-16 is much lower than that of Orchestra, L-Orchestra and L-Orchestra-16. In Orchestra and L-Orchestra, the nodes frequently back-off due to external interference. SmarTiSCH solves this problem by blacklisting the interfered channels and then skip the transmissions in them. Note that SmarTiSCH-16 further lowers the latency than SmarTiSCH, because it has more available channels after channel blacklisting. Specifically, under interference "7-h", after blacklisting channels, SmarTiSCH only has one thirds of channels to use while SmarTiSCH-16 has half of the channels to use. Take the link in SmarTiSCH-16 using data channel set 11, 16, 21, 13, 18, 23 for example: channel 16, 18 and 21 are blacklisted, leaving channel 11, 13 and 23 to use. Under interference "7-h", the end-to-end latency of SmarTiSCH-16 is lower than Orchestra, L-Orchestra and L-Orchestra-16 by 86.7%, 73.1% and 79.2%, respectively.

Duty cycle. Fig. 11(c) and Fig. 12(c) show that the duty cycle of SmarTiSCH is lower than that of Orchestra and L-Orchestra under external interference, i.e., SmarTiSCH exhibits a higher energy efficiency. The duty cycle of Orchestra and L-Orchestra increases quickly with more interfered channels. The duty cycle of SmarTiSCH stays stable with more interfered channels as it skips the transmissions in blacklisted channels. The duty cycle of SmarTiSCH-16 is larger than that of SmarTiSCH, as SmarTiSCH-16 has more available channels that are not blacklisted (so more slotframes to transmit and receive). Under interference "7-h", the duty cycle of SmarTiSCH is lower than Orchestra, L-Orchestra and L-Orchestra-16 by 54.3%, 46.6% and 53.7%, respectively.

Link loss. As shown by Fig. 11(d) and Fig. 12(d), the link loss of SmarTiSCH is much lower than Orchestra and L-Orchestra under any type of interference. The link loss of L-Orchestra reaches 24.4% in average with one interfered channel and 32.9% with two interfered channels. SmarTiSCH keeps link loss as low as 12.2% with one interfered channel and 16.6% with two interfered channels. Under interference "7-h", the link loss of SmarTiSCH is lower than Orchestra and L-Orchestra by 63.7% and 50.6%, respectively, and the link loss of SmarTiSCH-16 is lower than L-Orchestra-16 by 56.4%.

The effect of channel blacklisting. We show the effect of channel blacklisting used in SmarTiSCH by analyzing the distribution of

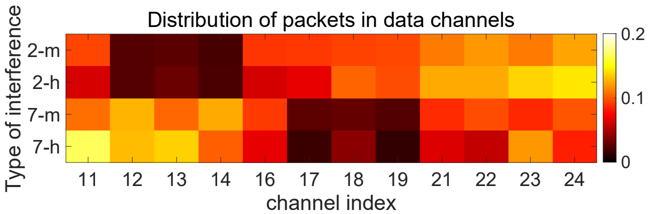

Figure 15: SmarTiSCH-16 blacklists interfered channels.

packets sent in different channels. As shown in Fig. 14 and Fig. 15, L-Orchestra transmits packets in all data channels with no difference, while SmarTiSCH and SmarTiSCH-16 blacklist interfered channels and avoid transmissions in these channels.

We first consider L-Orchestra and SmarTiSCH as they both use 4 channels. In L-Orchestra, under no matter what type of interference, the proportion of packets in every channel is from 22% to 33%. In SmarTiSCH, the proportion of packets in every channel is in inverse proportion to the link loss in this channel shown in Fig. 13. The impact of "2-m" and "2-h" on channel 15, 20, 25 and 26 is very similar, so the packets distribution under these two types of interference is also similar. In average, only 7% of packets are transmitted in channel 15, which is blacklisted by SmarTiSCH. Under interference "7-m", 45% and 47% of packets are transmitted in channel 15 and 20 that are free from external interference. Interference "7-h" incurs high link loss in both channel 15 and 20, so 68% of packets are transmitted in channel 25 in this case.

We do not show packet distribution of L-Orchestra-16 as it is similar to Fig. 14: the nodes select all available channels for transmission with no difference under any type of interference. The packet distribution of SmarTiSCH-16 in data channels is shown in Fig. 15. Combing this figure with Fig. 13, we find that the higher link loss the packets suffer from in the channel, the fewer packets the nodes transmit in it as more nodes blacklist it. Under "2-m", the packets in channel 12, 13 and 14 all together are fewer than 7.1%. Under "7-m", the packets in channel 17, 18 and 19 all together are fewer than 8.1%. This shows the effectiveness of SmarTiSCH to detect and handle external interference including high-power Wi-Fi transmissions by blacklisting interfered channels.

Comparison to other protocols. Note that SmarTiSCH is explicitly designed to increase the performance of TSCH-based networks in the presence of external interference. We focus on TSCH-based network, as they are standardized and widely used in real-world deployments. However, it is important to acknowledge that solution based on concurrent (synchronous) transmissions [1, 56] have recently proven their effectiveness in mitigating the impact of external interference, for example in the context of the EWSN Dependability Competition [5, 35], where solutions such as [12, 16, 27] could sustain interference levels higher than those tested here and obtain both lower duty cycles and latencies. However, concurrent transmissions are not yet standardized [2], and SmarTiSCH is the first work that empowers TSCH-based networks with a solution that retains the simplicity and energy-efficiency of autonomous scheduling, while increasing the awareness as well as robustness to both internal and external interference.

Table 2: The performance of SmarTiSCH-16 with different configurations.

Configuration	PDR	Duty cycle	Link loss
Th1 = -60 dBm; Th2 = 0.3	0.875	0.058	0.174
Th1 = -50 dBm; Th2 = 0.3	0.856	0.059	0.177
Th1 = -50 dBm; Th2 = 0.5	0.823	0.062	0.193

10.4 SmarTiSCH with Different Configurations

We also evaluate the performance of SmarTiSCH-16 with different configurations under the external interference type "7-h". We first vary the threshold mentioned in Sect. 7. Th1 ($-60dBm$ by default) is the threshold to infer the presence of external interference in short-term observations, and Th2 (0.3 by default) is the threshold to judge the severity of interference in long-term observations.

As shown in Table. 2, to increase the Th1 or Th2 leads to a lower PDR and higher duty cycle. The nodes use a more strict condition to identify the presence of interference or to trigger reaction, so fewer links blacklist the interfered channels. Note that we cannot decrease Th1 or Th2 arbitrarily as it may lead to false positives when the channel is clear from interference. The false positive rate of our default setting is nearly zero under no interference.

11 CONCLUSIONS

In this paper, we present SmarTiSCH, an interference-aware engine for IEEE 802.15.4-based networks. SmarTiSCH increases the awareness and robustness to interference without extra cost of scheduling. It lets each link in the network passively collect statistics about the presence of interference, infers the severity of it and then ensures that the node pair uses a consistent strategy to mitigate interference. Based on the engine to observe and react to both internal and external interference, SmarTiSCH expands the channel resources, increases the network capacity, and improves the network performance without extra control traffic.

ACKNOWLEDGMENT

This work is supported in part by National Science Fund of China under grant No. 61772306 and the R&D Project of Key Core Technology and Generic Technology in Shanxi Province (2020XXX007).

REFERENCES

[1] M. Baddeley et al. 2020. The Impact of the Physical Layer on the Performance of Concurrent Transmissions. In *Proc. of the 28th ICNP Conf.* 1–12.

[2] M. Baddeley et al. 2021. 6TiSCH++ with Bluetooth 5 and Concurrent transmissions. In *Proc. of the 18th EWSN Conf.*

[3] C.A. Boano et al. 2011. Jamlab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *Proc. of 10th IPSN Conf.*

[4] C.A. Boano et al. 2013. External Radio Interference. In *Radio Link Quality Estimation in Low-Power Wireless Networks*. Springer.

[5] C.A. Boano et al. 2017. EWSN Dependability Competition: Experiences and Lessons Learned. *IEEE Internet of Things Newsletter* (2017).

[6] A. Cerpa et al. 2005. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *Proc. of the 6th MobiHoc Conf.*

[7] IE Commission et al. 2010. Industrial Communication Networks-Wireless Communication Network and Communication Profiles-WirelessHART. *Switzerland: IEC* (2010).

[8] P. Du et al. 2012. Adaptive Time Slotted Channel Hopping for Wireless Sensor Networks. In *Proc. of the 4th CEEC Conf.*

[9] S. Duquennoy et al. 2015. Orchestra: Robust Mesh Networks through Autonomously Scheduled TSCH. In *Proc. of the 13th SenSys Conf.).*

[10] A. Elsts et al. 2017. Adaptive Channel Selection in IEEE 802.15.4 TSCH Networks. In *Proc. of the GIoTS Conf.*

[11] A. Elsts et al. 2020. An Empirical Survey of Autonomous Scheduling Methods for TSCH. *IEEE Access* 8 (2020).

[12] A. Escobar et al. 2018. BigBangBus. In *Proc. of the 15th EWSN Conf., competition session.*

[13] P.H. Gomes et al. 2018. MABO-TSCH: Multihop and Blacklist-based Optimized Time Synchronized Channel Hopping. *TETT* 29, 7 (2018).

[14] IEEE 802.15.4 Working Group. 2012. *IEEE Standard for Local and Metropolitan Area Networks - Part 802.15.4e-2012: Low-Rate Wireless Personal Area Networks (LR-WPANs); Amendment 1: MAC sublayer.*

[15] F. Hermans et al. 2013. SoNIC: Classifying Interference in 802.15.4 Sensor Networks. In *Proc. of the 12th IPSN Conf.*

[16] T. Istomin et al. 2018. Interference-Resilient Ultra-Low Power Aperiodic Data Collection. In *Proc. of the 17th IPSN Conf.*

[17] K. Jain et al. 2005. Impact of Interference on Multi-hop Wireless Network Performance. *Wireless networks* 11, 4 (2005).

[18] K. Jeon et al. 2017. Adaptive Channel Quality Estimation Method for Enhanced Time Slotted Channel Hopping on Wireless Sensor Networks. In *Proc. of the 9th ICUFN Conf.*

[19] M. Jin et al. 2016. Smoggy-link: Fingerprinting Interference for Predictable Wireless Concurrency. In *Proc. of the 24th ICNP Conf.*

[20] D. Jung et al. 2017. Vibration Analysis for IoT Enabled Predictive Maintenanc. (2017).

[21] J. Jung et al. 2020. Distributed Slot Scheduling for QoS Guarantee over TSCH-based IoT Networks via Adaptive Parameterization. In *Proc. of the 19th IPSN Conf.*

[22] H. Kim et al. 2017. Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey. *IEEE Communications Surveys Tutorials* 19, 4 (2017).

[23] S. Kim et al. 2015. Freebee: Cross-technology Communication via Free Side-Channel. In *Proc. of the 21st MobiCom Conf.*

[24] S. Kim et al. 2019. ALICE: Autonomous Link-based Cell Scheduling for TSCH. In *Proc. of the 18th IPSN Conf.*

[25] S. Kim et al. 2021. A3: Adaptive Autonomous Allocation of TSCH Slots. In *Proc. of the 20th IPSN.*

[26] P. L'ecuyer. 1999. Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure. *Math. Comp.* 68, 225 (1999).

[27] X. Ma et al. 2020. Harmony: Saving Concurrent Transmissions from Harsh RF Interference. In *Proc. of INFOCOM Conf.).* 1024–1033.

[28] A. Morell et al. 2013. Label Switching over IEEE 802.15.4e Networks. *Transactions on Emerging Telecommunications Technologies* 24, 5 (2013).

[29] E. Municio et al. 2016. Decentralized Broadcast-based Scheduling for Dense Multi-hop TSCH Networks. In *Proc. of the Workshop on MobiArch Conf.*

[30] J. Ortiz et al. 2010. Multichannel reliability assessment in real world WSNs. In *Proc. of the 9th IPSN Conf.*

[31] M.R. Palattella et al. 2012. Traffic Aware Scheduling Algorithm for Reliable Low-power Multi-hop IEEE 802.15.4e Networks. In *Proc. of the 23rd PIMRC Conf.*

[32] M.R. Palattella et al. 2013. On Optimal Scheduling in Duty-cycled Industrial IoT Applications using IEEE 802. 15.4e TSCH. *IEEE Sensors Journal* 13, 10 (2013).

[33] M.R. Palattella et al. 2015. On-the-fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks. *IEEE Sensors* 16, 2 (2015).

[34] A. Saifullah et al. 2010. Real-time Scheduling for WirelessHART Networks. In *Proc. of the 31st RTSS.*

[35] M. Schuß et al. 2017. A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge. In *Proc. of the 14th EWSN Conf.*

[36] M. Schuß et al. 2018. Moving Beyond Competitions: Extending D-Cube to Seamlessly Benchmark Low-Power Wireless Systems. In *Proc. of the 1st CPSBench Worksh.*

[37] M. Schuß et al. 2019. JamLab-NG: Benchmarking Low-Power Wireless Protocols under Controllable and Repeatable Wi-Fi Interference. In *Proc. of the 16th EWSN Conf.*

[38] M. Schuß et al. 2020. Poster: Making D-Cube an Open Low-Power Wireless Networking Benchmark. In *Proc. of the 17th EWSN Conf.*

[39] D. Son et al. 2006. Experimental Study of Concurrent Transmission in Wireless Sensor Networks. In *Proc. of the 4th SenSys Conf.* ACM, 237–250.

[40] R. Soua et al. 2012. MODESA: An Optimized Multichannel Slot Assignment for Raw Data Convergecast in Wireless Sensor Networks. In *Proc. of the 31st IPCCC.*

[41] R. Soua et al. 2016. Wave: A Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks. *Transactions on Emerging Telecommunications Technologies* 27, 4 (2016).

[42] K. Srinivasan et al. 2008. The β-factor: measuring wireless link burstiness. In *Proc. of the 6th SenSys Conf.*

[43] L. Stabellini et al. 2010. Energy-efficient Detection of Intermittent Interference in Wireless Sensor Networks. *International Journal of Sensor Networks* 8, 1 (2010).

[44] ISA Standard. 2009. Wireless systems for industrial automation: process control and related applications. *ISA-100.11 a-2009* (2009).

[45] R. Tavakoli et al. 2015. Enhanced Time-slotted Channel Hopping in WSNs using Non-intrusive Channel-Quality Estimation. In *Proc. of the 12th MASS Conf.*

[46] R. Tavakoli et al. 2018. Dependable Interference-aware Time-slotted Channel Hopping for Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* 14, 1 (2018).

[47] R. Tavakoli et al. 2018. Topology Management and TSCH Scheduling for Low-latency Convergecast in In-vehicle WSNs. *IEEE Transactions on Industrial Informatics* 15, 2 (2018).

[48] T. Watteyne et al. 2009. Reliability through frequency diversity: why channel hopping makes sense. In *Proc. of the 6th PE-WASUN Conf.* 116–123.

[49] T. Watteyne et al. 2013. Technical Overview of SmartMesh IP. In *Proc. of IMIS Conf.*

[50] J.M. Williams et al. 2017. Weaving the Wireless Web: Toward a Low-Power, Dense Wireless Sensor Network for the Industrial IoT. *IEEE Microwave Magazine* 18, 7 (2017).

[51] T. Winter et al. 2012. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.* Technical Report. IETF.

[52] P. Zand et al. 2013. D-MSR: A Distributed Network Management Scheme for Real-time Monitoring and Process Control Applications in Wireless Industrial Automation. *IEEE Sensors* 13, 7 (2013).

[53] H. Zhang et al. 2009. Optimal Link Scheduling and Channel Assignment for Convergecast in Linear WirelessHART Networks. In *Proc. of the 7th WiOpt Conf.*

[54] X. Zheng et al. 2014. Zisense: Towards Interference Resilient Duty Cycling in Wireless Sensor Networks. In *Proc. of the 12th SenSys Conf.*

[55] G. Zhou et al. 2006. Crowded Spectrum in Wireless Sensor Networks. *Proc. of the 3rd Workshop on EmNets* 6 (2006).

[56] M. Zimmerling et al. 2021. Synchronous Transm. in Low-Power Wireless: A Survey of Comm. Protocols and Network Services. *ACM Comp. Surv.* 53, 6 (2021).