Troubleshooting Wireless Coexistence Problems in the Industrial Internet of Things

Ulf Wetzker*‡ Ingmar Splitt* Marco Zimmerling† Carlo Alberto Boano‡ Kay Römer‡
*Fraunhofer Institute for Integrated Circuits, Division Engineering of Adaptive Systems, Dresden, Germany
†Center for Advancing Electronics Dresden, Dresden University of Technology, Germany
‡Institute for Technical Informatics, Graz University of Technology, Austria
{ulf.wetzker, ingmar.splitt}@eas.iis.fraunhofer.de marco.zimmerling@tu-dresden.de {cboano, roemer}@tugraz.at

***Abstract*—The ever-growing proliferation of wireless devices and technologies used for Internet of Things (IoT) applications, such as patient monitoring, military surveillance, and industrial automation and control, has created an increasing need for methods and tools for connectivity prediction, information flow monitoring, and failure analysis to increase the dependability of the wireless network. Indeed, in a safety-critical Industrial IoT (IIoT) setting, such as a smart factory, harsh signal propagation conditions combined with interference from coexisting radio technologies operating in the same frequency band may lead to poor network performance or even application failures despite precautionary measures. Analyzing and troubleshooting such failures on a large scale is often difficult and time-consuming. In this paper, we share our experience in troubleshooting coexistence problems in operational IIoT networks by reporting on examples that show the possible hurdles in carrying out failure analysis. Our experience motivates the need for a user-friendly, automated failure analysis system, and we outline an architecture of such system that allows to observe multiple communication standards and unknown sources of interference.**

I. Introduction

A *Computerworld* article from March 8, 1993 reported on the vision of General Magic, a former Apple Inc. spin-off developing the ancestor of the modern smartphone [1]:

> "... given another generation,
> wireless will be everywhere."

This vision has become a reality. In 2015, the total number of mobile subscriptions surpassed the world's population of 7.3 billion [2], and even conservative predictions by Ericsson, McKinsey, and Garnter estimate that the *Internet of Things (IoT)* will consist of 20–30 billion connected devices by 2020 [3], the vast majority of which will be wireless.

The range of IoT applications is huge. Applications such as monitoring of parking spaces and waste containers are already being deployed today. In the future, the IoT will increasingly embrace smart sensors and actuators to directly control the physical world, including the machines, factories, and infrastructure that define our modern society. Referred to as *Cyber-Physical Systems (CPS)* or the *Industrial IoT (IIoT)*, the corresponding applications such as industrial automation and process control are *safety-critical* in nature and require that the underlying wireless networks operate dependably [4].

In addition to the notorious unreliability and unpredictability of wireless communications, one of the major challenges for a dependable IIoT is *cross-technology interference (CTI)*. This is because the ever-growing number of wireless IoT devices and technologies makes the unlicensed industrial, scientific, and medical (ISM) frequency bands increasingly crowded. Imagine, for example, a smart factory scenario, where thousands of devices using heterogeneous wireless standards, such as Bluetooth Low Energy (BLE), ZigBee, and Wi-Fi, operate in the same frequency band together with cordless phones and certain radio-frequency identification (RFID) systems. How can we ensure that these networks coexist without degrading each other's performance?

To this end, wireless standards have passive coexistence mechanisms built-in. For instance, *carrier sense multiple access with collision avoidance (CSMA/CA)* lets a device send only if it has sensed the channel to be free, and using *adaptive frequency hopping (AFH)* a device continuously switches its carrier frequency while trying to avoid frequencies that experience a high packet error rate. Improving the robustness of wireless networks against CTI is also an active area of research, exploring the use of, for example, forward error correction [5] and MIMO capabilities [6]. The goal of all these mechanisms is to *prevent* performance degradation and failures due to CTI. But it is hard to foresee all future changes and potential problems that come along with them, especially given that new wireless standards emerge frequently and that changing environmental conditions have a significant impact on wireless networks [7]. So how can we *troubleshoot* wireless co-existence problems once they have surfaced as a partial or complete failure of an IIoT application?

Our experience in troubleshooting coexistence problems in operational IIoT installations shows that this is often a labor-intensive and time-consuming task. The main reasons for this are that the process is largely manual, involves the use of many different tools, and requires expert knowledge in diverse fields. To illustrate the steps, pitfalls, and surprises one may encounter during troubleshooting, we share in this paper our experiences with the research community.

After giving some background on wireless coexistence in Section II, we report in Section III on our experience in troubleshooting coexistence problems in a prototypical smart factory and in an open-pit mine. Based on the lessons we learned from these real-world cases and the methodology we developed, discussed in Section IV, we describe in Section V

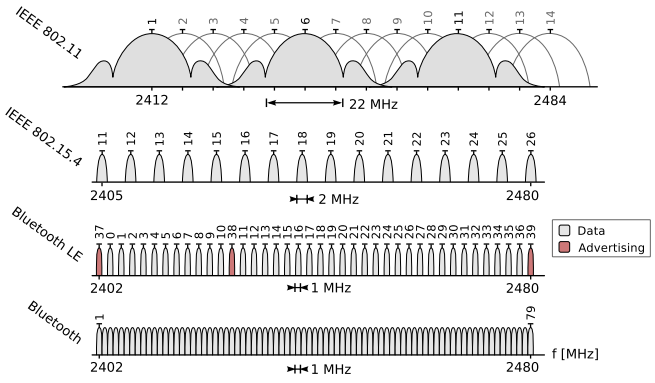

Figure 1. Popular technologies operating in the 2.4 GHz ISM band.

a novel architectural concept, which represents a first step towards a more systematic and (semi-)automated approach to troubleshooting wireless coexistence problems. We conclude this paper by discussing related work in Section VI and by providing guidelines for future research in Section VII.

II. Background on Wireless Coexistence

The growing proliferation of wireless devices causes an increasing congestion in the radio spectrum, turning it into an expensive resource [8]. Indeed, several standardized radio technologies operate in increasingly crowded ISM frequency bands, that is, freely-available portions of the radio spectrum reserved for industrial, scientific, and medical purposes [9].

The 2.4 GHz ISM band is a notable example of how crowded a portion of radio spectrum can be. Its worldwide availability made this band one of the most popular choices for wireless personal and local area networks. Figure 1 shows the four most pervasive technologies operating in the 2.4 GHz band: IEEE 802.11 (better known as Wi-Fi), IEEE 802.15.4 (the basis of the physical and media access control layer for low-rate wireless personal area and industrial networks such as ZigBee, ISA100.11a, and WirelessHART), IEEE 802.15.1 (commercialized as Bluetooth) and its evolution Bluetooth Smart or Bluetooth Low Energy (BLE). These standards specify different signal management functions, modulation schemes, power levels, data rates, channel bandwidths and separations. However, as visible in Figure 1, they all use overlapping frequencies. As a result, standard-compliant devices need to compete for medium access and may experience *cross-technology interference (CTI)* from surrounding appliances.

CTI may cause packet loss, unpredictable medium access delays, and high end-to-end latencies. Moreover, especially for low-power wireless devices with constrained energy budgets typically employed in IoT applications, CTI may also lead to reduced energy efficiency due to longer listening and contention times, packet re-transmissions, and loss of time synchronization. This is an important observation for low-power wireless sensor and actuator networks used in safety-critical IIoT scenarios, such as industrial control [10] and automation [11], health care [12], and high-confidence transportation systems [13], where guaranteeing high packet reception rates, bounds on end-to-end packet latency, and continuous system availability are of utmost importance.

The coexistence problem is further exacerbated by the fact that also common domestic appliances and other everyday devices can be a source of interference for wireless networks operating in the same frequency band. For instance, several works have shown that also the radio-frequency (RF) noise emitted by microwave ovens [14], electrodeless lamps [15], and other domestic appliances such as cordless phones, baby monitors, game controllers, presenters, and video capture devices [16] is harmful to low-power wireless technologies operating in the 2.4 GHz ISM band.

Also other ISM bands suffer from increasing congestion, although to a lower degree [17]. Telemetry networks [18] and cellular phones [19] can cause CTI in sub-GHz bands; for example, Global System for Mobile Communications (GSM) can impact low-power wireless transmissions during the first seconds of an incoming call [9]. The increasing popularity of long-range IoT technologies operating in sub-GHz bands (*e.g.*, IEEE 802.15.4g, LoRa, and SIGFOX) will soon saturate this portion of radio spectrum, further aggravating the problem of wireless coexistence. Similarly, industrial networks and localization systems based on ultra-wide band (UWB) technology powered by IEEE 802.15.4a devices begin to populate the 5.8 GHz band and hence need to coexist with Wi-Fi (IEEE 802.11n/ac) and a set of emerging technologies and solutions, such as Long-Term Evolution in unlicensed spectrum (LTE-U), radar systems and Google Loon[1].

It is clear that wireless networks enabling safety-critical IIoT applications urgently need solutions for connectivity prediction and validation. Connectivity prediction tools are important when using wireless technology to connect critical equipment (*e.g.*, accident prevention systems or cooling and corrosion detection units) and must account for the possible presence of heterogeneous radio-access technologies as well as allow for post-deployment network validation [21].

Such tools alone, however, are not sufficient. Even when following best-practice principles with precise connectivity information [22], it is hard to predict how the environment changes over time and therefore very difficult to ensure correct operation on a large scale for prolonged periods of time, even for technology experts. Debugging and troubleshooting tools are hence needed to analyze failures in depth and to provide hints on how to fix them. Such tools should accurately capture the deployment site's electromagnetic environment in the frequency, time, and spatial domain, as well as allow both an online analysis for on-sight troubleshooting and an off-line examination in case of more complex problems.

[1]Google Loon is a project aiming to provide Internet access in rural and remote areas using high-altitude balloons that are placed in the stratosphere and operate in the 2.4 and 5.8 GHz ISM bands [20].

To date, there is a clear lack of tools for troubleshooting coexistence problems in industrial networks [23], which is a major hurdle for engineers debugging cyber-physical and IIoT systems deployed on a large scale in remote locations. This lack of tools makes troubleshooting industrial wireless networks an exasperating, labor-intensive, and time-consuming task – sometimes a real *agony*.

III. Troubleshooting Industrial Wireless Networks: Our Hands-on Experience

This section details two exemplary cases from our experience in troubleshooting coexistence problems in operational IIoT installations, providing evidence of this agony.

A. Real-world Case 1: Gantry Robot

Developing an emulation platform for coexistence analysis in wireless automation systems is a viable approach to enable incremental deployment of new wireless communication components within an existing industrial wireless communication network [24]. In particular, the emulation of the production network at a test site enables test-driven development and integration under controlled, yet realistic conditions, while avoiding downtimes of the original production network.

To test wireless components under realistic coexistence conditions, one of the key ingredients of such an emulation platform is a parametrizable packet generator that mimics, for example, the cyclic traffic patterns of a programmable logic controller (PLC). As part of a research project to create such an emulation platform, we strove to develop statistical traffic models of typical wireless communication protocols in different IIoT application scenarios and environments.

Setting. To obtain such models, we chose the Experimental and Digital Factory (EDF) at Chemnitz University of Technology for a measurement campaign. The EDF is well-suited for experiments with industrial wireless communication systems, as it features all components of a future smart factory while accurately representing the environment of production lines with interconnected building blocks for adaptive factory systems. The EDF includes components for:

- machining (*e.g.*, manufacturing, assembly, test);
- material flow (*e.g.*, transportation, material handling);
- information flow (*e.g.*, control, interfacing);
- energy (*e.g.*, supply, management).

The industrial wireless network inside the EDF is based on IEEE 802.11n and the proprietary Siemens IWLAN protocol. During our measurements, we noticed two single-hop wireless transport systems connected to an access point (AP) acting as a gateway to the wired industrial communication system, as depicted in Figure 2: an Automatic Guided Vehicle (AGV) that receives mission data and sends position and collision avoidance information to a control module, and a gantry robot that connects wirelessly to a Human Machine Interface (HMI) to control the movement of the robot.

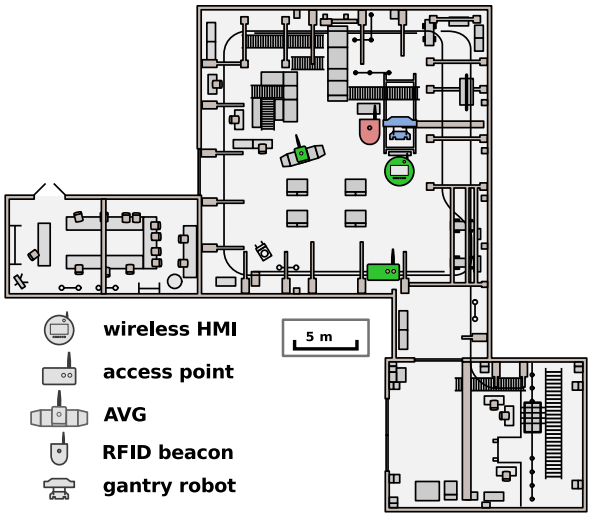

Figure 2. Floor plan of the Experimental and Digital Factory (EDF).

During the design phase of the IEEE 802.11n network in the EDF, a wireless network planning procedure was carried out as described in [25]. To mitigate interference factors, the guide recommends four remedies:

- increase the quality of radio coverage by increasing the number of APs in the network;
- use of directional antennas to extend and optimize the wireless connections;
- disable or re-position all coexisting wireless devices that operate on the same frequency;
- avoid the use of the wireless network for data transmissions if there are no effective countermeasures.

Following these guidelines, the position of the AP (see Figure 2) allowed for an adequate coverage of the operation area. To avoid interference with coexisting IEEE 802.11 networks, an unused frequency channel was assigned to the industrial wireless communication system. In addition, the guide recommends a simulation-based approach using the Siemens SINEMA E software for planning, simulating, and configuring a Wi-Fi network. In general, industrial plants are hard to model since they include large metal objects and moving components. Thus, a static simulation of complicated radio propagation environments can be misleading and was not performed during the planning of the wireless network.

Without the many restrictions of an industrial plant running in production, the EDF allowed us to reconfigure the network and deliberately interfere with the IEEE 802.11n network. To analyze possible changes in the traffic patterns caused by coexistence with other communication standards in the 2.4 GHz ISM band, we deliberately introduced interfering IEEE 802.15.4 and IEEE 802.11 networks. To determine the maximum interference level that the application could tolerate, we increased the data rate of the two interfering networks in a step-wise fashion until the transport system encountered an emergency stop and had to be reset.

Problem. At first, we successfully observed the communication pattern of the AVG driving in the hallway between the manufacturing and assembly modules. We recorded multiple traffic patterns from different driving maneuvers over a measurement period of 30 minutes. However, following the same procedure with the HMI-controlled gantry robot, the application faced emergency stops after a random period of operation. A local technician had observed the same behavior before, which could only be resolved by running the wireless network within the 5 GHz rather than in the 2.4 GHz band. We describe next the search for the cause of the emergency stops, which we assumed to be related to wireless coexistence.

Failure analysis. Initially, we inspected the diagnostic menu of the HMI to rule out obvious problems related to some misconfiguration or strong variations in the signal strength (*e.g.*, dead spots). Then, we used the network analysis tool Wireshark to observe the IEEE 802.11n traffic. A manual in-depth inspection of the captured packet traces provided a complete picture of the network structure, including a large number of previously unknown devices in the wired network behind the AP. This discrepancy was due to an unintended configuration of a managed switch, which resulted in broadcast and multicast messages from within the wired network being transmitted by the AP. Resolving this issue, however, did not fix the problem of emergency stops.

Capturing packets on overlapping Wi-Fi channels, we also observed multiple smartphones and a wireless VoIP telephone. Finding all coexisting devices of the same standard within reception range can be an exhausting task, especially if the cause of the system failure remains undetected even after disabling all observed devices in the surroundings.

Alternatively, the cause of a failure may be identified by reconstructing the technical background of the failure. We observed the packet stream between the HMI and the PLC, tracked down the communication scheme, and determined real-time critical heartbeat packets. Each emergency stop was preceded by at least one re-transmission of a heartbeat packet. Based on all these indications, we set up a simplified spectrum analyzer using an IEEE 802.15.4 node to search for other coexisting systems within the 2.4 GHz ISM band. Compared to professional measurement equipment, however, sampling the received signal strength (RSSI) using an IEEE 802.15.4 radio has low accuracy, low sampling rate, and includes spectral artifacts of the local oscillator harmonics. Nevertheless, our RSSI traces revealed another communication system with a bandwidth of 1 MHz that employed frequency hopping, as shown in Figure 3.

Bluetooth is the most widely used wireless communication system that matches these observations. After an extensive search for an active Bluetooth system, we noticed that the observed frequency hopping-based communication system constantly used the full 2.4 GHz ISM band even when we deliberately introduced strong interference using a coexisting Wi-Fi network. Most Bluetooth systems try to avoid interference with other wireless networks using the adaptive frequency hopping (AFH) algorithm[2]. In the presence of coexisting networks, the AFH algorithm dynamically changes the frequency hopping sequence of the devices, restricting the number of channels used by all Bluetooth nodes within the piconet and allowing other wireless systems to use the remaining frequency channels. As this was not the case, we ruled out Bluetooth as the cause of the emergency stops.

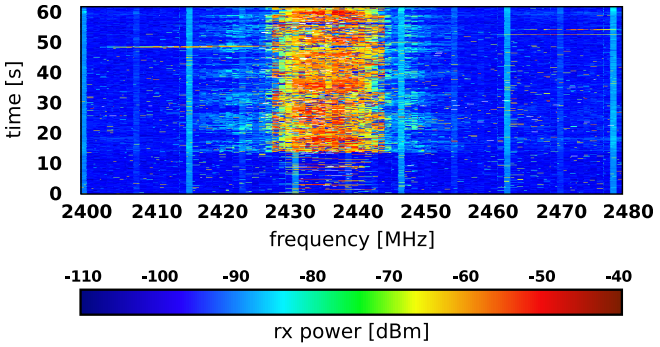

Figure 3. The 2.4 GHz spectrum captured at the EDF using an IEEE 802.15.4 node. Our measurements revealed a coexisting communication system with a bandwidth of 1 MHz employing frequency hopping.

Besides Bluetooth certain RFID systems operating according to the ISO 18000-4 standard can also use frequency hopping and a bandwidth of 1 MHz. Having this in mind, we started to narrow down the location where an emergency stop occurred after the shortest runtime of the system. By seeking the datasheets of all industrial devices in the area, we finally found a Siemens MOBY U active RFID transponder. After disabling the transponder, the HMI worked flawlessly. A high-resolution real-time spectrum analyzer might have helped to reveal the spectral differences between Bluetooth and RFID much earlier, but expert knowledge in wireless communications would have been required nevertheless.

Effort. The failure analysis was performed in 3 hours of intense work by two research engineers working in the field of industrial wireless communications and one technician of the EDF that supported the process with his detailed knowledge of the local installation setup. The time and effort we spent on this sudden troubleshooting process was relatively low considering that we had to improvise parts of our measurement equipment. Nevertheless, the work of three additional engineers was delayed by the faulty industrial system whereby the losses summed up to 18 person hours.

B. Real-worl Case 2: Surface Mining

Besides mobile machinery components, one of the most important application areas for industrial wireless communication systems is large-scale, inaccessible, or remote areas.

[2]The vast majority of Bluetooth devices uses adaptive frequency hopping, as this feature was introduced already in Bluetooth Revision 1.2 (2003).

Figure 4. The open-pit mine where the wireless network was installed.

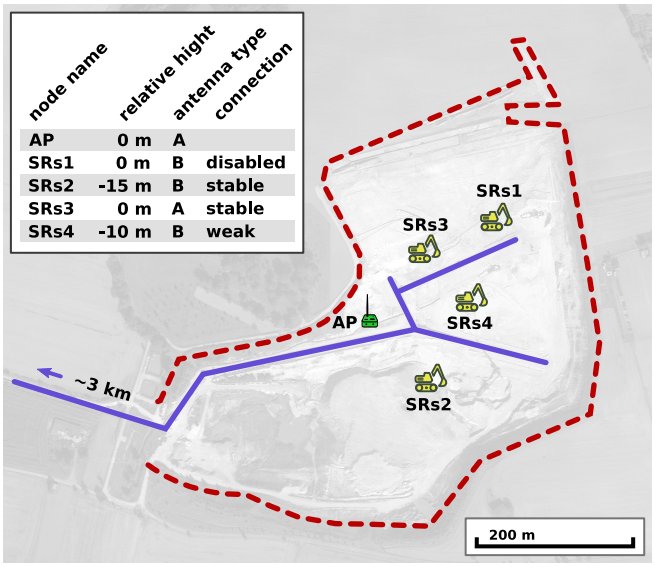

Figure 5. Floor plan of the open-pit mine, including the locations of the Wi-Fi AP and the four bucket-wheel excavators (SRs1–SRs4).

Setting up a wired network in a mining scenario is particularly challenging. This is, for example, because most of the mining equipment is mobile and there are often sections with no fixed cable installations. Also, the environmental conditions characterized by dust, humidity, undamped vibrations, and even explosive or abrasive substances are highly demanding. Thus, mining scenarios often rely on wireless communication to implement the required control loops. At the same time, however, the harsh environmental and extreme radio propagation conditions represent additional sources for failures of the wireless network. We describe next how we analyzed, together with a troubleshooting contractor for industrial communication systems, a failing wireless control network deployed in a kaolinite open-pit mine in Saxony, Germany.

Setting. As shown in Figure 4, the open-pit mine was about 0.25 km^2 in size and incorporated four TAKRAF SRs bucket-wheel excavators (SRs1–SRs4). Each excavator featured a flexible conveyor belt to carry the material to one of two fixed conveyor belts. These conveyor belts were placed between the working areas of the bucket-wheel excavators and converged into the main conveyor belt, as shown in Figure 5. With a total length of about 3 km, the conveyor belt system connected this smaller open-pit mine with the major mining complex, including the processing area and the main buildings.

A control mechanism prevented congestion in the hierarchical conveyor belt system. To this end, a wireless network based on industry-strength IEEE 802.11 equipment was used to interconnect the PLC of the conveyor belt system with the four SRs. A star topology connected the wireless stations on the bucket-wheel excavators (SRs1–SRs4) to an AP. As shown in Figure 5, all four wireless stations were within 220 meters from the AP with a maximum height difference of 15 meters with respect to the AP. The setup included two antenna types with omnidirectional radiation pattern: a 10 dBi collinear antenna (A) and a 3 dBi monopole antenna (B).

The IEEE 802.11 wireless network was designed according to the "Coexistence of Wireless Systems in Automation Technology" [22] guidelines of the German Electrical and Electronic Manufacturers' Association (ZVEI). The planning procedure was carried out by radio experts as recommended in the VDI/VDE 2185 guideline [26]. In addition, a site survey was performed using the Ekahau Site Survey (ESS) system to estimate the radio coverage in the open-pit mine. The latter consists of temporarily installing an AP, taking reception strength measurements at equally distributed locations across the area of interest, and deriving the resulting coverage as a heat map. The results revealed a weak signal strength only at a few locations behind the bucket-wheel excavators.

In addition, a spectrum analysis was performed to manually search for possible coexisting wireless devices. To this end, commercially available systems such as the Metageek Wi-Spy and the corresponding Channelizer software were used. As the mine is located far away from residential buildings, no coexisting device was detected at deployment time, and the IEEE 802.11 network was installed without precautionary measures or a coexistence management scheme.

Problem. Once the installation in the mine was finalized, the wireless network remained operational for about two years. Then the first connectivity problems occurred, and parts of the system had to be switched back to full manual control. These connectivity problems persisted over the following days, so a thorough failure analysis of the system was performed.

Failure analysis. A troubleshooting contractor for industrial communication systems started with an in-depth investigation to detect misconfigurations and hardware defects in the wired network. As no evidence for the cause of the connectivity problem was found, the wireless network was inspected next.

To this end, we used a distributed measurement system based on Raspberry Pi devices connected to a consumer-

grade IEEE 802.11 transceiver. At the time of the failure analysis, the wireless station at SRs1 was switched off, the connection between SRs4 and the AP was very unreliable, whilst SRs2 and SRs3 did not show connectivity issues. We therefore enclosed three Raspberry Pi nodes in a waterproof housing that included a power bank module and strapped them on each enabled bucket-wheel excavator. In order to get synchronized measurements from all nodes, the on-board crystal of the Raspberry Pi providing an accuracy of 140 ppm was backed up by a NXP PCF2127AT real-time clock with an accuracy of 3 ppm. The measurement procedure started with an NTP-based time synchronization by connecting the three Raspberry Pi nodes via Ethernet to a notebook. After this initial synchronization, the nodes were disconnected and started to capture wireless packets in monitor mode.

In addition to the distributed packet capture system, we also set up a simple spectrum analyzer using an IEEE 802.15.4 node (similar to the one described in Section III-A) to search for other coexisting systems within the 2.4 GHz ISM band. After observing the spectrum for more than three hours, we only saw Wi-Fi traffic in the surroundings. Most of this traffic used the same channel as the AP, and we could not detect any additional source of interference within the mining area. The observations made during the network planning phase described a similar coexistence scenario, but did not mention occasional IEEE 802.11 transmissions on other frequencies.

Next, we tried to rule out hardware failures. An investigation of the wireless equipment revealed that the fiberglass housing of all antennas was affected by environmental influences like UV degradation and that none of the transceiver inputs of the wireless stations had a lightning protection. The 10 dBi collinear antenna of the AP had a small crack inside its plastic cap, causing water to leak inside the fiberglass housing and changing the electromagnetic properties of the antenna. After replacing the antenna of the AP, however, the link between SRs4 and the AP was still unreliable. We also ruled out a hardware failure due to lightning, because this would have caused all stations to be affected by the same problem and is also more likely to compromise the wireless interface completely rather than causing partial packet loss.

As a next step, we analyzed the wireless traffic captured by the distributed Raspberry Pi nodes to further study the unreliable link between SRs4 and AP. As the spectral analysis revealed additional Wi-Fi traffic on other channels, we started to analyze the recorded traces searching for coexisting Wi-Fi networks employing common tools such as Wireshark and TCPdump. These tools focus on the dissection of the packet payload and provide the possibility to carry out basic data processing and filtering. We exploited these features to derive a snapshot of the role and the connection state of all detected wireless nodes. In addition to the industrial wireless network under study, the packet capture observed 29 different IEEE 802.11 stations during the observation period. Based on the traffic behavior and the manufacturer ID of the MAC address, we could recognize 22 smartphones, 5 tablet PCs, and 2 notebooks. However, none of these devices had an active connection creating a significant amount of traffic.

To confirm that none of these devices was causing the failure, we developed traffic analysis scripts that revealed the full network topology, including connections to 7 nodes in the wired network behind the AP. We found no significant correlation between the traffic from the 29 coexisting wireless nodes and the connectivity drops between SRs4 and the AP. Thus, to our surprise, these results were a clear sign that coexisting devices could not be the root cause of the failure.

A closer observation of the recorded packet traces revealed a low reception rate of packets sent by the AP. We found that the position of the monitoring node had a poor antenna path alignment with respect to the narrow beam width of the 10 dBi collinear antenna. Most of the packets received from SRs4 and the AP were indeed corrupted, given that the content of the packet did not match the frame check sequence (FCS). To include also these corrupted packets in our analysis, we used packet pre-processing algorithms to check the validity of the header information and to correct bit failures based on preceding packets. We further exploited the spatial diversity of the three monitoring nodes and cross-checked the corrupted information across all packet traces.

The enhanced dataset allowed us to identify a discrepancy between the received signal strength at SRs4 and the number of disassociation packets sent by the AP. By analyzing the management frames of the wireless network, we could notice that the link between SRs4 and AP experienced on average 6 associations and disassociations within 10 seconds. In other words, SRs4 continuously tried to initialize a connection, but the AP suspended the connection shortly afterwards. The reason code field of the disassociation packet indicated a high packet loss rate caused by low link quality. Interestingly, a monitoring node placed right next to the AP could receive packets from all active nodes, including SRs4, with a signal strength ranging between -70 dBm and -60 dBm, hinting that the link quality was good and similar across all connections.

However, consolidating a passively measured parameter and information extracted from within the wireless links gave a strong indication for a physical difference among the receive characteristics of the links. Considering the narrow beam-width of the 10 dBi collinear antenna, the minor differences within the orientation of the antennas and the gradually changing height and position of the four bucket-wheel excavators, the antenna path alignment might have changed up to a point where the directional antenna of the AP could barely receive the signal from SRs4. We therefore suggested a recalibration of the antenna system to reclaim full connectivity within the mining area. After restoring the connectivity between the AP and SRs4, we proposed as a long-term solution the use of omnidirectional wide-beam antennas and a denser wireless network with additional repeater nodes to compensate for the lower antenna gain.

Effort. The troubleshooting was particularly expensive in terms of personnel. Building the distributed measurement system involved two research engineers for 20 hours. The measurements in the kaolinite open-pit mine were escorted by a miner, who was responsible for the engineers' safety. Altogether the measurements required 14 person hours. The most labor-intensive part was the development of the offline analysis tools by two research engineers in about 128 hours. In total, it took about 4 person weeks to find and correct the failure within the wireless network of the open-pit mine.

IV. Discussion

The previous section highlighted that troubleshooting the IIoT can be tedious and time-consuming even for wireless communication experts. We now summarize the key lessons we learned and describe the methodology we developed while troubleshooting several industrial wireless networks.

A. Lessons Learned

Lesson 1: Manual approaches don't scale. An important lesson we learned is that almost every failure scenario requires a comprehensive black-box analysis of the wireless network, because the information provided by the network operator about the structure or state of the network and the surrounding conditions are often outdated or incomplete. Indeed, manual network planning and site survey procedures are typically only done once at deployment time, and manual coexistence management as proposed in industrial guidelines [22], [26] is too costly for most network operators. A (semi-)automated failure analysis to obtain complete, up-to-date information would help minimizing the overall troubleshooting time.

Lesson 2: Fusing data from diverse sources is key. To obtain such detailed view, it is necessary to collect accurate, high-resolution data from diverse sources, clean the data (*e.g.*, remove observations whose temporal order cannot be reliably reconstructed), and correlate them with one another. For instance, we learned from the open-pit mine case that passive observations can differ significantly from link-internal parameters due to differences in the hardware components (*e.g.*, amplifier, antenna, physical-layer implementation) or in the position of observation, which led to misconceptions that set us on the wrong track and delayed the troubleshooting process. Thus, collecting both passive (*i.e.*, external) and internal observations across multiple standards and technologies is an important prerequisites for purposeful troubleshooting.

Lesson 3: Dynamic changes in the environment matter. Another important lesson we learned is that the surrounding environment has a strong influence on the wireless network and thus should never be underestimated. Environments with moving equipment introduce a strong temporal impact on the wireless propagation conditions, whilst environmental effects such as heat and humidity [27], [28] can degrade the performance of a wireless system and lead to complicated failure scenarios. A system able to detect and incorporate these temporal effects as part of the failure analysis would simplify the engineer's work and save both time and resources. At the same time, the monitoring equipment must be rugged enough to sustain the environmental challenges of the operational area, including dust, heat, frost, moisture, and mechanical shocks [29].

Lesson 4: Cheap off-the-shelf hardware is viable. Finally, we learned that a home-made solution built from inexpensive off-the-shelf hardware components can be advantageous. For instance, using three Raspberry Pi devices we were able to built a distributed traffic capture system that satisfied our needs in the open-pit mine case, and in both failure cases an IEEE 802.15.4 node enabled a sufficiently accurate spectrum analysis. Compared to commercially available measurement equipment, these home-made solutions are cheaper, easier to extend, consume less power, and can be deployed in larger numbers in a distributed fashion.

B. Methodology

While troubleshooting industrial wireless networks including those presented in Sections III-A and III-B, we developed a specific methodology to find the root cause of the failure.

Phase 1: Preparation. In our experience, troubleshooting requires about 0.5 person days (of a wireless networking expert) of preparation, which involves customizing the monitoring system for the failure scenario at hand. Important decisions include whether a distributed or centralized monitoring setup should be used and which wireless standards and technologies are expected to be present in the area under investigation.

Phase 2: On-site analysis. This is the most important phase and typically requires from 0.4 to 1.6 expert person days. It consists of six subsequent steps, as described in the following, along with a breakdown of the approximate working time.

- **2.1** *Observation* of the network under study to gain in-depth knowledge of the environment while focusing on physical defects of the network equipment (**13 %**).
- **2.2** *Deployment* of the monitoring system, which includes identifying appropriate observation position(s) for the monitoring node(s) and time-synchronizing the nodes in case of a distributed setup (**7 %**).
- **2.3** *Structure analysis* using a black-box approach, which includes spectral analysis to detect CTI and revealing the topology of all networks in reception range (**32%**).
- **2.4** *Passive link quality analysis* of all links between nodes as an indication for possible failure causes (**24%**).
- **2.5** *Interference analysis* based on spectrum measurements or analyzing the failure pattern in packet traces (**8%**).
- **2.6** *Correlation* of all available information to derive the root cause of the failure (**16%**).

Phase 3: Off-line analysis. This phase is needed if the failure was not found on-site, which is typically due to one of the two following reasons. First, unsuitable observation position(s) have been chosen, resulting in incomplete or erroneous data

recorded by the monitoring node(s). Preprocessing algorithms may partially solve this problem, thereby improving all subsequent failure analysis algorithms. If preprocessing does not help, the complete on-site analysis must be redone. Second, the failure scenario and/or the environmental conditions are extremely challenging, demanding the development of new analysis algorithms. For example, developing new packet analysis scripts typically takes from 0.5 to 2 expert person days in our experience. Afterwards, the failure analysis may proceed with any of the steps 2.3–2.6 above.

V. Troubleshooting System for Industrial Wireless Networks: Requirements and Design

We now derive from the lessons we learned the requirements for a failure analysis system that can implement our methodology, and present a corresponding system architecture that is tailored to troubleshooting IIoT coexistence problems. We also describe a modular hardware platform we designed and currently use for a prototypical implementation of our proposed failure analysis system architecture.

A. Requirements of Failure Analysis System

A failure analysis system must satisfy 5 key requirements:

- **Automated:** Steps 2.3–2.6 above can and should be automated by implementing a blackbox approach without requiring manual interventions. As a result, the system reduces the amount of expert knowledge required and speeds up the troubleshooting process considerably.
- **Interactive:** On the other hand, failure analysis may require interacting with the environment, such as repositioning an object or reconfiguring the network. Thus, online processing and concurrent analysis algorithms are needed to allow for an interactive observation of the continuous data stream from all monitoring devices.
- **Comprehensive:** The system should detect all coexisting standards and technologies. This includes estimating the location of all signal sources and recovering the full network topology, including the role of each node. Cross-technology observations and analysis should be performed on multiple network layers. Temporal effects on wireless propagation (*e.g.*, due to moving parts and mobile equipment) should also be taken into account.
- **Flexible:** New standards and technologies appear frequently, leading to a constantly increasing fault tree that needs to be investigated. Thus, a flexible and extensible hardware and software design based on a generic multi-stage analysis structure is essential. The possibility to reconfigure the monitoring hardware and to modify the analysis pipeline is a plus. The system should support a single-node and a multi-node (*i.e.*, distributed) setup.
- **User-friendly:** To allow use by non-experts, the system should be easy to set up and configure, and should present the results of the analysis in an intuitive way.

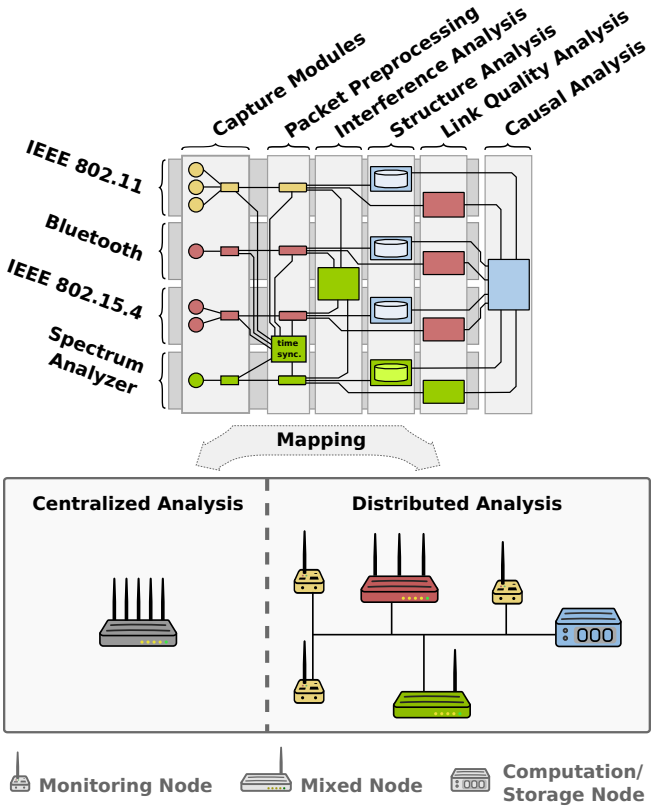

Figure 6. Architecture of the proposed failure analysis system for IIoT coexistence problems, and mapping onto centralized and distributed setup.

B. Architecture of Failure Analysis System

Based on these requirements, we detail a concrete hardware and software architecture for an IIoT failure analysis system.

Global structure. Figure 6 shows the system architecture. We partition the data flow into multiple horizontal planes. Each horizontal plane represents one communication standard (*e.g.*, IEEE 802.11, Bluetooth, IEEE 802.15.4) or observation parameter (*e.g.*, spectrum analysis) and hence one class of transceiver module. The vertical planes are derived from the system requirements and the analysis stages in Section IV-B.

Analysis stages. The first stage is the *Capture Module*, featuring a high sampling rate and a source-specific data format. A high-performance implementation of an information filter is the main prerequisite to tailor the analysis system to an embedded hardware platform. Moreover, the capture module enforces a unified data structure across all horizontal planes.

The *Packet Preprocessing* stage is mainly responsible for cleaning up the dataset and generating a tidy data structure. This includes the reconstruction of corrupted information, a high-precision time synchronization of all data sources, and structuring the datasets to facilitate further analysis.

The aim of the *Interference Analysis* stage is to extract, join, and process CTI-related information. It correlates the time-synchronized data streams of all capture modules to

detect packet collisions and all sources of interference.

The *Structural Analysis* stage leverages a database of observations to automatically reconstruct the network topology, including the role of each node in the network, and estimate the location of all signal sources. Hence, this stage plays a major role in our proposed black-box analysis.

The *Link Quality Analysis* stage utilizes the pre-processed information to estimate various characteristic link quality parameters and determines the health status of the network.

The *Causal Analysis* is the final stage. It takes into account the output of all prior analysis stages across all horizontal planes using static decision trees as well as supervised and unsupervised learning algorithms, which facilitates presenting the failure analysis results in an intuitive way to the user.

Building blocks. Each stage of the data flow architecture is based on slender analysis modules within the horizontal plane of a transceiver module. These building blocks are tailored to one specific task and are interconnected in a flexible manner to enable a fast reconfiguration of the analysis architecture. In this way, we can customize the troubleshooting system for the failure scenario at hand, perform on-site upgrades of the analysis structure, and easily extend the software system with regards to upcoming wireless communication systems. By specifying a well-defined interface between all vertical planes it is possible to decouple the development of analysis modules. To enable rapid prototyping, suitable software libraries and programming languages (*e.g.*, Python, R) may be used to implement a module. Even though the architecture primarily targets on-line analysis, storage modules may be introduced to save intermediate results for a subsequent off-line analysis.

Interconnection system. In addition to the block-based software structure, we propose a flexible networked interconnection system that makes it possible to easily map a failure analysis to a single monitoring node or to create a distributed failure analysis system. The latter allows the presence of multiple monitoring and computing/storage nodes that collaborate within one analysis task, as shown in Figure 6.

Hardware mapping. The mapping procedure must incorporate platform-specific dependencies within a heterogeneous distributed hardware architecture. Capture and packet preprocessing modules are directly bound to the monitoring node that includes the appropriate transceiver module. The hardware requirements of these monitoring nodes may be tailored towards filtering and pre-processing data on a per-packet base, which makes it possible to use commercial off-the-shelf hardware (*e.g.*, wireless sensor nodes or APs running OpenWRT). Algorithmic analysis modules may require more processing and storage resources and hence must be mapped onto more powerful platforms, such as a multicore single board computer (SBC) with a mass storage device.

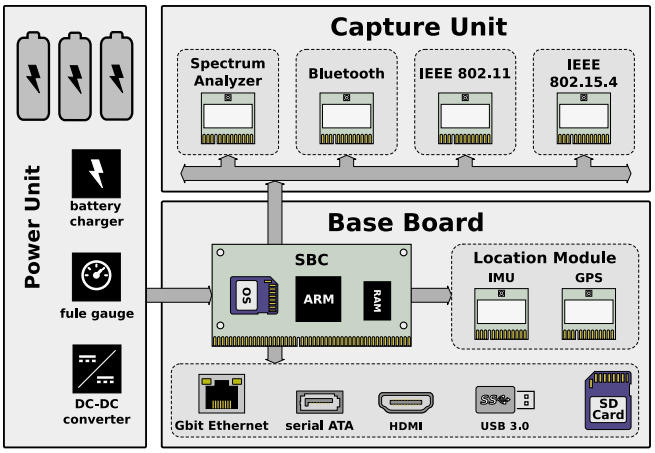

Figure 7. Overview of the proposed modular hardware platform.

C. Modular Hardware Platform

We now sketch a concrete hardware platform we have built in order to implement the aforementioned system architecture. The platform enables both an affordable implementation of mobile battery-powered failure analysis instruments and an integration into a permanent observation system.

In accordance with the requirements in Section V-A, we use a modular hardware platform that is easy to adapt and extend. Figure 7 provides an overview of the modular platform. It is based on affordable off-the-shelf components and consists of three separate modules: (*i*) a base board including an SBC of compact form factor, external interfaces (*e.g.*, Ethernet, HDMI, USB), a mass storage device to record all intermediate results for off-line analysis, and an indoor/outdoor location tracking module; (*ii*) a power unit with a battery lifetime of at least 5 hours; and (*iii*) a capture module to plug in standard transceiver modules (*e.g.*, via PCI Express Mini Card). The platform has a hand-held form factor and is enclosed by a rugged metal casing for use in harsh industrial environments.

D. Prototype

As of today, we have developed and commissioned a first prototype of the proposed hardware platform including the operating system and its device drivers. We accomplished an intensive performance evaluation with satisfactory evaluation results. Currently, a final revision including minor upgrades with an industrial-suited enclosure is developed. At the same time, we implemented the interconnection system for the modular building block based system architecture. So far, the communication framework is finished and tested on an IA-64 system. We implemented an IEEE 802.11 capture module in C++ and a corresponding packet preprocessing module in Python. In addition, we created a prototype for the structure analysis module in Python.

In a next step, we will finalize the remaining stages of the Wi-Fi plane. Afterward, we will test the performance of this

plane on our ARM-based hardware platform. We also intend to extend our analysis framework by adding all required modules for the other wireless communication standards in order to enable an improved and user-friendly troubleshooting process of coexisting wireless IIoT systems.

VI. Related Work

Wireless coexistence is a known problem in communication networks and attracted a large body of work. Many researchers have studied the coexistence problems in the ISM frequencies used by common IoT devices [9], especially in the crowded 2.4 GHz band [17], [30]. Most of these studies have focused on the coexistence among two wireless technologies, typically between IEEE 802.11 and IEEE 802.15.4 [31]–[33] or between IEEE 802.15.1 and IEEE 802.15.4 [34]. Others have analyzed how application-specific technologies can tolerate interference, e.g., body area networks [35]–[37] and industrial systems based on Wireless HART [38]. All these works highlight how strongly co-existing devices may affect the performance of a wireless networks, but specifically focus on a few exemplary technologies and do not provide solutions on how to enable coexistence in general.

Bluetooth was one of the first systems trying to minimize its interference on coexisting wireless technologies. Adaptive frequency hopping was introduced to dynamically change the hopping sequence of Bluetooth devices, thereby allowing other systems such as Wi-Fi networks to use the remaining frequency channels [9]. Although choosing orthogonal channels is a very simple but effective solution to enable coexistence, it becomes rather ineffective when multiple networks in the surroundings cover the whole ISM band.

Another body of works has studied how to *actively* ensure coexistence of heterogeneous wireless systems. Gummadi et al. [39] have proposed a coordination mechanism that uses multi-technology gateways and an expressive policy language to ensure coexistence between Wi-Fi, Bluetooth and IEEE 802.15.4 devices. An alternative approach consists in using indirect coordination by transmitting busy tones or special carrier signaling pulses for preventing another wireless technology to interfere [40]–[42] or by modulating the payload length to encode channel access parameters [43]. FreeBee [44] is an example of indirect cross-technology communication among Wi-Fi, ZigBee, and Bluetooth obtained by shifting the timing of periodic beacon frames without incurring extra traffic. Similarly, other works have aimed to establish a communication link between different wireless technologies to ensure coexistence, especially between Wi-Fi and IEEE 802.15.4 devices [45]–[48]. Most of these works, however, target two or three specific technologies (typically Wi-Fi and IEEE 802.15.4), but do not provide a generic solution to the coexistence problem in a given ISM band. Nevertheless, all these efforts show an increasing trend to ensure that the wireless networking standards operating in the same frequency band contain mechanisms to actively communicate and prevent causing harmful interference to each other, an objective shared by the newly-established IEEE 802.19 coexistence working group [49].

Despite these increasing efforts in ensuring co-existence between wireless technologies, the vast majority of wireless devices is deployed in the wild and still interferes with each other in an "anarchic and arbitrary manner" [39], causing a large number of network and deployment failures. For this reason, troubleshooting of wireless networks has been a hot research topic. Several solutions have been proposed by the wireless sensor networks community to simplify network debugging, the majority of them addressing passive monitoring of packets [50] to verify the health of the network [51], to reconstruct network dynamics [52], or to infer complexity bugs [53] and root causes of abnormal phenomena [54]–[56]. All these systems, however, analyze local traffic from the deployed sensor networks and lack information about the influence of co-located devices using other technologies, practically making it difficult to debug and study actual coexistence problems.

Similar problems exist outside the wireless sensor networks community: a number of works has, for example, proposed monitoring and debugging systems for Wi-Fi. Cheng et al. [57] have proposed Jigsaw, a system using multiple monitors to provide a unified view of physical, link, network and transport-layer activity. Sheth et al. [58] have proposed fine-grained detection algorithms that are capable of distinguishing between root causes of wireless anomalies at the depth of the physical layer of IEEE 802.11 systems. Finally, RFDump [59] is a software architecture for monitoring packets on heterogeneous wireless networks running on off-the-shelf (but expensive) software-defined radios.

In contrast to this body of literature, our work proposes a low-cost battery-powered system for failure analysis tailored to an automated and cross-technology troubleshooting of wireless coexisting systems. Based on our experience and the lessons we have learned in the time-consuming process of troubleshooting wireless coexistent networks, we also embedded specific features tailored towards a user-friendly and simplified observation of multiple communication standards and unknown sources of interference in the proposed device. We believe that the proposed system can augment several of the aforementioned debugging tools, simplifying and speeding-up troubleshooting in industrial wireless networks.

VII. Conclusions

Troubleshooting a failure is an almost unavoidable task during the lifetime of a real-world wireless network. CTI caused by the increasing number of wireless technologies and devices within the unlicensed ISM frequency bands increases the complexity of this task, making it difficult to identify and eliminate failures, especially in safety-critical environments.

In this paper, we have shared our experiences by reporting on two exemplary cases and the lessons we learned from

troubleshooting industrial wireless networks. We propose a generic system architecture and hardware platform that enables a systematic, automated troubleshooting of wireless coexistence problems in the IIoT. One of the key features of the proposed system is its ability to simplify the observation of heterogeneous wireless communication standards and unknown sources of radio interference in the surroundings.

This work represents a first step towards simplifying the troubleshooting of coexistence problems in the IIoT. We expect the research community to follow up on this work in the near future and propose novel systems to predict connectivity and automatically debug safety-critical wireless networks in the presence of coexisting wireless technologies. At the same time, we anticipate an increasing effort in designing strategies aiming to ensure cross-technology coexistence by means of active/passive communication as well as distributed coordination among heterogeneous wireless technologies.

Acknowledgments. This work was supported by the project "fast automation" and the Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF) within the initiative "Region Zwanzig20" under project number 03ZZ0510A, and by the German Research Foundation (DFG) within the Cluster of Excellence "Center for Advancing Electronics Dresden" (CFAED).

REFERENCES

[1] M. Fitzgerald, "Wireless – like magic," *Computerworld*, vol. 27, no. 10, 1993.

[2] Ericsson AB, "Ericsson mobility report – on the pulse of the networked society," 2015.

[3] Gartner, "Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015," Press Release, 2015.

[4] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. of the* 47^{th} *Design Automation Conf. (DAC)*, 2010.

[5] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi interference in low power ZigBee networks," in *Proc. of the* 8^{th} *Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2010.

[6] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the RF smog: Making 802.11 robust to cross-technology interference," in *Proc. of the ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2011.

[7] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions of Sensor Networks*, vol. 8, no. 4, 2012.

[8] G. Zhou, J. A. Stankovic, and S. H. Son, "Crowded spectrum in wireless sensor networks," in *Proc. of the* 3^{rd} *Workshop on Embedded Networked Sensors (EmNets)*, 2006.

[9] C. A. Boano and K. Römer, "External radio interference," in *Radio Link Quality Estimation in Low-Power Wireless Networks*, ser. SpringerBriefs in Electrical and Computer Engineering - Cooperating Objects, 2013.

[10] P. Suriyachai, J. Brown, and U. Roedig, "Time-critical data delivery in wireless sensor networks," in *Proc. of the* 6^{th} *IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2010.

[11] A. Frotzscher, U. Wetzker, M. Bauer, M. Rentschler, M. Beyer, S. Elspass, and H. Klessig, "Requirements and current solutions of wireless communication in industrial automation," in *Proc. of the IEEE Int. Conf. on Communications Workshops (ICC)*, 2014.

[12] O. Chipara, C. Lu, T. C. Bailey, and R. Gruia-Catalin, "Reliable clinical monitoring using WSNs: Experiences in a step-down hospital unit," in *Proc. of the* 8^{th} *Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2010.

[13] J. Jeong, "Wireless sensor networking for intelligent transportation systems," Ph.D. dissertation, University of Minnesota, MN, USA, 2009.

[14] A. Kamerman and N. Erkocevic, "Microwave oven interference on wireless LANs operating in the 2.4 GHz ISM band," in *Proc. of the* 8^{th} *IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications (PIRMC)*, vol. 3, 1997.

[15] Hewlett-Packard, "Coexistence in 2.4 & 5.8 GHz ISM Bands Backgrounder," Tech. Rep. 41.3.9, 2003.

[16] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proc. of the* 10^{th} *IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2011.

[17] M. Woehrle, M. Bor, and K. Langendoen, "868 MHz: a noiseless environment, but no free lunch for protocol design," in *Proc. of the* 9^{th} *Int. Conf. on Networked Sensing Systems (INSS)*, 2012.

[18] B. Kusy, C. Richter, W. Hu, M. Afanasyev, R. Jurdak, M. Brünig, D. Abbott, C. Huynh, and D. Ostry, "Radio diversity for reliable communication in WSNs," in *Proc. of the* 10^{th} *IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2011.

[19] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," in *Proc. of the* 6^{th} *Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2008.

[20] "Google loon: Balloon-powered internet for everyone," https://www.google.com/loon/.

[21] J. Winter, I. Muller, G. Soatti, S. Savazzi, M. Nicoli, L. Becker, J. Netto, and C. Pereira, "Wireless coexistence and spectrum sensing in industrial Internet of Things: An experimental study," *Int. Journal of Distributed Sensor Networks*, 2015.

[22] ZVEI Automation - German Electrical and Electronic Manufacturer's Association, "Coexistence of Wireless Systems in Automation Technology," 2009.

[23] G. Koepke, W. Young, J. Ladbury, and J. Coder, "Complexities of testing interference and coexistence of wireless systems in critical infrastructure," National Institute of Standards and Technology (NIST), Tech. Rep. 1885, 2015.

[24] M. Ullmann, S. Hoener, A. Frotzscher, U. Wetzker, I. Splitt, and M. Galetzka, "Emulation platform for coexistence analysis in wireless automation," in *Proc. of the European Microwave Conf. (EuMC)*, 2013.

[25] Siemens, "SIMATIC HMI, HMI Device, Mobile Panel 277 IWLAN, Getting Started," 2008.

[26] Guideline VDI/VDE 2185, *Radio based communication in industrial automation*, The VDI/VDE Society for Measurement and Automatic control, 2009.

[27] C. A. Boano, H. Wennerström, M. A. Zúñiga, J. Brown, C. Keppitiyagama, F. J. Oppermann, U. Roedig, L.-Å. Nordén, T. Voigt, and K. Römer, "Hot Packets: A systematic evaluation of the effect of temperature on low power wireless transceivers,"

in *Proc. of the* 5^{th} *Extreme Conf. on Communication (ExtremeCom)*, 2013.

[28] C. A. Boano, K. Römer, and N. Tsiftes, "Mitigating the adverse effects of temperature on low-power wireless protocols," in *Proc. of the* 11^{th} *Int. Conf. on Mobile Ad hoc and Sensor Systems (MASS)*, 2014.

[29] J. Beutel, B. Buchli, F. Ferrari, M. Keller, and M. Zimmerling, "X-SENSE: Sensing in extreme environments," in *Proc. of Design, Automation and Test in Europe (DATE)*, 2011.

[30] D. Yang, Y. Xu, and M. Gidlund, "Wireless coexistence between ieee 802.11- and ieee 802.15.4-based networks: A survey," *Int. Journal of Distributed Sensor Networks (IJDSN)*, vol. 2011, no. 912152, 2011.

[31] M. Petrova, L. Wu, P. Mähönen, and J. Riihijärvi, "Interference measurements on performance degradation between colocated ieee 802.11g/n and ieee 802.15.4 networks," in *Proc. of the* 6^{th} *Int. Conf. on Networking (ICN)*, 2007.

[32] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Experimental study of coexistence issues between ieee 802.11b and ieee 802.15.4 wireless networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, 2008.

[33] J. Winter, I. Muller, C. Pereira, S. Savazzi, L. Becker, and J. Netto, "Coexistence issues in wireless networks for factory automation," in *Proc. of the* 12^{th} *IEEE Int. Conf. on Industrial Informatics (INDIN)*, 2014.

[34] A. Sikora and V. F. Groza, "Coexistence of IEEE 802.15.4 with other systems in the 2.4 ghz-ISM-band," in *Proc. of the IEEE Conf. on Instrumentation and Measurement Technology (IMTC)*, 2005.

[35] F. Martelli and R. Verdone, "Coexistence issues for wireless body area networks at 2.45 ghz," in *Proc. of the* 18^{th} *European Wireless Conf. (EW)*, 2014.

[36] J.-H. Hauer, V. Handziski, and A. Wolisz, "Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks," in *Proc. of the* 6^{th} *European Conf. on Wireless Sensor Networks (EWSN)*, 2009.

[37] S. Seidman and N. LaSorte, "An experimental method for evaluating wireless coexistence of a Bluetooth medical device," *IEEE Electromagnetic Compatibility Magazine*, vol. 3, no. 3, 2014.

[38] C. M. D. Dominicis, P. Ferrari, A. Flammini, E. Sisinni, M. Bertocco, G. Giorgi, C. Narduzzi, and F. Tramarin, "Investigating WirelessHART coexistence issues through a specifically designed simulator," in *Proc. of the Int. Instrumentation and Measurement Technology Conf. (I2MTC)*, 2009.

[39] R. Gummadi, H. Balakrishnan, and S. Seshan, "Metronome: Coordinating spectrum sharing in heterogeneous wireless networks," in *Proc. of the* 1^{st} *Int. Conf. on Communication Systems and Networks (COMSNETS)*, 2009.

[40] X. Zhang and K. G. Shin, "Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi," in *Proc. of the* 12^{th} *ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2011.

[41] ——, "Cooperative carrier signaling: Harmonizing coexisting WPAN and WLAN devices," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, 2013.

[42] ——, "Gap sense: Lightweight coordination of heterogeneous wireless devices," in *Proc. of the IEEE Conf. on Computer Communications (INFOCOM)*, 2013.

[43] D. Croce, N. Galioto, D. Garlisi, C. Giaconia, F. Giuliano, and I. Tinnirello, "Demo: Unconventional wifi-zigbee communications without gateways," in *Proc. of the* 9^{th} *ACM Int. Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*, 2014.

[44] S. M. Kim and T. He, "FreeBee: Cross-technology communication via free side-channel," in *Proc. of the* 21^{st} *Int. Conf. on Mobile Computing and Networking (MobiCom)*, 2015.

[45] S. Yin, Q. Li, and O. Gnawali, "Interconnecting WiFi devices with IEEE 802.15.4 devices without using a gateway," in *Proc. of the* 11^{th} *Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2015.

[46] R. Lim, M. Zimmerling, and L. Thiele, "Passive, privacy-preserving real-time counting of unmodified smartphones via ZigBee interference," in *Proc. of the* 11^{th} *Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2015.

[47] K. Chebrolu and A. Dhekne, "Esense: Communication through energy sensing," in *Proc. of the* 15^{th} *Int. Conf. on Mobile Computing and Networking (MobiCom)*, 2009.

[48] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to share: Narrowband-friendly wideband networks," in *Proc. of the ACM Conf. on Data Communication (SIGCOMM)*, 2008.

[49] *IEEE Standard for Local and Metropolitan Area Networks - Part 19: TV White Space Coexistence Methods*, IEEE std 802.19.1-2014 ed., IEEE 802.19 Working Group, May 2014.

[50] M. Ringwald, K. Römer, and A. Vitaletti, "Passive inspection of sensor networks," in *Proc. of the* 3^{rd} *IEEE Int. Conf. Distributed Computing in Sensor Systems (DCOSS)*, J. Aspnes, C. Scheideler, A. Arora, and S. Madden, Eds., 2007.

[51] S. Rost and H. Balakrishnan, "Memento: A health monitoring system for wireless sensor networks," in *Proc. of the* 3^{rd} *IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (MASS)*, vol. 2, 2006.

[52] B. Chen, P. Geoffrey, G. Mainland, and M. Welsh, "LiveNet: Using passive monitoring to reconstruct sensor network dynamics," in *Proc. of the* 4^{th} *IEEE Int. Conf. Distributed Computing in Sensor Systems (DCOSS)*, 2008.

[53] M. M. H. Khan, H. K. Le, H. Ahmadi, T. F. Abdelzaher, and J. Han, "Dustminer: Troubleshooting interactive complexity bugs in sensor networks," in *Proc. of the* 6^{th} *ACM Conf. on Embedded Network Sensor Systems (SenSys)*, 2008.

[54] K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, "Passive diagnosis for wireless sensor networks," in *Proc. of the* 6^{th} *Int. Conf. on Embedded Network Sensor Systems (SenSys)*, 2008.

[55] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proc. of the* 3^{rd} *Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2005.

[56] R. Jurdak, A. G. Ruzzelli, A. Barbirato, and S. Boivineau, "Octopus: Monitoring, visualization, and control of sensor networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 8, 2011.

[57] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proc. of the ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2006.

[58] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "Mojo: A distributed physical layer anomaly detection system for 802.11 wlans," in *Proc. of the* 4^{th} *Int. Conf. on Mobile Systems, Applications and Services (MobiSys)*, 2006.

[59] K. Lakshminarayanan, S. Sapra, S. Seshan, and P. Steenkiste, "RFDump: An architecture for monitoring the wireless ether," in *Proc. of the* 5^{th} *Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT)*, 2009.