Reliable Communications for the Industrial Internet of Things

Vasileios Kotsiou

▶ **To cite this version:**

Vasileios Kotsiou. Reliable Communications for the Industrial Internet of Things. Other [cs.OH]. Université de Strasbourg, 2020. English. NNT : 2020STRAD017 . tel-03616113

HAL Id: tel-03616113
https://theses.hal.science/tel-03616113

Submitted on 22 Mar 2022

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

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

UNIVERSITÉ DE STRASBOURG

ÉCOLE MATHEMATIQUES, SCIENCES DE

L'INFORMATION ET DE L'INGÉNIEUR

Laboratoire ICube - UMR 7357

THÈSE présentée par :

Vasileios KOTSIOU

soutenue le : **25 septembre 2020**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

Reliable Communications for the Industrial Internet of Things

THÈSE dirigée par :

M. THEOLEYRE Fabrice	Chargé de recherche, CNRS

RAPPORTEURS :

M. HEUSSE Martin	Professeur, Grenoble INP
Mme. MITTON Nathalie	Directrice de recherche, INRIA

AUTRES MEMBRES DU JURY :

M. WATTEYNE Thomas	Directeur de recherche, INRIA Paris
M. MONTAVONT Nicolas	Professeur, IMT Atlantique
M. PAPADOPOULOS Georgios Z.	Maitre de Conférence, IMT Atlantique

Université de Strasbourg

École doctorale MSII

Reliable Communications for the Industrial Internet of Things

THÈSE

présentée pour l'obtention du

Doctorat de l'Université de Strasbourg

(mention informatique)

par

Vasileios KOTSIOU

Composition du jury

Directeur de thèse :	Dr. Fabrice THEOLEYRE, CNRS, Université de Strasbourg
Co-encadrant de thèse :	Dr. Georgios Z. PAPADOPOULOS, IMT Atlantique Prof. Periklis CHATZIMISIOS, IHU, Thessaloniki, Grèce
Rapporteurs :	Prof. Martin HEUSSE, Grenoble INP Dr. Nathalie MITTON, INRIA
Examinateur :	Dr. Thomas WATTEYNE, INRIA Paris Prof. Nicolas MONTAVONT, IMT Atlantique

Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie
ICube — UMR 7357

To my twin sons Giorgios-Marios and Nikolaos

Acknowledgments

A long journey to complete this thesis has come to an end. During this trip, the most important thing in my life happened, the birth of my twin sons, Georgios-Marios and Nicholas, which filled me with immense joy. Their upbringing and my work for this thesis was a very difficult task. For this reason, I would like to thank my wife Eleni for her patience and for the support she showed me.

I would also to thank my supervisors Dr. Fabrice Theoleyre, Dr. Georgios Z. Papadopoulos, and Prof. Periklis Chatzimisios, for believing in my potential and for guiding me during my thesis. Thank you for your guidance, patience, constant support, and most importantly, your friendship during these years.

I would also want to thank Prof. Martin Heusse, Dr. Nathalie Mitton, Dr. Thomas Watteyne, and Prof. Nicolas Montavont for accepting to be part of my jury, for their questions, and for their useful suggestions. Your constructive comments helped me improve this manuscript.

Special thanks also go to my co-author Dimitrios Zorbas, for his valuable help and collaboration.

I would like to thank very much my brother Theofanis which supported me all this time in many different ways. Finally, I would like to thank my parents Georgios and Hermione for encouraging and supporting me to embark on this demanding and difficult journey of knowledge.

Abstract

Wireless communications are very popular and enabled an Internet access for any user (and thing). Indeed, during the last years, we have experienced the emergence of a new paradigm called Internet of Things (IoT) in which smart, uniquely identifiable, and connected objects cooperatively construct a (wireless) network of things. Those things can be deployed nearly everywhere, at homes, universities, cities, agricultural fields, even in human bodies. The Industrial Internet of Things (IIoT) is an emerging concept aiming at re-using the IoT mechanisms to make the production chains more profitable by maximizing flexibility and adaptability in the factories. However, industrial applications require often deterministic communications as well as end-to-end reliability close to 100%. Unfortunately, wireless communications mean also contention for the medium access. Moreover, a plethora of wireless devices may use the same unlicensed band, generating a large volume of interference. To address these requirements, the Time-Slotted Channel Hopping (TSCH) mode of the IEEE 802.15.4-2015 standard proposed to schedule the transmissions to avoid collisions, and a slow channel hopping technique to combat external interference. However, we still have to tackle several challenges to provide high reliability in any condition and to respect strict end-to-end delay constraints in multi-hop topologies. Thus, the main scope of this thesis was to propose the mechanisms to achieve the previously mentioned goals. In this thesis, we first conducted a series of experiments to characterize the IEEE 802.15.4 radio channels in an indoor testbed. We demonstrated in particular the existence of specific per-link characteristics, where external interference may be locally high for some radio channels. Thus, we proposed to improve the efficiency of the slow channel hopping technique with blacklisting techniques. The objective is to exclude from the channel hopping sequence the low-quality channels. First, we proposed a distributed blacklisting technique, that adopts a pseudo-random approach to avoid using the worst radio channels. While this approach allows each radio link to decide autonomously the best radio channels to use, some collisions may still arise pseudo-randomly. Therefore, we then proposed a centralized blacklisting scheme, able to adapt the blacklists for each radio link, while still making the full behavior deterministic, by re-arranging the conflicting blacklists. We also extended a hybrid blacklisting scheme that exploits the full radio spectrum, assigning all the channel offsets to increase the network efficiency when handling long blacklists. Finally, we proposed a scheduling function that aims to meet the requirements of IIoT for low latency and high reliability in a network with radio links subjected to external interference. With the contributions presented in this dissertation, we provide the IIoT with the appropriate tools to achieve reliable communications even in harsh industrial environments with multi-path fading, potential interference, and obstacles.

List of publications

The contributions of this thesis have been published or submitted in peer-reviewed international and national journals and conferences.

International Journals and Magazines

- *"LDSF: Low-latency Distributed Scheduling Function for Industrial Internet of Things"*
  Kotsiou, V., Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
  In IEEE **Internet of Things Journal**, vol. 7, no. 9, pp. 8688-8699, Sept. 2020, doi:10.1109/JIOT.2020.2995499.

- *"Whitelisting without Collisions for Centralized Scheduling in Wireless Industrial Networks"*
  Kotsiou, V., Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
  In IEEE **Internet of Things Journal**, vol. 6, no. 3, pp. 5713-5721, June 2019, doi: 10.1109/JIOT.2019.2905217.

- *"Blacklisting-Based Channel Hopping Approaches in Low-Power and Lossy Networks"*,
  Kotsiou, V, Papadopoulos, G. Z. Zorbas, D., Chatzimisios, P., & Theoleyre, F.
  In IEEE **Communications Magazine**, vol. 57, no. 2, pp. 48–53, February 2019, doi: 10.1109/MCOM.2018.1800362.

International Conferences and Workshops

- *"Adaptive Multi-Channel Offset Assignment for Reliable IEEE 802.15.4 TSCH Networks"*
  Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P.,& Théoleyre, F.
  In *2018 Global Information Infrastructure and Networking Symposium (**GIIS**)* (pp. 1-5). IEEE, Thessaloniki, October 2018, pp. 1-5, doi: 10.1109/GIIS.2018.8635751.

- *"Label: Link-based adaptive blacklisting technique for 6tisch wireless industrial networks"*
  Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
  In *Proceedings of the 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (**MSWiM**)* (pp. 25-33), ACM, Miami, November 2017, pp. 25–33, doi: 10.1145/3127540.3127541.

- *"Is local blacklisting relevant in slow channel hopping low-power wireless networks?"*
  Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
  In 2017 *International Conference on Communications (**ICC**)* (pp. 1-6), IEEE, Paris, May 2017, pp. 1-6, doi: 10.1109/ICC.2017.7996980.

Contents

Chapter 1

Introduction and Context

1.1 Wireless Sensor Networks

One revolutionary technology that emerged during the Cold War was Wireless Sensor Network (WSN) [Jin18]. In the 1950s, the United States Military deployed the Sound Surveillance System (SOSUS) that detected and tracked the quiet Soviet submarines using acoustic sensors. Today, WSN's applications have expanded beyond military applications to environmental, health, home, urban, and industrial usage. Thus, WSNs became one of the fundamental technologies towards the materialization of Mark Weiser's vision for Ubiquitous Computing [Wei91].

A set of wireless sensor nodes can form a network called WSN. Wireless sensor nodes are resource-constrained devices, which means that they have limited transmission power, memory capacity, processing power and energy (battery-powered). Furthermore, deployments of WSNs are usually dense, and in many cases, communication is carried out through multiple hops and possibly via lossy wireless links subjected to external interference. As a result, WSNs can be considered as Low-power and Lossy Network (LLN).

In a typical WSN application, a collection of WSN nodes (or motes) senses the natural environment and encapsulates the measurement(s) in a data packet. This data packet is then received by other WSN nodes that forward it to a sink, which is in charge of collecting the data packets. Thus, a WSN node may generate or process data packets. It also may forward packets to and from the sink, and aggregate them to reduce the volume of data to be transferred.

We commonly identify three different traffic patterns within a WSN:

- *Multipoint-to-point or convergecast*: The nodes send their data packets towards the sink;
- *Point-to-multipoint*: The sink sends data (such as commands, software updates, information to join the network, etc.) to the source nodes;
- *Point-to-point*: A node sends data to other nodes (e.g., control loop).

The source nodes, depending on the application they run, send data to the sink: *i)* periodically in constant time intervals (Constant Bit Rate (CBR), time-driven), *ii)* sporadically after the nodes detect an event (event-triggered, event-driven), *iii)* reactively, in response to a query they received (query-driven).

The evolution of WSNs in order to be connected to the Internet expanded their scope by making them an integral part of the Internet of Things (IoT).WSNs became a key enabler technology for IoT since they behave as an interface between the physical and the digital worlds and augment the awareness of the environment.

1.2 Internet of Things

In the last decades, we have witnessed the huge penetration of the Internet in people's daily lives and activities, where it has radically changed the way people work, have fun, communicate, get informed, get educated, and so on. The purpose of the Internet is to connect computing devices such as PCs, supercomputers, tablets, smartphones, etc. In recent years, a new concept in communications, the *Internet of Things*, has come to the fore. The IoT concept refers to uniquely identifiable and connected objects that form a wireless network of things [Atz+10]. These smart objects are embedded devices with processing, sensing, and communications capabilities. The expansion of the infrastructure and services of the Internet in the world of things, where heterogeneous objects collaborate and communicate seamlessly, fulfills the vision of "anytime/any place connectivity for anything" [Itu].

The expansion of the IoT is rapid, the forecast for 2020 was that the things connected to Internet would be 20 billion [Cis] while the forecast for 2025 is that the number of devices will reach 100 billion.

A huge number of applications relies on the IoT, such as:

- *Environmental applications*: wildlife monitoring [Zha+04], monitoring seismic activity [WA+08], humidity and temperature monitoring [Lan+06], monitor active volcanoes [WA+06], monitoring building's structural health [Xu+04];
- *Health monitoring:* monitoring patient's health [Jon+10; Dag+07] using body sensor networks, assist elderly people in their daily home activities [Sur+12];
- *Urban and home applications*: control traffic [Aro+04], parking assistance [Tan+06];
- *Smart grid*: monitoring the Electrical Distribution System [Lim+10], remote monitoring of wind or solar farms [EKM11], automated fault handling for the prevention of power outages [NK06];
- *Industrial Applications*: precision/smart agriculture [AI+11], real-time monitoring of nuclear power plant [Lin+04], real-time monitoring of industry carbon monoxide [Yan+15]
- *Terrestrial applications*: Measurement of the temperature, the humidity, and the pH of the soil, for more efficient irrigation and the use of pesticides and fertilizers [Bur+04; Cam+07];
- *Emergency rescue*: fire detection in a forest, emergency search and rescue cases such as fire [Sha+06] and earthquake [KL09].

The already pre-existing wireless communication protocols, such as IEEE 802.11 [OP99], are considered inappropriate in the case of WSN since they consist of resource-constrained devices. For this reason, wireless communication protocols tailored to the requirements and to the communication paradigm of the WSN were proposed such as ZigBee [All12] and the standard IEEE 802.15.4 [Ieeb]. The IEEE 802.15.4 standard, defines the operation of physical and Medium Access Control (MAC) layers for Low-Rate Wireless Personal Area Networks (LR-WPANs). The main characteristics of IEEE 802.15.4 are the low power transmission in order to save energy, the maximum transfer rate is 250 kbit/s, the use of Industrial, Scientific and Medical radio bands (ISM) band (2.4 GHz) and the Maximum Transmission Unit (MTU) size is 127 bytes.

A prerequisite for making WSNs an integral part of the IoT is the seamless access of wireless sensors to the Internet. The adoption of the Internet Protocol version 6 (IPv6) as an Internet access protocol is undoubtedly a one-way option due to the vast number of nodes. Still, it raises significant challenges, such as the significant difference in the size of the MTUs between IPv6 and IEEE 802.15.4. Internet Engineering Task Force (IETF) Working Groups (WGs) have standardized several protocols for the LLNs and therefore for WSNs, to address the above challenges. So, the IETF's IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) protocol is responsible for enabling the transportation of IPv6 large sized packets over IEEE 802.15.4's low layers. The topology of a WSN is, in many cases, multihop, so an effective routing

protocol that takes into consideration the unique characteristics of a WSN is required. Thus, IETF ROLL WG standardized IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) as a routing protocol for LLNs, which became the defacto routing protocol for WSNs.

1.3 Industrial Internet of Things

The application of the IoT technologies in industrial processes (Industrial Internet of Things (IIoT)) is a key enabler to the transition to the 4^{th} Industrial Revolution (aka. Industry 4.0) [Wol+17]. By the term Industry 4.0, we mean merging the IoT paradigm with the Cyber-Physical Systems idea [Sis+18]. The IIoT consists of a large collection of wireless sensors and actuators for various industrial applications. Typically, networked control systems comprise sensors, actuators and a controller [Lon+17]. The sensors send their measurements regularly to a controller. According to this feedback, the controller may trigger a reaction by activating some actuators. This control loop has to react in real-time.

However, the communications requirements of the industrial systems (e.g., safety systems, monitoring systems, alerting systems, and information-gathering systems) are the high reliability and the bounded delay, which is a challenging task for the wireless communications of the IIoT.

Wired industrial networks meet the above requirements and also provide high data transfer speeds, are resilient to external interference, and transfer data over long distances. For many years, the industry has accepted widely, the wired protocols such as Highway Addressable Remote Transducer (HART) [Fun06], FieldBus [Tho05], and real-time Ethernet (RTE) [Dec05]. However, wired infrastructures present a high cost of wiring, the cost of their maintenance is high too, especially in extended deployments. The installation of wired infrastructures is challenging in hazardous environments (e.g., flammable, explosive, and hot), and wired infrastructures are unsuitable in case of mobile nodes (e.g., robots, autonomous vehicles, etc.) [Sef+20].

Due to the above drawbacks of wired infrastructures, the Wireless Sensor and Actuator Networks (WSANs) as an integral part of IIoT have gradually begun to replace the wired connections in industrial networks. The benefits of the adoption of IIoT from the Industry can be summarized as follows:

- *Flexible and reconfigurable manufacturing chain*: For instance, industrial robots tend to exploit wireless communications since cables are prone to breakage after a few thousands of flexions [Pap+]. The network topology can also be changed easily to reconfigure the production lines [Wan+16].
- *Cost reduction*: Wireless deployments are cheaper, easier to maintain, and quick to install compared to traditional wired deployments.
- *Safety*: The exploitation of IIoT can enhance the safety of the workers in the industry. For example, the maintenance and reconfiguration of the wireless sensors located in hazardous environments can be done without human intervention.

In a typical IIoT application scenario, all data packets transmitted over the network should be delivered to their destination within strict time limits; in other words, real-time communications are required. The layer of the networking stack, which is charged of achieving the requirement for real-time communications, is the MAC layer.

The MAC layer is responsible for coordinating the wireless nodes' transmissions in the shared wireless medium, specifying how and when a node will attempt to transmit. The primary goal of IIoT's MAC layer is to save energy as its nodes have constrained energy resources, which is achieved by shutting downing the node's radio communications -sleep mode- for long periods. Furthermore, the MAC layer should cope with the causes of wasting energy, which are collisions, idle listening, overhearing, and deafness. At the same time, the MAC layer should aim for a low and deterministic packet delivery delay by orchestrating the transmissions properly in a multihop network. The MAC techniques based on random access (contention based), such as slotted

Aloha and Carrier Sense Multiple Access (CSMA), are inappropriate to meet the requirements of the industrial networks. This class of MAC protocols presents a low performance in terms of reliability, latency when used in dense networks, or when there is heavy data traffic.

The scheduled-based (or synchronized) MAC protocols seem to be the best candidates to meet the requirements of the industrial networks and, at the same time to deal with the constrains of the WSNs. IEEE 802.15.4-2015 [Ieea] oriented on this direction has proposed in 2016, the Timeslotted Channel Hopping (TSCH) mode of the IEEE 802.15.4 MAC layer. The TSCH is based on the proprietary protocols Time Synchronized Mesh Protocol (TSMP) [PD08] which, also inspired, the proprietary protocols, WirellessHART [Spe08], and ISA100.11a [ISAy]. TSCH exploits Time Division Multiple Access (TDMA) to avoid collisions due to simultaneous transmissions from interfering nodes and slow channel hopping to cope with the external interference and mitigate the multi-path fading effect.

The IIoT's MAC protocols, which have been proposed so far, allow the exploitation of IIoT mainly in the non-critical industrial systems. To expand the range of applications of the IIoT in the industry and to materialize the vision of Industry 4.0, the following challenges required to be addressed [Raz+19]:

- **Reliability**: The wireless medium by nature is shared, unreliable, and the radio links quality varies over time. Unreliability is exacerbated in industrial environments due to the reflective metallic surfaces, equipment noise, heat, dust, fading, and electromagnetic interference. An obvious solution to tackle the unreliability is to apply packets' retransmissions at the MAC layer; however, the main drawback of this technique is the increased latency and increased energy consumption.
- **Real-Time Performance**: MAC has to ensure real-time performance. This requirement varies according to the industrial domain, for example, the required latency is 10 ms for safety applications and 100 ms for monitoring applications [WJ16]. Addressing this challenge is further aggravated by unexpected radio link quality degradation, which causes undesired fluctuations in latency.
- **Energy efficiency**: Energy conservation is still a major concern since their sensor nodes are battery-powered, and their battery replacement is not an option in inaccessible industrial areas. However, the demand for high reliability and real-time performance increases energy consumption, making it imperative to find the trade-off between contradictory IIoT's requirements.
- **Scalability**: The MAC should have the flexibility to adapt to changes in the network topology, addition and removal of nodes, and to be able to handle a large number of nodes. However, existing protocols such as TSCH and WirelessHART are based on TDMA, where handling of a large number of nodes and meeting the requirements for Quality of Service (QoS) are particularity challenging.
- **Coexistence and Interoperability**: The concentration of a large number of devices in the same area that may belong to different networks, the use of the over-crowded ISM band, used by a plethora of wireless technologies, is expected to cause interference between devices and, consequently, extensive packets losses. Therefore, it is considered necessary for the devices to detect, classify, and mitigate external interference. In this case, there are two alternatives: each network can mitigate the external interference independently (e.g., by applying a channel hopping technique, Clear Channel Assessment (CCA), blacklisting, etc.) or the co-existing networks have to cooperate to share fairly and efficiently the available bandwidth.

1.4 Motivation

As exposed throughout this chapter, IIoT presents several challenges that are an obstacle towards their widespread adoption from the Industry. More specifically, the most considerable difficulty

is the exploitation of IIoT in critical industrial applications.

The use of the shared wireless medium, and in particular, the use of the 2.4 GHz ISM band, creates significant difficulties in fulfilling the requirement of high reliability. The ISM band is an unlicensed portion of the radio spectrum reserved internationally for industrial, scientific, and medical applications. The ISM band is preferred due to the absence of regulations and ease of the deployment from a wide range of wireless communication technologies. In particular, IIoT suffers because of their low transmission power compared with other wireless communication technologies such as Wi-Fi, due to their constrained energy resources. Thus, co-located wireless networks (e.g., Wi-Fi) with much higher transmitting power, create external interference in IIoT, thus making their radio links unreliable [Pap+17; Pap+16b]. The problem of the external interference is sharpened by the rapid growth of the co-located wireless devices.

One of the most promising techniques for coping with external interference is the slow channel hopping, where the nodes interchange their operating radio channel in a synchronized manner, which is deployed by IEEE 802.15.4-TSCH, WirelessHART, and ISA100.11a. The performance of the slow channel hopping technique can be further improved with the deployment of the blacklisting technique.

The role of a Blacklisting/Whitelisting technique is to evaluate the available radio channels of the channel hopping sequence, to identify the low-reliability radio channels (*'bad'*), to transmit the list of the *bad* radio channels (*blacklist*) to the appropriate nodes, and to exclude the *bad* radio channels from the channel hopping sequence.

Typically the IIoT networks support multi-hop communications due to the low transmission power of the nodes and its large deployment area; moreover, its radio links may be unreliable. Achieving low and bounded end-to-end delay is an essential prerequisite of real-time communications. However, multi-hop IIoT networks depend on the total buffering delay of the data packets in the relay nodes, while minimizing buffering delay is achieved by transmitting data packets immediately after receiving them from the relay nodes. However, in a wireless network with unreliable radio links, retransmission opportunities should be provided for the failed packets' transmissions. At the same time, the retransmission opportunities should be allocated suitably in order not to cause considerable increase in the end-to-end delay. Blacklisting/Whitelisting techniques reduce the end-to-end delay by improving the reliability of radio links and, therefore, the number of retransmissions. Still, they cannot guarantee low and deterministic end-to-end delay. It is, therefore, imperative to propose novel scheduling algorithms that can, at the same time, address the conflicting requirements of end-to-end reliability and of bounded end-to-end delay

1.5 Contributions

The purpose of this dissertation is to provide reliable communications for the IIoT. To support critical industrial applications, we need to make the wireless infrastructure reliable. Indeed, external interference and multi-path fading is a major cause of unreliability. Thus, we have to improve the MAC mechanisms to be more robust in this kind of environments. Hereafter, we list the contributions presented in this dissertation:

To bootstrap our investigation, we conducted a thorough experimental study to characterize the IEEE 802.15.4 radio channels in an indoor testbed. More specifically, we conduct experiments on FIT IoT-LAB [Adj+15] by employing OpenWSN, which is an implementation of the IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) stack. We study in-depth the spatial and temporal characteristics of the radio links and the diversity in radio link's quality among the radio channels. Our objective is here to investigate the relevance of using blacklisting techniques to improve reliability.

Secondly, we introduce a distributed Link-based Adaptive BLacklisting (LABeL) Technique for 6TiSCH Wireless Industrial Networks. LABeL evaluates the quality of the radio channels of each radio link in a distributed manner and exploits a dynamic threshold algorithm to select the best radio channels for the data transmissions. Furthermore, we developed

techniques based on 6P [Wan+18] control packets to ensure blacklist consistency among the transmitter and the receiver of a radio link to avoid deafness. Finally, LABeL introduces a channel hopping modification technique to re-evaluate the low-reliability radio channels without using any control message, thus saving energy.

Third, we propose Whitelist-Aware, Common Whitelist per Timeslot, and Whitelist re-Ordering, new centralized whitelisting/blacklisting techniques to eliminate internal collisions. The main drawback of exploiting a local link-based blacklisting technique is the presence of internal collisions caused by the interfering links scheduled at the same timeslot and using different blacklists. These internal collisions are prejudicial to the reliability, and even worse, may exhibit a repetitive pattern. Thus, we explore in this chapter a dual approach, trying to construct whitelists so that no collision may occur. We investigate how a centralized algorithm may re-organize the whitelists to avoid collisions while still multiplexing the different transmissions through different radio channels for mutually interfering links.

Fourth, we propose a hybrid blacklisting technique (AMABO).
MABO-TSCH [Gom+17] is a well known blacklisting technique, which assigns a collection of channel offsets for each link so that the schedule may be computed centrally, while the blacklist may be constructed locally. However, the MABO approach is not adaptive, and the same amount of radio resources is allocated to each radio link. Thus, in this chapter, we propose AMABO, that allocates dynamically the channel offsets to improve the local adaptability.

Finally, we propose a Low-latency Distributed Scheduling Function (LDSF) for IIoT. LDSF aims to meet the requirements of IIoT for low latency and high reliability even if the radio links are unreliable. Thus, LDSF orchestrates the transmissions opportunities to minimize the time between receiving and transmitting a data packet from the relay nodes of the path towards the sink, with the ultimate goal of improving end-to-end delay. Furthermore, LDSF properly organizes retransmissions, necessary to deal with the radio links' unreliability, so as not to dramatically increase the end-to-end delay and, thus, achieve determinism. Finally, LDSF, based on the periodic traffic pattern of a typical IIoT, allocates transmission opportunities only when traffic expected in order to save energy.

1.6 Structure of the Thesis

This manuscript is organized into eight chapters. The first Chapter presents an introduction to WSN, IoT and IIoT, as well as the motivation and contributions of this manuscript. In Chapter 2, we provide a thorough state-of-the-art on IIoT, and in particular on radio characterization studies, on blacklisting techniques and on scheduling algorithms that aimed to optimize reliability and end-to-end delay. It also gives the reader the necessary elements for understanding the rest of this manuscript. In Chapter 3, we perform a thorough experimental study to characterize the IEEE 802.15.4 radio channels in an indoor testbed, and we investigate the relevance of a blacklisting technique in slow channel hopping low-power wireless networks. In Chapter 4, we present a distributed Link-based Adaptive BLacklisting (LABeL) technique. In Chapter 5, we present three centralized blacklisting/whitelisting techniques to eliminate internal collisions. In Chapter 6, we present AMABO, an extension of the MABO-TSCH that is coping with the low performance of MABO-TSCH into dense and high-traffic networks. In Chapter 7, we present LDSF, a scheduling function that provides high reliability and low and deterministic end-to-end delay at the same time. Finally, Chapter 8 concludes this manuscript by presenting concluding remarks and opening up some perspectives.

Chapter 2

Background and Related Work

As mentioned in the previous chapter, the purpose of this Ph.D. thesis is to provide reliable communications to IIoT networks. The unreliability of communications comes from the use of an unconfined communication medium. The propagation of radio signals can be hampered by various factors that result in degradation in the quality of communications. For this reason, we are conducting an in-depth literature review of existing studies on the determination of the radio link characteristics of a WSN and outline their main observations.

External interference due to numerous overlapping networks that utilize the 2.4 GHz ISM band is responsible for the radio links' unreliability. One widespread technique proposed in the literature for the mitigation of external interference is Channel Hopping or Frequency Hopping. In this technique, transmissions are made by constantly changing/hopping the operating radio channel/frequency so that transmissions cannot be blocked from the exclusive use of a low-quality channel. The proposals of the current thesis are based on the radio channel hopping technique of the IEEE 802.15.4-TSCH protocol and, thus, we describe its main characteristics. Furthermore, we present an overview of 6TiSCH, a standardization process that aims to enable the IPv6 protocol to low-power WSNs.

Utilizing channel hopping techniques bring in a limited gain on the improvement of reliability for a WSN since some channels with high external interference are still used, resulting in many retransmissions or packet losses. Using high-quality channels is the only promising technique to improve reliability. Thus, we perform a thorough literature review of blacklisting/whitelisting techniques that were proposed in the literature. We also classify these blacklisting/whitelisting techniques according to their characteristics and present how they cope with the issues that arise.

Another objective of our research is to reduce the time elapsed between the generation of a data packet from a network node to its delivery by the sink (*end-to-end delay*). Thus, we present the existing scheduling algorithms tailored to 6TiSCH wireless industrial networks that aim to optimize both end-to-end delay and reliability.

2.1 Radio Characterization

One crucial characteristic of the WSNs is the unreliability of their radio links. The main factors that lead to packet loss can be summarized as follows [Bac+12; DSP19]:

- Environment: The multi-path fading effect, signal attenuation, and noise;
- Interference: It is caused by the simultaneous use of the same part of the radio spectrum by wireless networks or other devices.

Therefore, the deep understanding of the radio spectrum characteristics is the first step in proposing solutions that provide high-reliability to WSNs to be used in industrial applications.

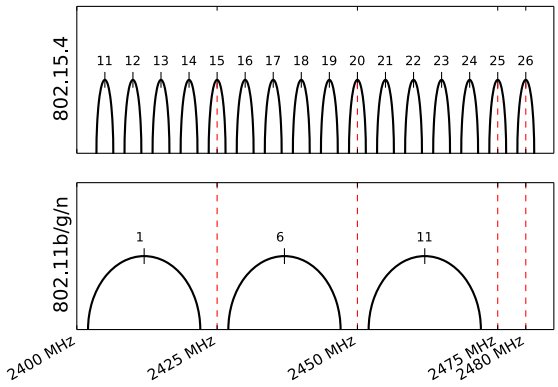


Figure 2.1: Overlapping IEEE 802.15.4 and IEEE 802.11 channels.

In the research community, many studies have been conducted to characterize wireless communications. We present here the key characteristics of a multi-hop wireless environment.

2.1.1 External Interference

Contrary to most wired networks, transmitters must share the medium in the wireless networks, resulting in potential collisions. This fact may cause reliability issues. There are two kinds of interference; the external and the internal. The source of external interference is the co-located/overlapping wireless networks that utilize the same frequency band. Microwave ovens [Gri+17] that use electromagnetic waves in the 2.4 GHz band to heat food, and 2.4 GHz cordless phones [Azm+14] also create external interference. On the other hand, the source of internal interference is the node of the same wireless network that transmits simultaneously at the same frequency. Our literature review will be concentrated on the external interference.

In [Hit+14], authors experimentally study the effect of external interference from communication technologies and home appliances that used the 2.4 GHz ISM band on the reliability of an IEEE 802.15.4 network. The analysis of their experiments exposes that IEEE 802.11 networks cause a substantial decrease on the Packet Reception Rate (PRR) of an IEEE 802.15.4 network. The above can be explained by the fact that IEEE 802.11 nodes transmitting in a much higher transmission power (x100) than the IEEE 802.15.4 nodes. Furthermore, the CSMA technique implemented in IEEE 802.11 networks is unable to detect the transmissions of IEEE 802.15.4 nodes resulting in the corruption of their transmissions. As far as Bluetooth-based networks are concerned, their impact on the reliability of IEEE 802.15.4 networks compared to IEEE 802.11 is insignificant [Nat+16; Boa+11]. Finally, appliances such as cordless phones, wireless cameras, and microwave oven emit continuously energy, causing connection loss among interfered IEEE 802.15.4 nodes.

In [Yaa+16], Yaala *et al.* investigate (both analytically and experimentally) the impact on the reliability of the co-existence of several IEEE 802.15.4-TSCH networks. The authors distinguish two cases of coexistence of TSCH networks, synchronized and non-synchronized. The results of their research indicate that in the case of the synchronized networks, the collisions are scarce, and they can provide high reliability. On the other hand, in the asynchronous TSCH networks, which is the most common, the number of collisions is much higher. Finally, reliability significantly decreases as the number of networks increases, especially in the case of non-synchronized networks.

2.1.2 Temporal and Spatial Characteristics

In the research community, several studies have been conducted to investigate the question *"Whether the quality of radio links depends on the location of the radio links in space and whether*

it varies over time?".

In [Cer+05a], authors study the properties of the radio links of a WSN by conducting a series of experiments in both indoor and outdoor environments. The authors use the reception rate as the radio links' quality metric, which is the ratio of successfully received packets to the total transmitted packets. First of all, they demonstrate that the radio link quality may not be perfectly correlated with its euclidean distance. The authors highlight that radio links with very low or very high reception rates tend to be more stable over time, while the radio links with medium reception rates values tend to be more unstable. Furthermore, the radio links with medium reception rate present marked asymmetry. On the other hand, radio links with very high or very low reception rates tend to be highly symmetrical.

Cerpa *et al.* [Cer+05b] study in depth the temporal properties of radio links in low power wireless networks by analyzing data collected from experiments conducted on a testbed of 55 wireless sensors located in a grid topology. They propose the Required Number of Packets (RNP) as a metric for radio link quality assessment instead of the Reception Rate (RR). RNP is defined as the average number of packets that must be sent before a packet is received. The RNP metric outperforms RR because it takes into consideration the underlying distribution of losses. Moreover, the authors highlight that good links are stable over very long periods of time. On the contrary, bad links tend to be stable for a shorter period of time. According to the authors, routing algorithms should prefer to route the data traffic through high quality radio links because they are quite stable over time.

Srinivasan *et al.* [Sri+10] study the radio link properties of a WSN by conducting a plethora of experiments with two different types of platforms in three different testbeds. First of all, they observe that the reliability of some radio links fluctuates between the two extremes values (0% or 100% PRR) in a short period of time. Furthermore, the PRR of a radio link depends on the communication channel. Another noteworthy observation was that IEEE 802.15.4 radio links exhibit periods of perfect and zero reception; in other words, the packet reception is temporally correlated. The co-existence of various network communication technologies in the same portion of the radio spectrum (ISM band) causes packet losses at multiple nodes. In particular, the IEEE 802.11 due to much higher transmission power comparing to IEEE 802.15.4, can cause spatially correlated losses to IEEE 802.15.4 networks. Furthermore, their results indicate that only IEEE 802.15.4 channel 26 is not affected by the interference from IEEE 802.11 since it does not overlap with any IEEE 802.11 channel (Fig. 2.1). The authors observe the existence of many asymmetrical links for short periods of time and few asymmetrical links for long periods of time. Finally, they observe that the Acknowledgment Reception Ratio (ARR) and PRR are not equal.

In [Pap+16a], authors experimentally investigate the time-dependency of the radio link quality. To do so, they repeated the experiments seven times over different days and time periods of each day. To characterize the radio links, they measure the Packet Delivery Ratio (PDR), and Received Signal Strength Indicator (RSSI) of each channel from the 16 considered IEEE 802.15.4 channels. The authors observe, the extremely low performance of a set of radio channels in a particular area with physical constraints of the testbed due to the multi-path fading effect. Furthermore, their experiments confirm that PDR and RSSI are not necessarily correlated. Moreover, their experiments show that the presence of IEEE 802.11 networks impacts the reliability of IEEE 802.15.4 networks. The authors identify that only very few links (i.e., less than 10%) remain stable and good over time.

Els [Els16] study the radio link properties over long time periods by deploying a 17-node sensor network in a campus building with many coexisted and heavily used wireless networks (IEEE 802.11g). The analysis of the collected data show that the quality of a radio link varies on at least three different timescales: seconds and minutes, hours and days, months timescale. Furthermore, they show that the quality of channels depends on their spatial characteristics and that there are some asymmetrical links.

2.1.3 Link Asymmetry

The magnitude of the difference in quality between the uplink and the downlink characterizes the asymmetry of a radio link. Link asymmetry plays a crucial role in designing metrics for assessing the quality of radio links. Moreover, it significantly affects the operation of higher layer protocols [Bac+12].

Sang *et al.* [San+10] study the radio link asymmetry by conducting experiments in an indoor testbed. The analysis of the results show that, especially in low-power WSNs, a significant portion of the radio links are asymmetrical. They conclude that link asymmetry is due to the low transmission power in dense networks and the euclidean distance between receiver and transmitter, thus, as distance increases, the probability to have an asymmetrical link increases too. As a second step, the authors propose a new one-way link quality metric, which is the Expected number of Transmissions over Forwarding radio links (ETF). ETF is used to exploit asymmetric links to improve convergecast routing in WSNs.

2.1.4 Multi-path Fading Effect

When the transmission of a signal follows more than one path towards the receiver it results in significant signal degradation and, thus, the multi-path fading effect occurs [BS15]. The reflection of the radio signal to the obstacles of the environment causes the multi-path effect.

Puccinelli *et al.* [PH06] study in depth the mulitpath fading effect in WSNs with static nodes through simulations, experiments, and the development of an analytical model. Their key findings are that the multi-path fading effect is a deterministic and spatial phenomenon. Furthermore, they show that the multi-path fading effect has a weak correlation with time and depends on the position of the nodes in the environmental space. Finally, they show that the wideband radios do not defuse the multi-path fading effect always, especially in indoor deployments.

In [Wat+10], the authors study the multi-path fading effect through experiments in the context of WSNs and how it can be mitigated through channel hopping. Research results highlight that the multi-path fading effect can be overcome either by changing the location of one of the radio link's nodes by at least 5.5 cm or by switching the communication frequency by 5 MHz for long-distance radio links and by 25 MHz for short-distance radio links. Based on the above results, the authors propose the use of channel hopping to combat the multi-path fading effect and, more specifically, recommend the use of a specific hopping pattern where successive channels are separated by 25 MHz. Moreover, they propose the complimentary use of antenna diversity, where nodes have multiple antennas spaced at least by 5.5 cm.

Watteyne *et al.* [Wat+15a] conducted an experimental study to record the connectivity between 350 nodes in a typical office environment. The analysis of the data gathered from the experiments show that the percentage of "good" radio links ($PDR \geqslant 90\%$ on all channels) is small and that the majority of radio links is characterized as "unbalanced" ($PDR \geqslant 90\%$ on some channels, $PDR < 90\%$ on others). Moreover, they observe that "good" radio links are present only when the distance between the transmitter and the receiver is short ($\leqslant$ 5m). The authors highlight that beaconing activity from the Wi-Fi Access Points (APs) is adequate to provoke significant downgrading of radio links' quality. Moreover, the authors study the effects of multi-path fading and conclude that it could cause radio links with 100% reliability to transition to 0% reliability due to changes in the environment. However, not all frequencies are affected at the same time. Finally, they prove that the channel hopping technique achieves more stable topologies.

2.1.5 Overview

The above studies resulted in the following conclusions/observations, as shown in Table 2.1.

- The coexistence of different communication technologies in the same frequency band causes the quality of IEEE 802.15.4 networks to deteriorate.

Literature	Temporal Characteristics	Spatial Characteristics	Channel Diversity	Link Asymmetry	Multi-path fading	External Interference
[Hit+14]	☐	☐	☐	☐	☐	☑
[Yaa+16]	☐	☐	☐	☐	☐	☑
[Cer+05a]	☑	☐	☐	☑	☐	☐
[Cer+05b]	☑	☐	☐	☐	☐	☐
[Sri+10]	☐	☑	☑	☑	☐	☑
[Pap+16a]	☑	☐	☐	☐	☐	☑
[Els16]	☑	☑	☐	☑	☐	☐
[San+10]	☐	☐	☐	☑	☐	☐
[Wat+15a]	☐	☐	☐	☐	☑	☑
[PH06]	☐	☐	☐	☐	☑	☐
[Wat+10]	☐	☐	☑	☐	☑	☐

Table 2.1: Summary of state-of-the-art contributions

- The radio link quality is time-dependent.
- The spatial characteristics of radio links affect their quality.
- The radio link quality varies according to the communication channel.
- There is a significant percentage of radio links that present asymmetry in their quality between uplink and downlink.
- The multi-path fading effect has a remarkable presence, especially in indoor WSN deployments.

In our study presented in Chapter 3, we try to confirm the above radio links characteristics using a TSCH network, which runs the OpenWSN [Wat+12] stack in an indoor environment. Furthermore, we investigate the necessity of the usage of a link-based adaptive blacklisting technique.

2.2 LLN Stack

The IEEE standardized the IEEE 802.15.4e TSCH MAC amendment to fulfill the requirements of industrial wireless networks. IEEE 802.15.4-TSCH is able to cope with the external interference and multi-path fading effect, which are the dominant causes of the radio link unreliability, as we showed in the previous section. IETF has standardized several protocols for the LLNs, compliant to IEEE 802.15.4 radios, aiming to provide IPv6 connectivity to resource-constrained devices. All these protocols form the LLN Stack (Fig. 2.2). In this section, we present the most important protocols of them, focusing on TSCH, which is the base of our proposals to provide reliable communications for the IIoT.

2.2.1 IEEE 802.15.4-TSCH

Using a different physical channel for successive transmissions allows to reduce the impact of external interference and to improve the network reliability [Wat+09; Gom+16]. Indeed, the

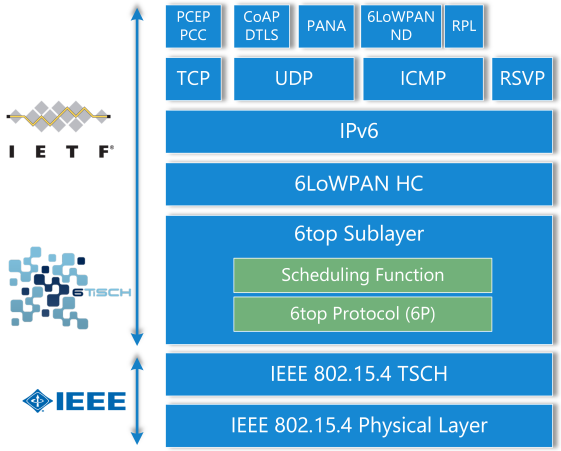


Figure 2.2: LLN Stack [Pal+14]

standardization bodies have proposed to use channel hopping techniques, which allow different packets to be transmitted over different frequencies. More specifically, the failed packet will be re-transmitted through another physical channel, to increase the probability of successful reception, particularly in the presence of narrowband external interference. In channel hopping networks, it is essential for the two parts of a radio link to follow the same channel hopping sequence and to change/hop their operation channel at the same time [Sta+13]. Otherwise, the nodes will try to communicate with each other using different radio channels, making communication impossible (*deafness*).

The frequency/channel hopping techniques can be classified into the following categories as in [Ziv16]:

- **Slow Frequency Hopping:** *"When the hopping rate is lower than the bit rate of data i.e., one or more data bit are transmitted within one hop"*. This technique is suitable for low-cost hardware.
  - **Slotted Frequency Hopping:** *"When the operating channel changes/hops in every timeslot"*. A technique that belongs to the category of the Slow Frequency Hopping techniques, suitable for TDMA MAC protocols.
- **Fast Frequency Hopping:** *"When the hopping rate is higher than the bit rate of data, i.e. one data bit is divided over multiple hops"*.

Bluetooth is a frequency hopping protocol; the operating channel changes up to 1600 times/sec. The number of available channels is 80 (1 MHz wide). Bluetooth Low Energy (BLE) was proposed by Bluetooth Special Interest Group (SIG) in the Bluetooth 4.0 specification [Blu10]. BLE targets short-range communications (up to 50 m) and considers devices with energy constraints. BLE (like Bluetooth) operates in the 2.4 GHz ISM band but uses 40 channels (2 MHz wide). An adaptive frequency hopping algorithm may be used to exclude the channels with the poorest quality from the frequency hopping sequence to further improve the reliability [Pal+16].

TSMP [PD08] is a proprietary protocol that combined time synchronization and slow (slotted) channel hopping. TSMP introduces the concept of channel blacklisting; thus, the set of available channels for channel hopping sequence can be restricted to avoid interference from co-existing networks. Furthermore, TSMP follows a centralized approach to schedule the transmissions.

The proprietary protocols WirelessHART (2007) [Spe08] and ISA100.11a (2009) [ISAy] were based on the core ideas of TSMP and they have targeted to the industrial market by providing high reliability and low energy consumption.

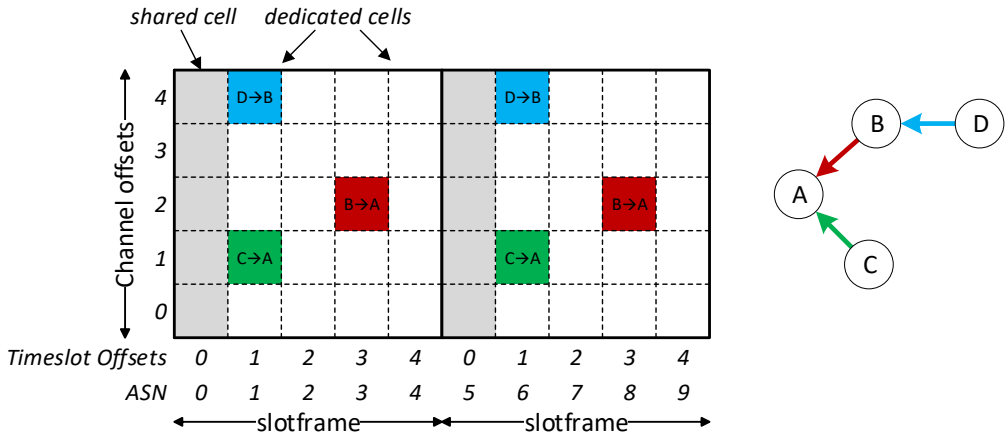


Figure 2.3: An example of TSCH scheduling for 4 nodes topology.

WirelessHART employs a central network manager to schedule communications among the devices, while it replaces CSMA in the IEEE 802.15.4 with TDMA. Furthermore, WirelessHART uses a channel hopping approach across the 16 available frequency channels in the 2.4 GHz band.

The Standards & Practices Board of the International Society of Automation (ISA) approved ISA-100.11a that guarantees deterministic communication latency, while increasing the network reliability. ISA-100.11a provides a combination of three channel hopping techniques (slotted, slow and hybrid) to support different communication schemes as TDMA and CSMA. In slotted channel hopping, the communication channel hops (changes) on every timeslot whereas in slow channel hopping it hops every two or more timeslots. It should be noted that ISA-100.11a channel hopping techniques are classified to the category of slow frequency hopping according to the definitions of [Ziv16] . Furthermore, the hopping pattern separates the radio channels by at least three IEEE 802.15.4 channels (i.e., 15 MHz).

IEEE 802.15.4-2015 [Ieea] has proposed the TSCH mode, inspired mainly by the previous WirelessHART and ISA100.11a standards and the TSMP. More specifically, the TSCH mode aims to provide a high level of reliability by exploiting slow channel hopping combined with a strict schedule of the transmissions. We present next the key characteristics of TSCH.

Slotframe structure

In TSCH networks, time is divided into timeslots of equal length. The length of the timeslot should be sufficient, so a data packet and its corresponding acknowledgment may be exchanged between a transmitter and its receiver. At each timeslot, a node may transmit or receive a packet, or it may turn to sleep mode for saving energy. A set of timeslots constructs a slotframe that repeats over time. Each timeslot is labeled with an Absolute Sequence Number (ASN), a variable which counts the number of timeslots since the network was established.

To avoid confusion from now on, we provide the following distinction:

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

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

Organization of the transmissions

Based on the ASN and the schedule, a node can transmit or receive a data packet or go to sleep. A schedule is a matrix of a fixed set of cells, as depicted in Fig. 2.3. Each cell consists of a pair of a timeslot offset and a channel offset.

The TSCH standard supports two medium access techniques:

Shared cells can be used by multiple, possibly interfering transmitters. A device dequeues a packet and transmits it immediately in the next shared cell. If an acknowledgment was required and was not received, the transmitter assumes that a collision occurred and selects a random backoff, skipping the corresponding number of shared cells;

Dedicated cells are allocated carefully to avoid collisions. Thus, a transmitter does not implement random access during these cells. However, a packet can be retransmitted if no acknowledgment is received, due to a bad radio link quality or external interference.

Different centralized and distributed scheduling algorithms have been proposed for the allocation of cells to the radio links [Her+17]. In a centralized approach, a controller needs the global knowledge of the radio and interfering topology, as well as the traffic requirements of each flow to allocate a set of cells to each radio link. Inversely, distributed solutions need to avoid collisions while reacting quickly to changes. In distributed algorithms, nodes negotiate only with their neighbors to allocate the appropriate cells (timeslot and channel offsets) to each active radio link.

The organization of the schedule in cells enables parallel transmissions. Thus, interfering radio links can be scheduled at the same time using different channel offsets (in order to avoid collisions) and non-interfering radio links can be scheduled at the same cell (timeslot and channel offsets). As it can be easily understood, parallel transmissions efficiently exploit the network resources and enable the construction of compact schedules.

In Fig. 2.3, a typical schedule is illustrated. It consists of a matrix of channel offsets and timeslots (a slotframe), which repeats indefinitely over time. One shared cell is placed at the beginning of the slotframe for best-effort control traffic and broadcast packets (Enhanced Beacon (EB), DODAG Information Object (DIO), etc.) and three dedicated cells (C, A),(D, B) and (B, A). The links (C, A) and (D, B) have been scheduled in the same timeslot, but over a different channel offset (1 and 4, respectively), since they are interfering with each other.

Time Synchronization

To operate smoothly, the nodes that participate in a TDMA network need to be synchronized. Two (or more) nodes are synchronized when they use the same ASN for the current timeslot and when their timeslots' bounds are aligned within a margin of error.

Nodes that join a TSCH network are informed about the current ASN value by the receiving EB packets, which are sent by the nodes participating in the network. In a TSCH network, the Personal Area Network (PAN) coordinator disseminates the time information outwards. Each device selects another device as a time source neighbor to synchronize its network time at periodic time intervals. A device is synchronized with its time source neighbor using the *frame-based* or the *acknowledgment based* method [Ieea].

In the *frame-based* synchronization, every time a node receives a data packet from its time source neighbor, it records its arrival time. The receiver then calculates the time interval (time correction) required to add or subtract from its own clock to be synchronized (Fig. 2.4). In the *acknowledgment based* method, every time a node receives a data packet, it records its arrival time. It then calculates the time correction value and sends it to the sender through an ACK packet. The sender receives the ACK, and in the case where the receiver node is its time source neighbor, it adjusts its clock according to the received time correction value (Fig. 2.4).

In conclusion, regardless of the synchronization method, the time correction (T_{corr}) is calculated by the receiver as follows:

$$T_{corr} = TsRxOffset + TsRxWait/2 - T_{arr} \tag{2.1}$$

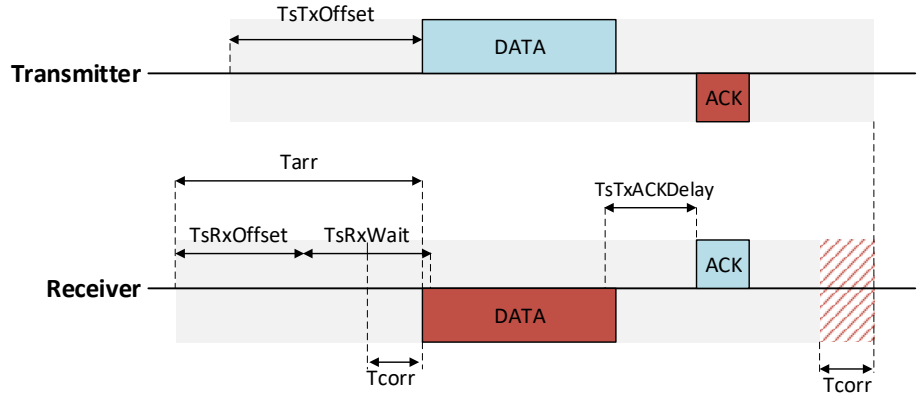


Figure 2.4: Timeslot timings.

The T_{arr} corresponds to the actual time of the packet arrival, $TsRxOffset$ is the time interval between the start of the timeslot and the beginning of listening to the medium for receiving an upcoming packet, and $TsRxWait$ is the maximum listening time interval of the medium before starting receiving a packet. In addition, the following condition holds:

$$TsTxOffset = TsRxOffset + TsRxWait/2 \tag{2.2}$$

where $TsTxOffset$ is the time interval from the beginning of the timeslot where the transmitter begins to transmit the packet.

In the case where a node has not communicated with its time source neighbor for a determined time period (*Keep Alive period*) then it sends to its time source neighbor a Keep-Alive message. After the reception of the Keep-Alive message, the time source node embeds the time correction to the ACK (*acknowledgment based* synchronization) [Duq+17].

Channel Hopping

IEEE 802.15.4-2015 TSCH implements a channel hopping approach to combat noise and interference in order to achieve high network reliability [Wat+09]. TSCH maintains a schedule and assigns a set of cells to each radio link. At the beginning of each timeslot, the channel offset is translated into a physical channel using the ASN value as it shown below:

$$frequency = F\Big(\big(ASN + channelOffset\big)\ \%\ nFreq\Big) \tag{2.3}$$

where ASN denotes the Absolute Sequence Number of the timeslot, $channelOffset$ the channel offset of the current cell, $nFreq$ is the number of available channels (e.g., 16 when using IEEE 802.15.4-compliant radios at 2.4 GHz with all channels in use) [Wat+15b] and $F()$ maps the index of the channel hopping sequence to a physical channel.

To assign a different channel to the same cell in successive slotframes, it is necessary the slotframe length and the number of physical channels to be mutually prime numbers.

Medium Access in Shared Timeslots

During a shared cell, more than one transmitter may attempt to transmit at the same time resulting in a packet collision. In order to resolve collisions, TSCH implements a transmission backoff algorithm. If the transmission of a packet on a shared cell failed i.e., the sender did not receive the corresponding acknowledgment packet, then the sender initiates the Collision Avoidance (CA) retransmission algorithm as follows [Ieea]:

1. The sender defers the transmission of the corresponding packet for w shared cells where w is a random number in the range of 0 to $2^{BE} - 1$ and BE is the backoff-exponent (initial value $BE = macMinBE$).

2. If the retransmission on a shared cell failed (the corresponding ACK is not received) and the number of retransmissions is below the maximum allowed number, the backoff exponent is increasing ($BE = min(BE + 1, macMaxBE)$) and the algorithm goes to step 1.

3. In case of a successful transmission, the backoff exponent is reset to its initial value ($BE = macMinBE$), and the algorithm is terminated.

In TSCH, the CCA is used to defer the transmission in case of strong external interference.

2.2.2 6TiSCH Overview

The IETF IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) Working Group is a standardization effort to augment the TSCH's protocol stack with the suitable layers to support the IPv6 protocol. The ultimate goal of 6TiSCH is to create an open-standard based protocol stack for deterministic IPv6-enabled wireless networks by combining the lower IEEE 802.15.4 PHY and IEEE 802.15.4e TSCH MAC layers with the upper layers of IETF (i.e., 6LoWPAN, RPL, Constrained Application Protocol (CoAP), etc.), as depicted in Fig. 2.2. The management of the TSCH schedule is one of the gaps that fill the 6TiSCH. The 6TSCH Operation Sublayer (6top) Protocol (6P) [Wan+18] and the Scheduling Function (SF) are the two entities responsible for the scheduling functionality.

6P is a protocol, which allows two neighbor nodes to negotiate how to modify their schedules. The 6top sublayer defines two cell types, the hard and soft cells. A soft cell can be managed (added, removed, etc.) by 6P. On the other hand, hard cells cannot be reallocated by 6P and can be used by central scheduling algorithms. The neighbor's schedule management is handled by the following commands ADD, DELETE, RELOCATE, COUNT, LIST, SIGNAL, CLEAR. 6P commands are executed through "6P transactions", which can be either a two-step or a three-step message exchange between negotiating neighbors. The result of each transaction is, the participating nodes either committing or aborting it.

A 2-step 6P ADD transaction proceeds as follows (Fig. 2.5):

1. The transmitter sends a 6P request in unicast, with a list of available cells (e.g., $[(2, 2), (3, 5)]$). The receiver acknowledges this request;

2. The receiver verifies a sufficient number of these cells is available in its schedule. It then constructs a 6P reply transmitted in unicast, acknowledged by the transmitter.

When the transaction has completed, both the transmitter and the receiver have consistently modified their schedule.

SF performs the cell allocation policy by triggering the appropriate 6P transactions (commands) according to application needs and network traffic. A plethora of SFs has been proposed in the literature to address different traffic patterns or to optimize a network feature. We will detail them later in section 2.4.

2.2.3 6LoWPAN

The IPv**6** over 6LoWPAN [Kus+07] protocol allows the transportation of IPv6 packets over IEEE 802.15.4 radio links; thus, it provides Internet Protocol (IP) connectivity to constrained resources devices. The IEEE 802.15.4 networks can benefit from the pervasive nature of IP networks and use the existing infrastructure. Furthermore, it can benefit from the already developed IP-based technologies, mostly open and free.

The minimum packet size of the IPv6 packet is much larger than the maximum packet size of an IEEE 802.15.4 packet, so it required an adaption layer (6LoWPAN) between IP and IEEE

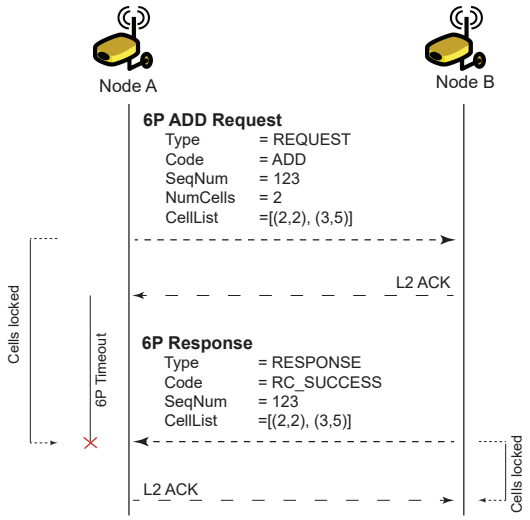


Figure 2.5: An Example 2-Step 6P Transaction

802.15.4 layers to fragment and reassemble the packets. 6LoWPAN is responsible for the compression of the upper layers packet's headers (IPv6, User Datagram Protocol (UDP)/Internet Control Message Protocol (ICMP)) to maximize the useful payload of the packets that carried out in the lower layers [Ols14].

2.2.4 RPL

The nodes in a WSN should transmit the data that sense/generate to a central node-sink. Hence, it is necessary to construct a routing path from them towards the sink [DG+16]. IETF ROLL Working Group standardized IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) [Win+12] as the routing protocol for LLNs. LLNs consist of nodes with limited resources (such as processing power, memory and energy) and their radio links are unreliable, unstable and mainly support low data rates.

RPL constructs routing paths for each node towards the sink (aka border router or gateway) by building a Destination-Oriented Directed Acyclic Graph (DODAG). We note that in a network with multiple sinks, there are more than one DODAGs, directed to each one of the sinks. RPL assigns to each node a rank that represents its position in the DODAG. During the DODAG construction, the nodes create a set of parents and select as a preferred parent, the node with the lowest rank. The Objective Function estimates the rank of each node by converting a set of routing metrics (e.g., delay, link quality, hop distance) into ranks. The sink initiates the construction of the DODAG by sending DIO control messages periodically. The DIO messages inform the receiving nodes about the set of their neighbors and their ranks. A node after the delivery of a DIO message uses the Objective Function and the rank of its neighbors to compute its rank and to select its preferred parent. The node forwards the packets to the preferred parent in order to be delivered by the sink. Afterwards, the node transmits new DIO messages to disseminate the routing information. The nodes disseminate Destination Advertisement Object (DAO) messages to support upward traffic (from the sink to the nodes) [Anc+13; GK12; Acc+11].

2.2.5 CoAP

The WSNs consist of nodes/devices with constrained resources. Thus, IETF proposed CoAP as the application-layer protocol. CoAP is a specialized web transfer protocol that tailors the HTTPbased Representational State Transfer (REST) [FT00] interface for the demands of the

constrained nodes. CoAP supports one-to-many and many-to-one communication patterns and uses UDP instead of Transmission Control Protocol (TCP) [She+14; Kov+11].

2.3 Blacklisting/Whitelisting Techniques

The channel hopping technique, as mentioned earlier, copes with the existence of external interference and the multi-path fading effect achieving to improve network reliability [Wat+09]. However, the performance of this technique is limited when there is a strong diversity in the quality of the channels. As a result, applying a channel hopping technique when all available channels are evenly used does not yield to optimal performance. A possible enhancement is to exclude from the channel hopping sequence the channels that exhibit the lowest reliability (*'bad'* channels) to avoid using them for transmission, a technique known as *blacklisting*. On the other hand, *whitelisting* consists of identifying the channels that exhibit the highest reliability (*'good'* channels) to use them to the generation of the channel hopping sequence.

The terms *blacklisting* and *whitelisting* may be used interchangeably since they correspond to dual concepts. Blacklisting consists of constructing the list of radio channels that *cannot* be used for transmission while whitelisting references all the radio channels that could be employed for transmission. While they correspond to two complementary sets, they lead exactly to the same behavior. Let $\mathcal{C}$ denote the set of all radio channels, we have:

$$Blacklist \cup Whitelist = \mathcal{C} \tag{2.4}$$

There are two approaches to construct a global or local blacklist. In the global blacklisting approach, the network nodes exclude from their channel hopping sequence the same channels that present the lowest quality. However, blacklisting a channel globally might be suboptimal since the quality of a specific channel may vary among the radio links due to their spatial characteristics. The construction of a global blacklist requires from a central entity to collect the appropriate statistics of each channel on all radio links and decide which channels to blacklist globally. Such a technique burdens the network with additional packets and doesn't adapt quickly to temporal changes in channel quality.

In local blacklisting, the two parts of a radio link negotiate which channels to blacklist according to their quality. Local blacklisting is more efficient but can cause deafness between transmitter and receiver in case of inconsistent blacklists due to the lack of coordination between them.

2.3.1 Blacklisting Process

The process of applying a blacklisting technique to channel hopping networks consists of the following steps (Fig. 2.6):

- **Estimation** of the quality of the channels.
- **Classification** of channels into *good* or *bad*.
- **Modification** of the channel hopping sequence in order to utilize only the *good* channels.
- **Dissemination** of the modified channel hopping sequence to the appropriate nodes.

Channels quality estimation

The identification of *bad* channels is performed by the exploitation of a Link Quality Estimator (LQE). LQEs can be classified in *hardware-based* and *software-based* [Bac+12].

The noise level-energy detection- represents a typical hardware-based estimator. The estimator module reads the value of RSSI register from the radio chipset for a specific channel when none of the nodes in the network transmits. As a result, the retrieved value corresponds to the

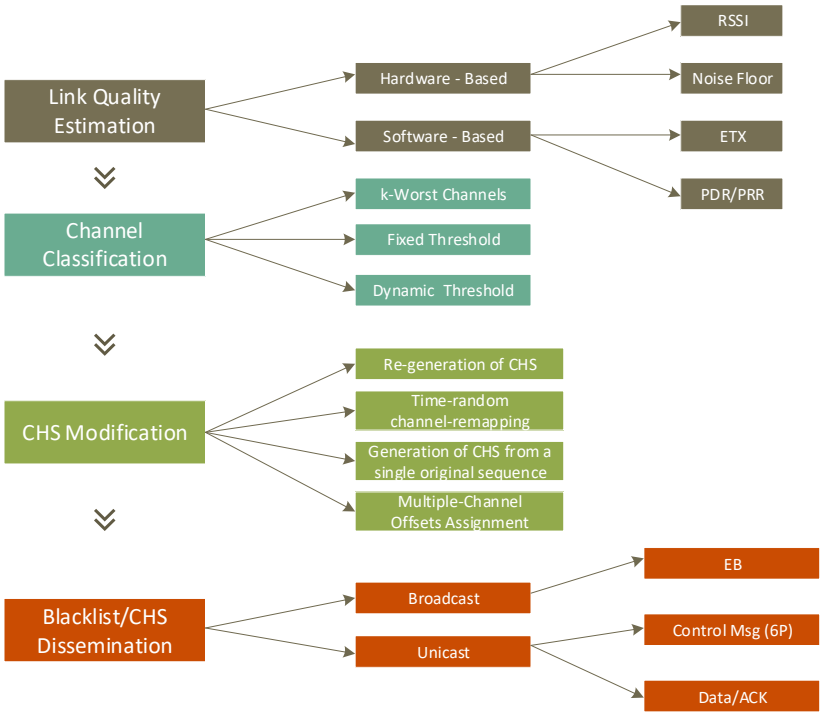


Figure 2.6: The components of a Blacklisting Process

energy level of the channel's noise. The noise level metric has the advantage that it can be also applied to the excluded channels. However, it consumes additional energy by listening to the wireless medium in inactive periods. Finally, noise level LQEs cannot detect the intra-network collisions because the RSSI value is measured when the network nodes do not transmit. The estimators based on the measurements of Link Quality Indicator (LQI) and Signal-to-Noise Ratio (SNR) are also classified into *hardware-based* LQEs.

The *software-based* LQEs are based on the measurements of the received/transmitted data packets and received/transmitted of ACK packets. There are located either on the receiver or the sender side. PDR (PRR in [Els+17]) is a sender-side estimator that commonly used in the classification of the channels in blacklisting algorithms. PDR is defined as the ratio of delivered ACKs to transmitted packets, calculated on a per-channel basis. PDR can detect the multi-path fading effect and takes into consideration the radio link asymmetry. PDR's main drawback is that it can not assess the blacklisted channels since they are not used, resulting in remaining these channels forever in the blacklist regardless if their reliability has been improved or not.

Another approach to identify the bad channels is to use dedicated nodes. They estimate the quality of the radio links without burdening the network nodes. Such an approach of course burdens the cost of the deployment of a WSN.

Usually, the LQEs predict with accuracy the quality of the channels by incorporating history measurements using some filtering/smoothing technique such as Exponential Weighted Moving Average (EWMA), Window Mean Exponentially-Weighted Moving Average (WMEWMA) [WC03] and Simple Exponential Smoothing (SES).

Channel classification

One challenging issue is to classify the available channels as blacklisted or whitelisted. One option is to blacklist a pre-defined number of channels presenting the lowest estimated quality

($k - worst\ channels$). However, it is quite difficult to choose the optimal number of blacklisted channels because it has to take into consideration the maximum number of parallel transmissions of the schedule, the required number of channels of channel hopping sequence in order to tackle abrupt variations on channels reliability and do not include low-quality channels in the list, thus, keeping the reliability at high levels.

The other option is to blacklist the channels with a quality below a predefined threshold. The apparent advantage of this approach is that the whitelist consists of the best quality channels, thus, improving the link reliability but there is uncertainty regarding the number of the blacklisted channels. Therefore, considering the worst-case scenario, a large number of channels even all can be blacklisted, limiting the benefits of the channel hopping technique and limiting the number of parallel transmissions.

Finally, another option is not to arbitrarily choose a threshold value, but to calculate it dynamically in order to always select the best channels.

Channel hopping sequence modification

After identifying the *bad* channels, the next step is the modification of the Channel Hopping Sequence (CHS) in order to include only the *good* channels or use less frequently the *bad* radio channels. The default CHS is generated in such a way as to minimize the probability of interference between two radio links using different channel offsets - *good/optimal property.*

In this thesis, we use the term *internal collision* to describe the collision of packets when the *good property* is violated. That is when on two (or more) radio links are assigned different channel offsets, but the Equation 2.3 outputs for the same ASN the same physical radio channel for both radio links resulting in packets' collision. We chose the term internal since the collision comes from the transmitters of the same network.

The techniques presented in the literature are the following:

- **Regeneration of CHS:** The CHS is regenerated to maintain the above property using only the *good* channels. However, such an approach burdens the node/nodes that are responsible for the modification of CHS [Ban+18].

- **Time-random channel-remapping:** In this technique, all the nodes use the same (default) CHS and use in their transmissions only the *good* channels bypassing the use of the *bad* channels. Consequently, there is no extra cost to rebuild the CHS. The sequence of channels used by the nodes is random and depends on the channel's blacklist. Thus, in the case of using per radio link local blacklists, it is possible to have internal collisions [DR12; Kru+19].

- **With the generation of CHS from a single original sequence:** Generation of CHSs using a primary generated sequence, we avoid the regeneration cost of the CHS every time the channel's blacklist is modified. The above technique preserves the good/optimal property of the default CHSs [Shi+15].

- **Multiple-Channel Offsets Assignment:** Each schedule's cell assigned to a radio link consists of a timeslot offset and a list of channel offsets instead of one channel offset, such as MABO-TSCH does [Gom+17]. The blacklist can then be negotiated locally, among the transmitter and the receiver.

  At the beginning of a timeslot, a node uses its channel offsets list to derive the frequency that will be used. More precisely, if the first channel offset corresponds to a blacklisted radio channel, it uses the next channel offset in its list. The process stops when a good radio channel is obtained or the last channel offset is scanned.

  This method is illustrated in Fig. 2.7. The link (A, B) received three different channel offsets. The first channel offset (0) gives a bad radio channel and is not used. Finally, only the third channel offset corresponds to a good radio channel, which will be used for the transmission.

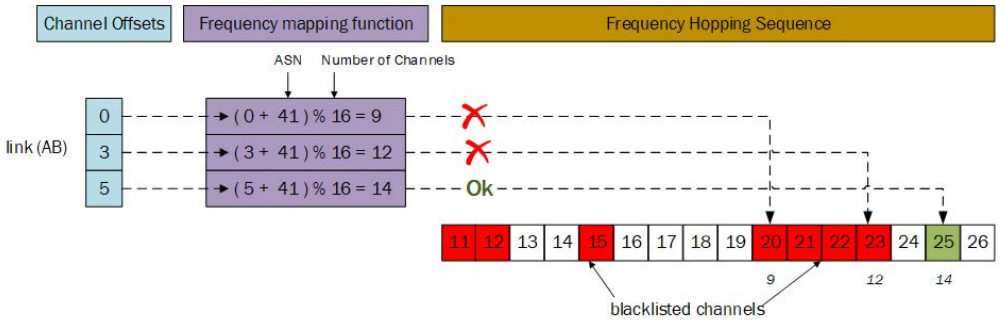


Figure 2.7: Physical channels generation process using multiple channel offsets.

This technique does not create any intra-network collision; the channel offsets are distributed orthogonally (a pair of interfering links never receives the same channel offset). This multi-channel offset scheme supports both centralized and distributed scheduling algorithms: a link has just to reserve several channel offsets.

Channel hopping sequence - blacklist dissemination

The final step of the blacklisting process is to disseminate the CHS to the nodes. Depending on the approach of the modification of the CHS, is required either the dissemination of the whole CHS or only the dissemination of the channels' blacklist. The CHS/Blacklist dissemination ensures that the transmitter and the receiver of a radio link use the same CHS to avoid deafness between them. The CHS/Blacklist is disseminated either from one of the two ends of the radio link or from the border-router, according to the blacklisting strategy (global or local). The CHS/blacklist dissemination could be performed using broadcast or unicast messages.

An Information Element (IE) has been defined by IEEE 802.15.4 TSCH standard on EB broadcast messages to distribute the channel hopping sequence. The absence of the acknowledgment packet for broadcast messages does not ensure that the blacklist was successfully received by the recipients and this could possibly lead to deafness. Additionally, EB messages are transmitted through shared cells that are prone to repetitive collisions. The use of EB messages proves to be inefficient if the rate of channel quality changes is significantly higher than the rate of EB's transmission.

In the case of unicast messages, there are two options either use dedicated control messages (6P) or embed the appropriate CHS/blacklist information into data packets or ACK packets. The usage of control messages increases the network traffic since it increases the congestion on shared timeslots in case of the 6P messages. The embodiment of blacklist/CHS information into each data packet increases the packet payload, thereby burdening network performance.

2.3.2 Global

WirelessHART [Son+08] adopts a global blacklisting scheme in which the network operator manually excludes from the channel hopping sequence the channels with an observed poor performance. Such a technique doesn't need a specific blacklist dissemination mechanism. However, as we noticed in subsection 2.1, the quality of the channels may exhibit very location-dependent and time-dependent characteristics making such an approach suboptimal. Therefore, WirelessHART's blacklisting technique can be beneficial when the wireless network is subject to steady in time external interference, in all of its extents to specific channels.

Tavakoli *et al.* [Tav+15] propose a centralized global whitelisting technique called ETSCH, mainly intended for in-vehicle networks. In ETSCH, the coordinator device applies a Non-Intrusive radio Channel quality Estimation (NICE) by performing Energy Detection (ED) during

idle periods' of the timeslots. The above channel quality estimation technique can be characterized as non-intrusive since it does not use network resources (e.g., dedicated timeslots). However, the EDs are power-consuming, limiting the network's lifetime. NICE is efficient when the timeslot duration is sufficient to compensate clock drifts while letting enough time for energy detection. The coordinator node assesses the quality of each of the available channels by smoothing the gathered energy detection measurements. It then selects the k best quality channels to compile the whitelist. Subsequently, the coordinator node broadcasts the whitelist to the network nodes by using EBs. ETSCH uses a second, less dynamic, whitelist to broadcast EBs to the network, to increase the reliability of EBs transmissions, and eliminate the whitelist inconsistencies, between coordinator and nodes.

The authors extend their technique (ETSCH+DCS [Tav+18]) to take into consideration the interference experienced by nodes that it is not visible from the coordinator node. In this technique, each node estimates the quality of the channels using performed CCA and packet reception status and transmits it to the coordinator node using data packets. The whitelist is constructed and disseminated by the coordinator node after combining the information it receives about the quality of the channels from the nodes with its measurements (energy detections). Addressing multi-hop topologies is still an open challenge, and dedicated timeslots for energy detection may be required.

Jeon *et al.* propose an enhancement of E-TSCH [Tav+15] called E-TSCH with ACQE [JC17] to minimize energy consumption due to performed EDs. In E-TSCH with ACQE, the rate of EDs is adapted to the dynamics of the quality of the channels. Thus, the rate of EDs increases in the case of highly dynamic changes in channels' quality and decreases in more steady radio environments, thereby saving energy.

Gunatilaka *et al.* [Gun+17] study the impact of the number of the available channels of the channel hopping sequence on network topology, routing, and scheduling by conducting experiments on testbeds. Their routing approach defines multi-path routes towards to destination and exploits only the radio links whose PRR is above a threshold in all used channels. Their empirical studies show that a large number of used channels increases the channel diversity but reduces the route diversity at the same time since the number of links that meet the above criterion is reduced. The authors, based on the above observations, propose a global channel selection (whitelisting) algorithm based on WirelessHART. Their algorithm ranks the channels using a new metric that considers the criticalities of nodes to the network topology. It then estimates the maximum (k) number of channels that can be used to successfully generate the routes and the schedule for a given set of flows and selects the k-best channels to be used by all the network nodes. However, such a technique is very time-consuming with heterogeneous blacklists, and most of the approaches first propose to construct the blacklists and then to modify the schedule accordingly.

Zorbas *et al.* [Zor+18a] theoretically show that in local blacklisting techniques based on per radio link multiple channel offsets assignment([Gom+17], [Zor+18b]) can cause a significant deterioration of network performance when blacklist size is large, and the assigned channel offsets are few. More specifically, their analytical model demonstrates that as the blacklist size increases, the probability of generating a whitelisted channel is reduced. As a result, the nodes use for their transmissions more often low-quality blacklisted channels lowering the network's reliability. To overcome the above problems, the authors propose a global blacklisting technique where the nodes locally decide which channel to blacklist ($PDR < 90\%$) and disseminate the blacklisted channels to all the network's nodes using data packets and ACKs. Each time a node blacklists a channel, it associates it with a future time (time label), so that all nodes will exclude that channel after this time, thus, avoiding deafness and network disconnection. Such a technique does not create internal collisions since all nodes use the same channel hopping sequence. However, this is not the optimal solution since it does not take into account the spatial characteristics of external interference and, thus, leads to the use of large blacklist sizes.

2.3.3 Local

Software-based link quality estimators

ISA100.11a [PC11] implements a global blacklisting technique similar to WirelessHART and a local blacklisting scheme called Adaptive Channel Hopping. Each node can estimate the channels' quality by counting the number of successful CCA or ACK or by measuring the received packets' RSSI values [An09]. The node has the right to transmit during a cell if the channel offset does not give a blacklisted physical channel. The above strategy increases significantly the delay and decreases the throughput. The main advantage of this approach is that the receiver does not need to be aware of the channels that the transmitter has excluded from the channel hopping sequence. Therefore, it does not require any type of message to disseminate the blacklist nor synchronization between the nodes. However, such an approach has a very negative impact on the delay, the throughput, and energy consumption; the transmitter has to defer its transmission until the channel hopping sequence provides a non-blacklisted channel.

Shi et al. [Shi+15] provide mechanisms to change on-the-fly the pseudo-random hopping sequence without regenerating from scratch the whole sequence each time the whitelist changes. The proposed algorithm generates the Frequency Hopping Sequence (FHS) by maintaining the good/optimal property of the original FHS. Their proposal based on the generation of the initial sequence, either using the M-sequence as defined on the IEEE 802.15.4e-2012 standard [Ass+12] or using the Cyclotomic Classes [CC05]. The simulation results show that the probability of interference of the proposed algorithm for different whitelist sizes approximate the performance of the default and optimal FHS.

Li *et al.* [Li+15] are the first that use the multi-arm bandit problem to model the process of the selection of the best channels for their use in the channel hopping sequence. They actually propose an adaptive channel selection scheme for TSCH. Each channel is considered to be an independent process to which a variable called the Gittins index [Git79] is associated. The calculation of the Gittins index has been simplified in order to meet the constrained resources of WSN nodes. Thus, a combination of the successful transmissions, the failures of CCA and the consecutive transmissions failures are used. The transmitter of each radio link selects the channels with the best Gittins index and adds them to an IE of the TSCH packet, waiting for the coordinator to acknowledge it. The proposed technique defines some timeslots (*explore timeslots*) in which all available channels are used, so that blacklisted channels can be included in the whitelist (and vice versa if the channel quality is changed). The variability of the radio environment adjusts the frequency of *explore* timeslots. Additionally, it is required that the receiver node remains aware of the transmission packet rate, which is not a realistic assumption (for example when adaptive of event-driven sensing is used).

Gomes *et al.* propose a localized distributed blacklisting protocol called Multi-hop And Blacklist-based Optimized TSCH protocol (MABO-TSCH) [Gom+17]. MABO-TSCH utilizes a receiver-based channel offset assignment technique that associates a set of channel offsets to each non-leaf node so that no internal collisions occur when interfering radio links are scheduled at the same timeslot. The multi-channel offsets assignment process is performed centrally by Path Computation Element (PCE) that uses a graph coloring based algorithm. The main drawback of this approach is that when large blacklists are used and the number of the channel offsets assigned to each radio link is small, the bad channels are used in the transmissions inevitably. As a consequence, the overall reliability is decreasing. The channels' quality estimation is modeled as a multi-armed bandit problem as follows: Each of the 16 channels is considered as an arm of a slot machine. The player's (node) task is to choose at each round, the arm (channel) with the maximum mean reward without the need for any learning phase. MABO-TSCH uses the ϵ-greedy strategy to implement the multi arm bandit problem. Thus, in each round, the player selects the best arm (channel) with probability $1 - \epsilon$ and selects a random arm with probability ϵ, therefore, re-evaluating the previously blacklisted channels. The blacklist is disseminated by encapsulating it into data or ACK packets depending on which node (transmitter or receiver, respectively) selects the channels to be blacklisted, thus, avoiding the use of extra control packets.

Zorbas *et al.* [Zor+18b] propose LOST, a distributed scheduling algorithm that also considers per link-local blacklists. In LOST, nodes blacklist the channels for which the PDR is below a fixed threshold and transmit the blacklist to their preferred parent through data or ACK packets. LOST tackles the issue of internal collisions by assigning multiple channel offsets per radio link. Thus, at the beginning of the timeslot, the transmitter and the receiver of a radio link try one by one the channel offsets that have assigned to them until Equation 2.3 outputs a good channel. The nodes postpone their communication if none whitelisted channel is derived from the above process. In LOST, the process of assigning multiple channel offsets is distributed as the nodes only need to know the maximum degree of the network. On the other hand, in MABO-TSCH [Gom+17], a central entity that is aware of the networks' characteristics (e.g., routing information, network topology, and central schedule) assigns the multiple channel offsets to the radio links. In the case where the network density is heterogeneous, the optimal number of channel offsets is not assigned to each radio link, since the multi-channel offsets assignment is based on the maximum vertex degree of the network.

Banik *et al.* [Ban+18] propose SmartHop. The most innovative part of SmartHop is the blacklist construction on the cloud, avoiding the waste of computational resources of the WSN nodes. The cloud application for each radio link constructs a channel hopping sequence, using only the channels of the highest quality. Furthermore, channels are repeated over the channel hopping sequence according to their quality. Each radio link's final channel hopping sequence is disseminating through the transmission of EBs packets to the participating nodes. The assessment of the quality of each radio link's channels is performed by the cloud application using the channel response metric. The channel response is the ratio of the total packets received from the border router, which are transmitted in the first hop using channel k to the expected number of the total packets that are transmitted by source node on channel k. SmartHop takes into consideration the temporal characteristics of the radio links by re-evaluating the blacklisting channels. For this reason, nodes periodically transmit a short dummy packet using one of the channels in the blacklist. Such a technique consumes a significant amount of energy. Moreover, SmartHop does not exploit any aging technique, so it is difficult for a blacklisted channel to be re-entered on the channel hopping sequence.

Hammoudi *et al.* [Ham+] propose a localized blacklisted technique called Enhanced Timeslotted Channel Hopping (E-TSCH). E-TSCH evaluates the available channels using an Intelligent Link-Quality Estimation process (I-LQE) based on the smoothed values of the Goodness and Badness metrics. Goodness and Badness metrics count the successful and unsuccessful transmissions at collision-free cells, respectively. These metrics are updated accordingly from packet retransmission count (sender-side) and ACK count (receiver-side). Both ends of the radio link measure the quality of the channels, but the selection of the blacklisted channels is performed from the sender. Although a bad channel is not used in node's transmissions, it can be used in packet receptions, so it's quality index is not appropriately updated, and in the case where its quality improved is then removed from the blacklist. However, a specific channel may be blacklisted by all radio links that terminate or start from a node due to spatial characteristics of the external interference, so its quality indicator is not updated at all and the channel remains blacklisted.

Hardware-based link quality estimators

Du *et al.* [DR12] propose a localized blacklisting method for TSCH in which specific timeslots (Noise Floor (NF) timeslots) are reserved to measure the noise level on each physical channel (Fig. 2.8). Transmissions between nodes are prohibited during NF timeslots, so the measurements collected correspond to the noise level of the used channel. The gathered measurements, once smoothed out (SES), are used by the nodes to assess the quality of the channels and to construct their local blacklist. There are two alternatives to construct the local blacklist, to select channels with a quality below a predetermined threshold, or to select the k channels with the lowest quality. A node's local blacklist dissemination takes place using Advertisement (ADV) packets for downstream neighbors and sending unicast messages to upstream neighbors through

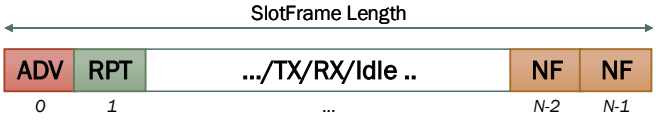


Figure 2.8: A-TSCH defines NF and RPT timeslot types additionaly to the pre-existing ADV, TX, RX, and Idle timeslot types of TSCH.

additional auxiliary RePorT (RPT) timeslots (Fig. 2.8). Nodes transmit their data packets taking into account both their local blacklist and the receiver's local blacklist (the intersection of both blacklists). Thus, this method takes into account the potential asymmetry in the quality of the two directions of a radio link. A-TSCH does not provide any mechanism for avoiding internal collisions. The reservation of two additional timeslots reduces the available network resources and wastes energy. The size of the slotframe affects the noise floor sampling rate of A-TSCH.

Elts *et al.* [Els+17] propose three adaptive channel selection methods (PRR downstream, RSSI upstream and RSSI downstream), based on the evaluation of channels quality using RSSI and PRR measurements. The RSSI sampling should be done when there are no packet transmissions from any node to successfully detect the external interference. Thus, the nodes measure the RSSI during the inactive time interval between the start of the TSCH timeslot and the start of the transmission of a data packet. As a result, this approach does not waste network resources such as dedicated timeslots. The quality characterization of the channels is achieved from the smoothing (EWMA) of the gathered samples. Channels performing below a predetermined threshold are excluded from the channel hopping sequence. Nodes then use EB messages to transmit the blacklist to their downstream neighbors (upstream driven) or unicast notification messages to transmit the blacklist to their upstream neighbors (downstream driven). In the latter case, it's possible to detect temporal inconsistency between the two parts of a radio link, in that case, the transmitter postpones its transmission. The evaluation of the proposed methods through simulations and experiments in the testbed highlights the superiority of the PRR based method since it can successfully detect both the external interference and the frequency-selective deep fading.

Link quality estimator nodes

Queiroz *et al.* [Que+18] propose a localized blacklisting technique called Adaptive Blacklist TSCH (AB-TSCH), applicable in star and tree topologies. The considered WSN is organized into groups (clusters) containing one parent (cluster head) and multiple child nodes. The radio link quality estimation and the construction of the blacklist are performed by a dedicated device called Link Quality Estimator node (LQE node), located near the cluster heads. The blacklists are disseminated through EB packets and they have a fixed length. Simulations of AB-TSCH for star topologies show that the best performance is achieved when the size of the blacklist is the maximum (15 channels). On the contrary, simulations for a tree topology show that there is a trade-off among the blacklist size and the network performance due to the simultaneous transmissions among certain nodes.

SFSB [Kru+19] inserts additional nodes (LQE nodes) to the network, where they are in charge of detecting possible interference in the channels by continuously sensing the spectrum. The nodes utilize the quality assessment of the channels by LQE nodes and construct their blacklist. They then transmit their blacklist to 1-hop neighboring nodes via control messages (6P signal). Nodes skip the cells in which their channel offset produces a physical channel that belongs either to their blacklist or to the blacklist of the other end of the radio link (SFSB

simple). This approach increases the delay and wastes the schedule's resources. For this reason, the authors propose a variant of the above technique (SFSB-extended) in which the two parts of a radio link, repeatedly increment cell's channel offset until a non-blacklisted channel for both parts is found.

2.3.4 Overview

Table 2.2 presents the main characteristics of the presented blacklisting/whitelisting algorithms so far.

As can be remarked in Table 2.2, most of the blacklisting proposals take into consideration the spatial characteristics of the quality of the radio links and, therefore the blacklists are constructed locally. Of course, the implementation and construction of a global blacklist can be efficient in the cases of small-area WSNs and one-hop networks such as [Tav+15; Tav+18], which are mainly intended for in-vehicle WSNs.

Among all the presented algorithms, only SmartHop [Ban+18] exploits a dynamic approach of channel classification where the channel hopping sequence is modified to use only the channels that present reliability close to the best channel's reliability. Furthermore, the repetition factor of the whitelisted channels on the channel hopping sequence depends on their estimated quality.

In chapter 4, we tackle the above problem by proposing a dynamic threshold algorithm. The most efficient blacklist/CHS dissemination techniques are those based on embedding the blacklist/CHS into the data packets [Li+15; Gom+17; Zor+18b; Zor+18a]. This approach can guarantee that both the receiver and the transmitter have the same blacklist/CHS since the data packets are acknowledged. Furthermore, an even more effective technique is that of MABO [Gom+17], where it implements a distributed blacklist negotiating technique based on Data Sequence Numbers (DSN).

The majority of blacklisting algorithms assess the available channels using *software-based* LQEs since they are easy to be implemented and can detect the multi-path fading effect, intra-network collisions, and the asymmetrical radio links [Gun+17; Zor+18a; Shi+15; Li+15; Gom+17; Zor+18b; Ban+18; Ham+].

2.4 Scheduling

Providing high reliability by combining IEEE 802.15.4-TSCH with a blacklisting technique into a WSN does not mean that low latency is provided at the same time. An appropriate organization of the transmissions is required to minimize the end-to-end delay. In other words, an efficient scheduling algorithm is necessary. The IEEE 802.15.4-TSCH standard does not specify how to build a TSCH radio link schedule, leaving the construction of the schedule to designers.

The scheduling algorithms can be classified in the following categories:

- **Centralized:** A central entity in the network (e.g., PCE) builds and distributes a common schedule by considering the gathered information by network's nodes such as network topology, network traffic, and routing tree;

- **Distributed:** Each node constructs its schedule based on the information exchanged with its neighbors;

- **Autonomous:** Nodes construct their local schedule, without the intervention of any central or distributed scheduling entity.

We here focus on the scheduling algorithms that try to minimize the end-to-end delay and to guarantee a minimum end-to-end PDR.

Algorithm	Blacklist Construction	Channel Quality Estimation	Channel Classification	Blacklist Dissemination
WirelessHART [Son+08]	Global			
ETSCH [Tav+15]	Global	Energy Detection	k-best channels	EB
ETSCH+DCS [Tav+18]	Global	CCA & packet reception		Data Packets
ETSCH+A-CQE [JC17]	Global	Energy Detection		
[Gun+17]	Global	PRR	k-best channels	
[Zor+18a]	Global	PDR	Fixed Threshold	Data Packets, Acks
ISA100.11a [An09]	Global/Local	CCA, ACK, RSSI		
[Shi+15]	Local	PRR	Fixed Threshold	Data packets
[Li+15]	Local	PRR with CCA		IE, packets
MABO-TSCH [Gom+17]	Local	PDR	k-worst channels	Data packets, ACKs
Lost [Zor+18b]	Local	PDR	Fixed threshold	Packets, Acks
SmartHop [Ban+18]	Local	Channel response	Dynamic Threshold	EB, IE
[Ham+]	Local	Retrans count, ACK, Goodness, Badness	Fixed Threshold	
A-TSCH [DR12]	Local	Noise Floor	k-worst, Fixed threshold	Downstrean: ADV Upstream: RPT
[Els+17]	Local	RSSI, PRR, Noise Floor	Fixed Threshold	EB downstream, Dedicated unicast messages
AB-TSCH [Que+18]	Local	LQE Node (RSSI, LQI)	k-best channels	EB
SFSB [Kru+19]	Local	Sensing Spectrum		1-hop 6P signal

Table 2.2: Blacklisting Whitelisting Techniques Overview

2.4.1 Centralized

Palattella *et al.* [Pal+12; Pal+13] propose a centralized traffic-aware scheduling algorithm (TASA). TASA aims to construct a compact schedule where all the packets generated by the network's nodes, are delivered to the sink with minimum latency. TASA uses a matching algorithm to assign timeslots to radio links by preferring the radio links with the largest amount of packets to transmit. Furthermore, it assigns channel offsets to the radio links using a vertex coloring algorithm, thus, the derived schedule is conflict-free. The performance evaluation of TASA through simulations showed that it is up to 80% more energy-efficient than IEEE 802.15.4 MAC pro-

tocol. TASA assumes that the reliability of the radio links is perfect, but this does not hold for real WSNs deployments. Thus, in case of a data packet failed transmission, the transmitter retransmits it in the next slotframe, increasing the delay.

Gaillard *et al.* [Gai+16a] propose an extension of TASA ($TASA_{rtx}$) which handles unreliable radio links by over-provisioning slots hop-by-hop for retransmissions. $TASA_{rtx}$ computes for each flow, the total number of cells in order to satisfy the expected end-to-end PDR. In particular, packets have to be retransmitted through unreliable links. However, the number of overprovisioned cells does not adapt to the radio link's quality temporal variations. The authors also take into account the fragmentation of the long packets to satisfy the end-to-end reliability constraints. Simulation results show that $TASA_{rtx}$ improves the reliability of the original algorithm.

Kausa [Gai+16b] is a centralized scheduling algorithm which copes with lossy links by allocating in the schedule ad-hoc opportunities. Furthermore, it aims to limit the buffer occupation and the end-to-end delay. Cell allocation is based on each data flow's Key Performance Indicators (KPI). Kausa estimates the minimum number of over-provisioning cells for every hop to satisfy both reliability and delay constraints considering also the radio link's quality. The allocation of cells is performed by a greedy algorithm. A backtracking procedure blacklists the most loaded and most vulnerable links and reiterates the process until ends to a valid schedule. The proposed algorithm, although satisfying the reliability and delay constraints, fails to respond on time to changes in a dynamic network consisting of unreliable connections.

Overall, centralized algorithms need a precise view of the network conditions and generate a large overhead when the schedule has to be updated.

2.4.2 Distributed

Random Cell Allocation

MSF [Cha+19] is one of the default scheduling functions of 6TiSCH. MSF provides two types of cells:

- *Autonomous (pseudo-random)*: Allocated by each node independently without any negotiation with the neighboring nodes;
- *Manageable*: Scheduled using 6P transactions.

Autonomous cells are used for exchanging unicast packets among a node and its neighbors, while the manageable (dedicated) cells are employed for data traffic. Nodes adapt to network traffic by adding/removing cells from their schedule according to the current usage of the cells towards their parent. Since the node's scheduling is entirely distributed, two neighborhood nodes' schedules can have the same 'managed' cell. In this case, when the nodes transmit on this cell at the same time, a 'schedule collision' is created. The nodes detect the 'schedule collisions' by monitoring the reliability (PDR) of the 'managed' cells towards their parent. Thus, if the PDR of a 'managed' cell is lower than the average one, it means that a collision has occurred. This approach requires a significant amount of packets to be transmitted before a node decides that a specific 'managed' cell belongs to another node's schedule too. MSF successfully adapts to changes of the radio link's quality by adding/removing cells from the schedule. However, the cells are randomly allocated in the schedule, which does not guarantee a low end-to-end delay.

Daisy chain

The minimization of the end-to-end delay may be achieved by reducing the buffering delay in each of the relay nodes along the path to the border router. The buffering delay is minimized by allocating the receiving timeslot and transmitting the timeslot of a packet as close as possible.

Accettura *et al.* [Acc+13; Acc+15] propose a decentralized traffic-aware scheduling algorithm (DeTAS) for 6TiSCH networks aiming to reduce end-to-end delay and to manage efficiently the queue size. In DeTAS, each node calculates the total sum of the packets it receives from its

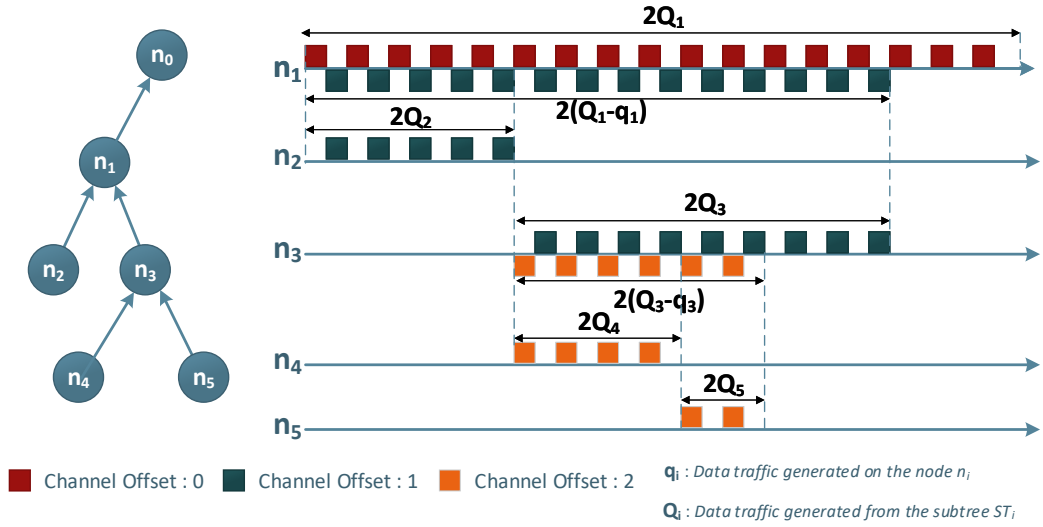


Figure 2.9: Odd-schedule.

children plus the number of packets it generates and forwards the above information recursively along the routing tree until it reaches the DODAG root. The DODAG root assigns to each of its children a sufficient number of consecutive even or odd receiving timeslots to receive the aggregated data traffic from their corresponding sub-trees. Then the nodes recursively downward of the routing tree allocate to their schedule alternatively reception/transmission timeslots to receive and forward their child packets thus minimizing the end-to-end delay (Fig. 2.9). In DeTAS the channel offset assignment is performed according to the distance in hops from the DODAG root and the channel offsets are reused only after W hops. DeTAS achieves the least possible end-to-end delay as reception and transmission cells are allocated in consecutive timeslots (Fig. 2.9). Nevertheless, it considers that the network's radio links are perfect, which is an unrealistic assumption.

LLSF [Cha+16] also exploits the daisy chain technique but additionally tackles unreliable radio links by adding over-provisioning cells. In order to reduce the buffering delay, whenever a node inserts a transmission timeslot to its schedule, it allocates it as close as possible to a receiving timeslot. In the case of multiple receiving timeslots, selects the receiving timeslot where its distance (in timeslots) from the previous receiving timeslot from the same neighbor is the maximum. Thus, in the example of Fig. 2.10, the node L inserts a transmission timeslot in the first available timeslot (19^{th}) immediately after the 17^{th} timeslot since its distance from the previous timeslot (10^{th}) is the maximum (6 timeslots). Similarly, during the removal of a transmission timeslot from a node's schedule, the transmission timeslot is selected, which is farthest from the immediately preceding receiving timeslot from the same neighbor. The main drawback of LLSF is that the relay nodes do not take into consideration the data traffic from different source nodes (data flows). Thus, the relay nodes are unable to allocate the transmission timeslots close to the reception timeslots of each of the data flows they forward.

Theoleyre *et al.* [TP16] propose a distributed algorithm for 6TiSCH to isolate the traffic through tracks. For each application a track is reserved, which is the set of cells that are reserved on each hop along a path towards the sink. The bandwidth estimation is performed locally for each track and depends on the track's number of packets in the queue and the Expected Transmission Count (ETX) of each track's reserved cells. Furthermore, the authors propose a cell allocation policy (contiguous) to minimize the buffering delay and they compare their proposal against the random policy.

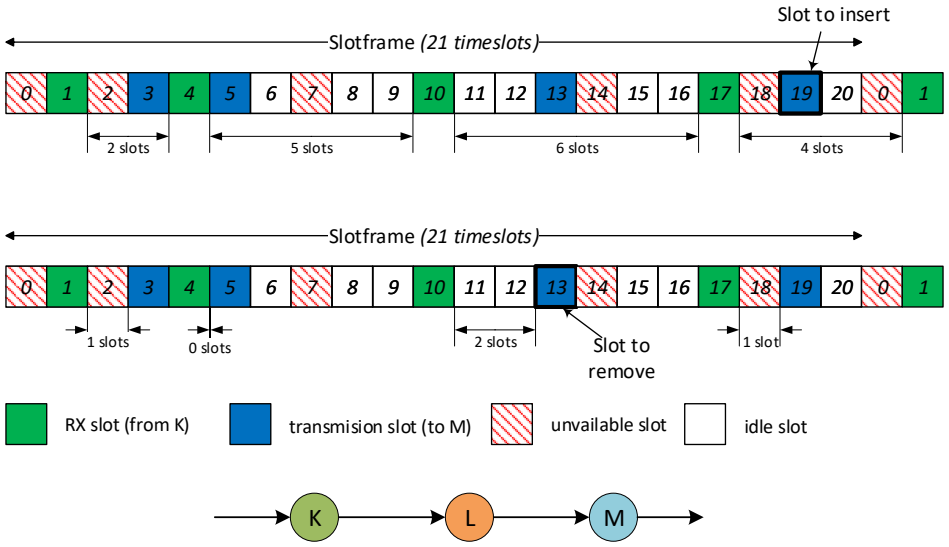


Figure 2.10: The TSCH schedule of node L when LLSF adds/removes a transmision slot to node M.

Daneels *et al.* propose the Recurrent Scheduling Function (ReSF) [Dan+18], which aims to achieve low end-to-end delay for WSNs that consider recurrent traffic. Under the assumption that each node is aware of its packet generation period, it reserves a series of receiving/transmitting timeslots back-to-back along the path from source to sink. The reserved timeslots are only activated in the slotframes where traffic is expected, thus, saving energy. ReSF addresses packet loss by adding additional slots depending on the quality of the utilized radio links. However, the ReSF reservations may cause schedule collisions, which occur when two or more reservations on a particular node use the same cell at the same time. Thus, ReSF tries to allocate the reservations in such a way to minimize the collision rate. Furthermore, ReSF does not guarantee a deterministic latency, because if the over-provisioning timeslots are insufficient to transmit a packet (e.g., due to the sudden degradation of a radio link), the data packet remains in the queue for an entire packet generation period. In [Dan+19], the authors extend the ReSF by providing an improved schedule collisions avoidance algorithm and supporting sporadic traffic.

Daisy chain cell allocation technique achieves lower end-to-end delay compared to random cell allocation technique. [Cha+16; Dan+18; TP16] considering also, a sufficient number of over-provisioning cells to deal with the unreliable radio links. However, over-provisioning cells increase the end-to-end delay and their possible reassessment due to changes in radio link quality cause chained cell's re-allocation in local schedules along the path to the border-router.

Consecutive Range of Cells

Another technique is to allocate all the receiving cells from a specific neighbor in a range of consecutive timeslots as well as to allocate the transmission cells immediately afterward in a range of consecutive timeslots. The number of cells in the range must correspond to the quality of the overlying radio links so that to guarantee the successful reception of the packets within the above range of cells.

Stratum scheduling [Hos+16] divides the network in stratums (Fig. 2.11), regrouping the devices by their hop distance from the border router. The algorithm associates each stratum with a contiguous region of the schedule (block) so that consecutive stratums are associated with consecutive blocks. Thus, the nodes of the same stratum allocate cells from the same

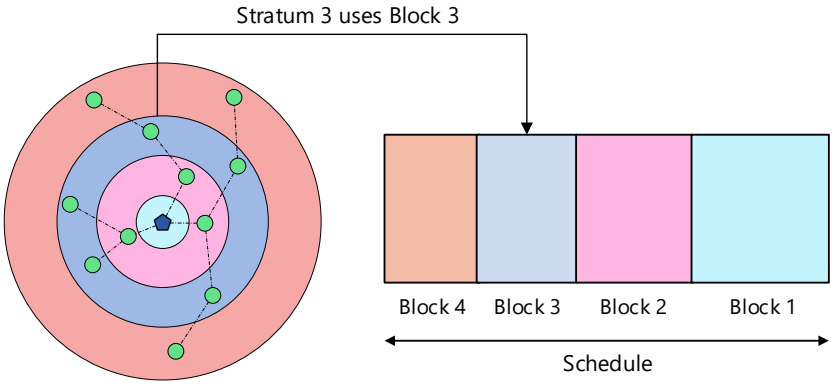


Figure 2.11: Dividing the schedule in Stratums.

block. This allocation strategy ensures that every packet is delivered before the end of the slotframe, even if the packet is retransmitted. The size of the blocks is not identical because the nodes close to the sink forward more data traffic, thus, the stratums near the sink should be larger. Hosni *et al.* [HT17] derive the right size of each block to minimize the collision probability based on the assumption that nodes are uniformly distributed in a given area and generate the same data traffic. Nodes (in order to adapt to the network traffic) add and remove cells from the schedule's block assigned to their stratum, implementing a localized scheduling strategy. Stratum provides a self-healing mechanism to detect and solve collisions since the cell assignment is performed randomly. According to the above technique, when a node detects that a cell exhibits a significantly lower PDR from the average PDR of all cells from the same neighbor, considers that there are repetitive collisions and reallocates the corresponding cell. However, the packet is guaranteed to be delivered at the end of the slotframe, which can correspond to a very long delay. Furthermore, stratum scheduling does not allow frequency re-use and increases the collision probability.

Minimize data converge cast delay

Soua *et al.* [Sou+16] propose a distributed conflict-free scheduling algorithm called Wave for IEEE 802.15.4e-based networks. Wave aims to minimize data converge cast delays by constructing a schedule with a minimum number of timeslots. In the first wave, the nodes allocate one transmission opportunity (cell) in their schedule by exchanging the appropriate control messages with their conflicting nodes. For the next waves, each node (based on the computations of the first wave) assigns, without the need of exchanging messages, the transmitting and receiving timeslots of its schedule. The process is terminated when all the generated data packets are delivered from the sink. Wave achieves to deliver any generated packet to the sink within a slotframe. The evaluation performance of Wave shows a similar performance with TMCP [Wu+08], which is a centralized scheduling algorithm. Wave does not take into consideration the unreliable radio links. As a consequence, the converge-cast delay is increased in case of failed transmissions.

PID based scheduling uses the well-known proportional, integral, and derivative algorithm to decide how many cells to reserve [DP+16]. In this way, additional cells are reserved when a burst of packets has to be delivered. Inversely, cells are maintained in the schedule to retransmit the packets if required in the future. The performance evaluation of the PID based scheduling through simulations shows that it could handle various types of data traffic by autonomously reacting to abrupt demand variations. In particular, PID scheduling aims to the stabilization

and minimization of cells in the schedule and the queue.

2.4.3 Autonomous

Duquennoy *et al.* [Duq+15] propose an autonomous scheduling algorithm called Orchestra to cope with the unpredictable network traffic and the frequent joining/leaving of nodes in a TSCH network. In Orchestra, nodes autonomously compute their local schedules without the intervention of any central entity and without exchanging messages with their neighborhood nodes. The time and channel offset of an assigned slot are derived from a hash function with input, the sender's or the receiver's identifier (MAC address or a unique network node ID). Orchestra defines three types of shared slots:

- *Common Shared Orchestra Slots (CS)*: They can be used from all nodes to transmit and receive data and they are also used to handle broadcast traffic.
- *Receiver-based Shared Orchestra Slots (RBS)*: The slots are dedicated to receiving packets a particular (node) receiver.
- *Sender-based Shared Orchestra Slots (SBS)*: The slots are dedicated for a node's transmissions.

Orchestra provides to TSCH the flexibility of an asynchronous MAC protocol that is able to handle random access traffic. However, Orchestra's slot allocation strategy is not flexible since the number of slots and their position on the slotframe is fixed. As a consequence, additional slots to handle the lossy links can not be inserted to the schedule, resulting in increased end-to-end delay and degradation of reliability.

Hwang *et al.* [Hwa+17] propose a distributed scheduling algorithm called DIS_TSCH to reduce the end-to-end delay. Every node is aware of its logical location in the tree topology (network location ID). This ID contains the depth of the node and its rank among the children of its parent node. DIS_TISCH allocates consecutive time slots to the nodes along the routing path towards the sink to minimize the end-to-end delay. The main assumption of DIS_TSCH is that RPL constructs a "perfect tree," which means that the tree is balanced, and all intermediate nodes have the same network degree. In the case where this assumption does not hold, there are unused timeslots and the network's resources are wasted. Futhermore, DIS_TSCH does not implement any cell over-provisioning mechanism.

Oh *et al.* [Oh+18] propose an autonomous scheduling algorithm for TSCH networks called Escalator based on RPL routing protocol [Ale+12]. The nodes exploit RPL control messages such as DIO and DAO to be aware of the set of their direct children and their preferred parent. Afterward, the nodes utilize the above local routing information to construct their schedule autonomously, without exchanging any control messages with their neighbors. In Escalator, the nodes allocate one receiving and one transmitting cell back-to-back to their local schedule for each of their children (direct or indirect). In this way, the minimization of packet's transmission end-to-end delay is achieved. Escalator's channel offset assignment algorithm is based on the rank of the radio link (receiver's hops from the sink) and the radio link's communication type (broadcast, unicast), guaranteeing the elimination of internal collisions. However, Escalator does not take into consideration the unreliable radio links and for this reason does not have any over-provisioning cell mechanism. Consequently, in the case of a packet transmission failure (e.g., due to external interference), the packet is retransmitted on the next slotframe, resulting in the considerable increase of the end-to-end delay.

2.4.4 Overview

Table 2.3 summarizes the prime characteristics of the scheduling algorithms that presented previously.

As can be remarked in Table 2.3, the presented scheduling algorithms address one of the IIoT requirements (the low end-to-end delay) by implementing a variety of cell allocation policies such

Algorithm	Category	Cells' allocation policy	Unreliable links
TASA [Pal+12]	Centralized	Compact	No
$TASA_{rtx}$ [Gai+16a]	Centralized	Compact	Yes
Kausa [Gai+16b]	Centralized	Compact	Yes
MSF [Cha+19]	Distributed	Random	Yes
DeTAS [Acc+13]	Distributed	Daisy chain	No
LLSF [Cha+16]	Distributed	Daisy chain	Yes
[TP16]	Distributed	Daisy chain	Yes
ReSF [Dan+18]	Distributed	Daisy chain	Yes
Stratum [Hos+16]	Distributed	Consecutive Range	Yes
Wave [Sou+16]	Distributed	Compact	No
PID [DP+16]	Distributed	Compact	No
Orchestra [Duq+15]	Autonomous	Random	No
DIS_TSCH [Hwa+17]	Autonomous	Daisy chain	No
Escalator [Oh+18]	Autonomous	Daisy chain	No

Table 2.3: An overview of scheduling algorithms.

as compact schedule, daisy chain and consecutive range of cells. However, managing the over-provisioning cells required to cope with unreliable links while keeping the end-to-end delay low is still an open issue.

There are two approaches in the literature to address the above challenge. In the first approach, the quality of the radio links is estimated for a specified period (learning phase). Consequently, each radio link's over-provisioning cells are calculated based on the radio links' estimated quality. Finally, the over-provisioning cells allocated in such a way to minimize the end-to-end delay. Obviously, such approach cannot combat the temporal variations of the radio links' quality. The centralized algorithms $TASA_{rtx}$ and Kausa follow the above approach. It should also be noted that even in the case of dynamic determination of over-provisioning cells, the process of updating the schedule from PCE would require a large number of control messages and would be both time and energy-consuming.

In the second approach, the nodes continuously assess (either directly or indirectly) the radio links' quality and dynamically adjust the required number of over-provisioning cells. The nodes then remove or add the appropriate number of the over-provisioning cells from their schedule in such way to optimize the end-to-end delay [Cha+16; TP16; Dan+18]. However, when a node inserts/removes over-provisioning cells in its schedule, the nodes towards the sink should modify their schedules appropriately. This process takes time to converge and requires additional control messages. Therefore, it is not convenient for networks with time-varying radio links.

We should note that the autonomous scheduling algorithms can achieve low end-to-end delay by allocating the cells in a daisy chain fashion [Hwa+17; Oh+18]. However, they cannot manage unreliable radio links since the schedule that they produce is static over time.

2.5 Conclusions

In this Chapter, we have set up the stage for the rest of this manuscript.

First of all, we performed a thorough literature review related to radio characterization. More specifically, we focused on studies where they exploit testbeds and real-deployments of WSNs to investigate the characteristics of the radio links.

Moreover, we made an overview of the existing WSN's communication protocols that exploit the channel hopping techniques to mitigate the external interference and multi-path fading effect. Consequently, we focused on the details of IEEE 802.15.4-TSCH since we selected it as the candidate MAC protocol to develop our solutions to provide dependability to WSNs.

A significant number of blacklisting/whitelisting techniques have been proposed and evaluated in several WSN scenarios. We classified the above techniques according to their main characteristics, and we presented their advantages and drawbacks.

Furthermore, in this Chapter, we performed a thorough literature review related to scheduling algorithms of a IEEE 802.15.4-TSCH network. Our main concern was to study the proposals that aimed to optimize both the end-to-end delay and reliability.

In the next chapters, we will present the contributions of this thesis, starting with the investigation of the necessity of the development of link-based adaptive blacklisting techniques.

Chapter 3

IEEE 802.15.4 Channels Characterization in an Indoor Testbed

The tremendous growth of the IoT, which exploits communication technologies that use the same frequency band ($2.4GHz$ ISM band), results in a large concentration of wireless devices in the same area, causing external interference (section 2.1.1). Another cause that heavily contributes to the unreliability of wireless radio links is the multi-path fading effect. The multi-path fading effect can be caused when the radio signal arrives at the receiver through multiple paths due to reflection of obstacles (section 2.1.4). In industrial wireless networks, the multi-path fading effect is magnified due to highly reflective structures such as metallic objects in the industrial environment [Che16].

In this chapter, we investigate whether slow channel hopping protocols can mitigate the external interference and the multi-path fading effect. Moreover, whether they can meet the requirements of the industrial wireless sensor networks for high reliability and strict and on-time delivery guarantees.

Towards this aim, we perform a thorough experimental study to characterize the radio (for all IEEE 802.15.4-2015 radio channels) and connectivity among the nodes of an indoor testbed. More precisely, we study in depth the spatial and temporal characteristics of the radio links. In particular, our results highlight the fact that the radio channel quality for the radio links is location-depend and varies over time. Therefore, enhancing the channel hopping technique by applying a blacklisting approach where the low-quality channels are excluded from the channel hopping sequence seems promising. Our study tends to justify the need for local blacklisting techniques, demanding more control packets, but dealing more efficiently with spectral re-use.

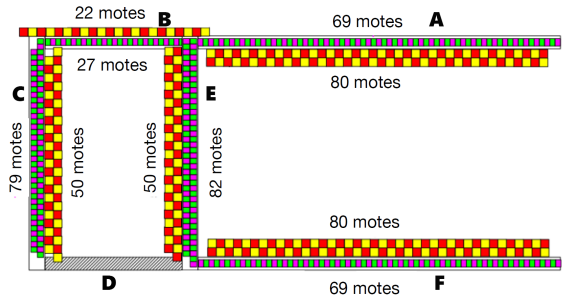


Figure 3.1: Grenoble FIT IoT-LAB testbed map.

Contribution

This chapter presents the following contributions:

1. We first experimentally study a TSCH network by employing the OpenWSN stack to characterize the radio link quality in an indoor environment, i.e., FIT IoT-LAB;
2. We then analyze the time variability of the characteristics of the radio link quality, and particularly the dependency on the physical channel;
3. We investigate the geographical dependency of the bad radio channel list;
4. We finally studied through simulations the efficiency of a blacklisting technique, and the temporal variability and the location-based heterogeneity of the blacklists.

3.1 Experimental Study

In this section, we present a thorough experimental study over the FIT IoT-LAB platform* that is part of the FIT†, an open large-scale and multiuser testing infrastructure for IoT-related systems and applications.

3.1.1 FIT IoT-LAB Platform: Grenoble site

In this investigation, our study was conducted over the testbed located in Grenoble (cf. Fig. 3.1). This testbed belongs to the real-world testbed category, since several Wi-Fi APs are deployed in the building. Under such a realistic indoor environment i.e., a typical office space, the nodes are subjected to external interference originated from wireless devices using other technologies, such as Wi-Fi (in the 2.4 GHz band).

As depicted in Fig. 3.1, this testbed consists of 380 nodes deployed in an area of 65 m $\times$ 30 m. Most of the deployed sensor nodes (i.e., 90%) are placed under the raised floor, while the remaining 10% are deployed above the dropped ceiling.

3.1.2 Experimental Setup and Parameters

In our experimental study, we employed M3 nodes, based on a STMicroelectronics 32-bit ARM Cortex-M3 micro-controller (ST2M32F103REY) that embeds an AT86RF231 radio chip, providing an IEEE 802.15.4 compliant PHY layer.

*https://www.iot-lab.info/
†https://fit-equipex.fr/

Table 3.1: Experimental setup.

Topology	**Parameter**	**Value**
	Testbed organization	Grenoble site
	Number of nodes	2
	Number of Experiments	200
	Link Distance	$[0.6 - 17]$ *meters*
Experiment	**Parameter**	**Value**
	Duration	90 *min*
	Payload size	48 bytes
Protocol Stack	**Parameter**	**Value**
CoAP	CBR (*Unicast*)	1 *pkts*/3 *sec*
RPL	DAO period	50 *s*
	DIO period	8.5 *s*
TSCH	Slotframe length	101
	NShared cells	5
	Timeslot duration	15 *ms*
	Maximum retries	3
Queues	Timeout	8 *s*
	Queue size	10 packets
	incl. data packets	Maximum 6 packets
Hardware	**Parameter**	**Value**
	Antenna model	Omnidirectional
	Radio propagation	2.4 *GHz*
	802.15.4 Channels	11 *to* 26
	Modulation model	AT86RF231 O-QPSK
	Transmission power	0 *dBm*

We focused on a scenario with two M3 nodes, a transmitter and a receiver, respectively, positioned in a distance that varies from 0.6 to 17 *m*. In particular, at each experimental round, we selected randomly two different M3 nodes (out of 380) in the testbed to achieve maximum pluralism and transparency in our performance evaluation. Other nodes may be reserved for concurrent experiments by other researchers, and may generate external interference. We implement a CBR traffic (20 packets / min), at 0 *dBm* transmission power, resulting in more than 1800 *pkts* transmissions in total. We utilize a 48 *byte* data size, which corresponds to the general information used by monitoring applications (e.g., node ID, packet sequence, sensed value). We use the default TSCH and 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4e) configurations as depicted in Table 3.1. We performed a thorough analysis of the radio links by iterating the previously presented set of experiments over all IEEE 802.15.4 channels from 11 to 26. Finally, we ran more than 200 experiments, while each experiment lasted for 90 *min*. The details of the setup are exposed in Table 3.1.

To conduct our experiments, we employed OpenWSN[‡], an open-source implementation of a full protocol stack based on loT standards (IPv6, 6TiSCH, RPL, CoAP). In particular, we used the modified implementation of OpenWSN[§] to handle tracks and to provide distributed scheduling [TP16].

In this study, we kept our experimental setup as simple as possible, in order to focus on the actual performance of the open testbed. Hereafter, we detail the results obtained from our experimentations, in terms of radio link quality characterization, stability of the radio links in

[‡] `https://openwsn.atlassian.net/`

[§] `https://github.com/ftheoleyre/openwsn-fw/`, and `https://github.com/ftheoleyre/openwsn-sw/`

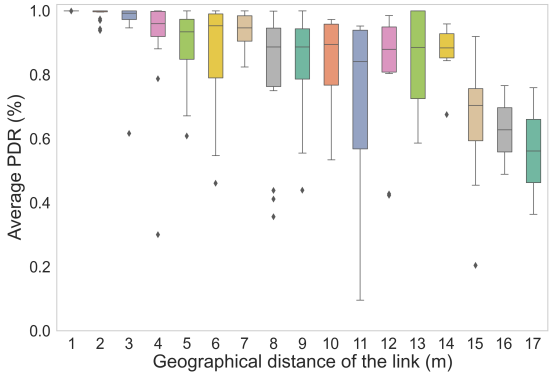


Figure 3.2: PDR versus distance from the source.

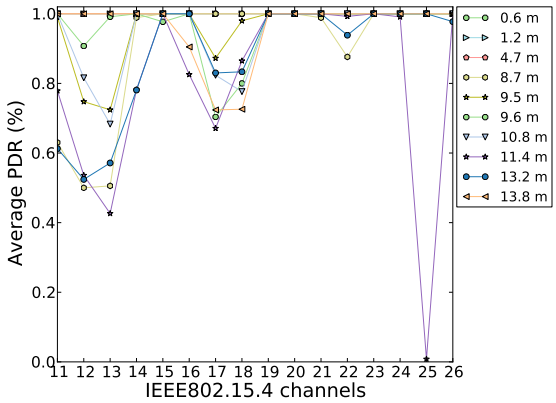


Figure 3.3: PDR through all IEEE 802.15.4 channels and over various distances (i.e., $0.6 - 13.8\ m$).

time as well as channel characterization.

3.2 Radio Link Quality Characterization

In this section, we investigate the impact of bad channels, due to external interference, on the performance of the system when a channel hopping approach is implemented.

3.2.1 Radio Link Reliability

We first measured the PDR for all pairs of nodes that were randomly selected. We then grouped the pairs that provide approximatively the same geographical distance, (i.e., more or less 1 *meter*). As it can be observed from Fig. 3.2, short distance radio links (< 3 *meters*) present very high link quality performance (i.e., close to 100%). Because the transmission power remains constant and the signal strength is high and, thus, limiting the number of errors of transmission. As a result, no blacklisting technique is required for such links.

On the contrary, the longest distance radio links present a very dynamic behavior. In particular, we can observe a straightforward relation between distance and link quality; if the distance between two nodes is longer, their PDR performance significantly drops, while the link quality discrepancy considerably increases. Thus, due to this strong variability, the long distance links need further investigation.

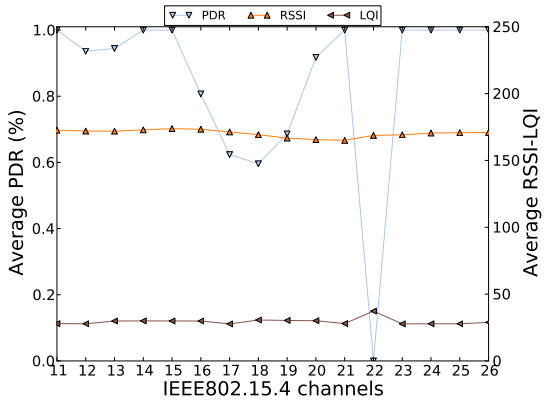


Figure 3.4: Link Quality Indicators for the link with distance of 13.2 m.

To this aim, we analyzed the PDR performance for all IEEE 802.15.4 radio channels illustrated in Fig. 3.3. As can be observed, the IEEE 802.11 channels that perform worse correspond to the most commonly used by Wi-Fi enabled devices (cf. Fig. 2.1).

Moreover, it is worth mentioning that not all radio links suffer similarly from external interference. In particular, while many links perform badly on channels 12 and 13, some others (e.g., 1.2 m, 4.7 m) still achieve a perfect reliability (100%). Indeed, short distance links tend to be less sensitive to external interference. Their signal strength may be higher and, consequently, these radio links are more robust.

3.2.2 Accuracy of the Link Quality Indicators

To further characterize the links, we focus on a single radio link (i.e., distance of 13.2 m, Fig. 3.4). RSSI and LQI serve as link quality indicators, since the level of the received signal is correlated with the Bit Error Rate (BER). However, these link indicators do not reflect here the actual PDR for each channel.

Indeed, RSSI and LQI can only be measured for correctly decoded packets. With the presence of external interference, some of the packets are corrupted and, thus, are not received correctly. While these dropped packets have an impact on the PDR, RSSI and LQI of the received packets remains unchanged. Thus, hereafter, in order to detect external interference, we explicitly focus on the estimation of PDR.

We proceed one step further in the investigation of the correlation of PDR and RSSI by analyzing the measurements from the previously conducted experiments. Thus, we analyzed the data we collected using the Pearson correlation coefficient [Ben+09], where it measures the linear correlation between two stochastic variables and is defined as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)} \tag{3.1}$$

where $\sigma(X)$ is the standard deviation of the stochastic variable X, and $cov(X,Y)$ is the covariance of the variables X and Y. The value of the Pearson correlation coefficient is a number in the range of -1 to 1, where -1 denotes the perfect negative linear correlation, 0 indicates no correlation between X, Y, and 1 denotes the perfect linear correlation.

RSSI has been proved to reflect very loosely the link quality. Let us consider Fig. 3.5 that illustrates the Pearson's correlation coefficient for the smoothed average PDR and RSSI for the different radio channels of each link. We use the WMEWMA filter (Window size = the last 16 transmitted packets, $a = 0.6$) to smooth the estimated values of the RSSI and PDR and take into consideration only the most resent values. The Pearson's correlation coefficient value for the links with the highest PDR is close to 1, denoting a strong correlation between the PDR and

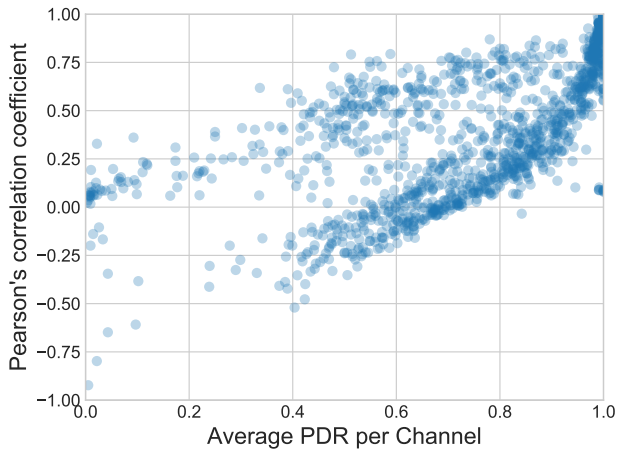
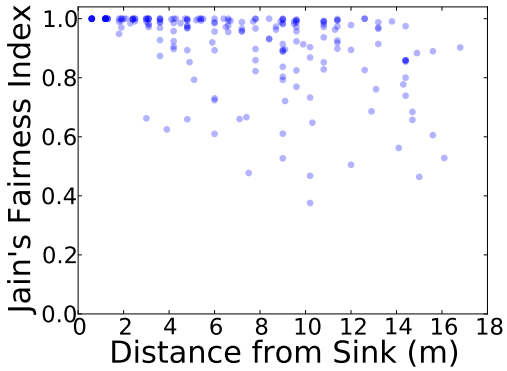
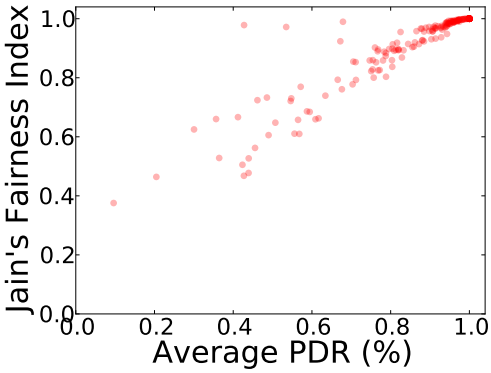


Figure 3.5: Correlation between RSSI and PDR



(a) Impact of the distance on the Channel Jain Index



(b) Channel Jain Index and average PDR correlation.

Figure 3.6: Fairness among the different channels

RSSI metrics. However, medium channel quality links exhibit very heterogeneous behaviors and we are unable to predict the PDR of one specific radio channel by only measuring the RSSI.

3.2.3 PDR Fairness among Channels

We now investigate the variability of bad links and we ask ourselves the following question: is PDR similar for all the physical channels or most packets are dropped because of external interference on *some* of the channels?

To quantify fairness, we measured the Jain Index of the PDR for all the channels. Thus, we define the *Channel Jain Index* of a link l as follows:

$$ChannelJainIndex(l) = \frac{\left(\sum_{c \in \mathcal{C}} Avg_{PDR}\,(c,l)\right)^2}{|\mathcal{C}| * \sum_{c \in \mathcal{C}} Avg_{PDR}(c,l)^2} \tag{3.2}$$

with $\mathcal{C}$ being the set of channels and $Avg_{PDR}(c,l)$ the average PDR for the link l on channel c.

Fig. 3.6a illustrates the distribution of the Jain Index of the different links according to their euclidean length. This result corroborates our observation about the variability; the links in the

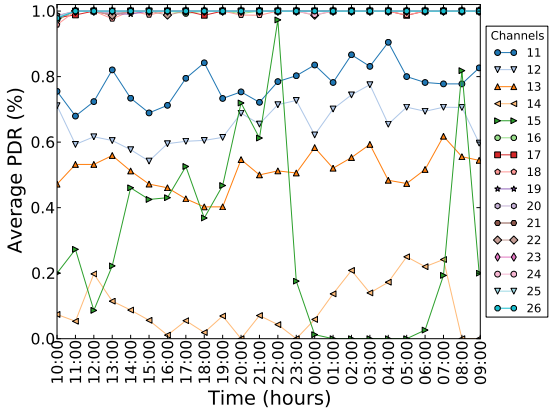


Figure 3.7: The variability of the link quality over time: studied case of 9 m distance.

gray zone exhibit very different characteristics. In particular, some radio links may perform very differently on all the channels: external interference is present on *some* channels, which implies a bad Channel Jain Index. Furthermore, the distance of radio links seems also correlated with fairness since the probability for a given link to behave differently on the different channels is higher for longer distance links.

Fig. 3.6b illustrates the strong correlation between PDR and fairness. Surprisingly, bad radio links indicate very strong unfairness. In other words, radio links with a low average PDR suffer from packet drops unfairly on *some* channels. Thus, most links with a bad PDR exhibit a very high channel variability. Our conclusion is that blacklisting the bad channels may help them to improve their average link quality.

3.3 Time Variability Characterization

We then studied the time variability of the link quality. Indeed, we performed an experiment during 24 hours, where 6 M3 nodes transmit to one single receiver, in a 1-hop star topology with different distances.

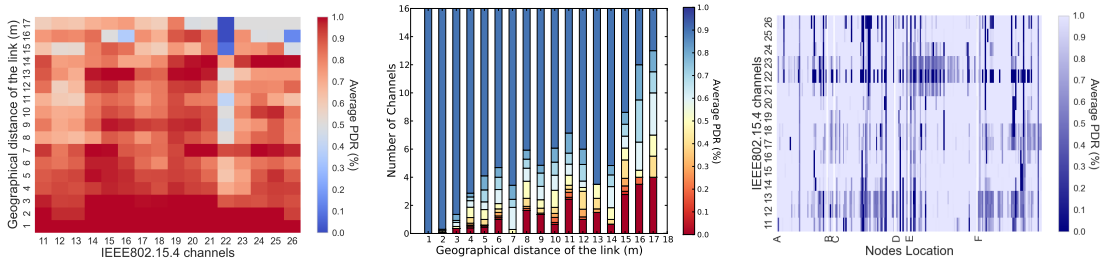
We identified two classes of links (stable *vs.* variable). Due to lack of space, we do not provide the graphs for stable, good links in which all channels perform similarly with very high PDR.

We actually focused on a scenario that considers link distance of 9 m (Fig. 3.7). More specifically, some of the physical channels perform very well and are very stable (e.g., channels 17, 22, 24, 26), while some other exhibit a very high variability. Furthermore, it is worth mentioning that for example channel 14, should be blacklisted globally, since it provides a bad PDR during the whole experiment (i.e., 24 h). On the contrary, channel 11 could be utilized during hours 02 to 04 and blacklisted between hours 14 to 16.

These results **advocate the relevance of a dynamic blacklisting method** in which the network must reactively discover the bad channels and should recover when a channel restarts to perform accurately. Moreover, links seem relatively stable for long periods (i.e., 1 h) and justify the decision to only temporarily blacklist a channel.

3.4 Spatial Variability Characterization

In this section, we study the relevance of the different blacklisting techniques. Thus, we first measured the PDR through each physical channel for different pairs of nodes selected randomly. We then grouped the pairs with similar geographical distance. Fig. 3.8a illustrates the heat-map of the PDR for different channels and links of a similar quality.



(a) Heatmap: average link quality through each channel depending on the radio link geographical length

(b) Classification of the link qualities for different geographical lengths

(c) Heatmap of the PDR for each channel, links being grouped by their geographical location

Figure 3.8: Variability of the list of *bad* channels.

As it can be observed, channel 22 performs badly for almost all radio links. For instance, the PDR of long links (11m) is reduced by 50%, compared to the channel 19 with a reliability over 95%. However, even this channel should not be blacklisted globally, since the shortest links keep achieving a perfect PDR performance. As a result, blacklisting a channel globally decreases network capacity vainly by $\frac{100}{16}$%.

Furthermore, we can isolate some local patterns. For instance, channels 15 and 16 provide a low PDR only in *some* locations (i.e. a few radio links present in a given geographical area have a low PDR for this channel).

Fig. 3.8b illustrates the amount of bad/good channels depending on the geographical distance between the transmitter and the receiver. If the distance of the receiver is higher, the unfairness becomes more intensive. We also observe that when the distance is equal to 17 meters, some channels perform very well, while the other ones provide a very low reliability. Thus, this behavior **advocates the usage of a local blacklist**.

Alternatively, the controller may blacklist a channel in a given geographical area. We measured the PDR for each channel according to the location of the links (Fig. 3.8c). We can remark a semi-global pattern; channel 22 performs badly for a set of radio links, wherever they are located. However, it seems to impact only the weakest links. On the other hand, we can isolate some additional local patterns: in the corridor EF, a few channels seem more perturbed by external interference (channels 21-24). However, the rest of the channels perform on average better than in the other corridors.

3.5 Why is blacklisting still required for channel hopping?

One may argue that radio channel hopping is sufficient since channel diversity allows to reduce the number of repetitive failures. However, not all channels exhibit similar characteristics. In that case, over-provisioning may improve the end-to-end reliability, but it impacts negatively the energy consumption and the network capacity. Blacklisting enhances the channel hopping technique to improve further the reliability of the radio links. In blacklisting, the channels that exhibit the poorest quality are excluded from the channel hopping sequence; thus, the high-quality channels are used in transmission, improving the radio links' reliability (cf. section 2.3). Hereafter, we will provide experimental results to defend the relevance of blacklisting techniques to avoid these unnecessary retransmissions.

3.5.1 Experimental dataset

To study the necessity of a blacklisting technique in a Channel Hopping MAC protocol, we collected a large dataset of measurements from the experiments (section 3.1.2) that conducted

to Fit-IoT-LAB to emulate real link qualities. We store the packet success / failure for 267 radio links, with one packet every 3 seconds, during 90 min. The distance between the transmitter and receivers varies from 0.6 to 17 m (Table 3.1). We have co-located Wi-Fi, and other concurrent experiments, which generate external interference. We inject this dataset in a custom made Python simulator, to decide if a packet is received or dropped because of external interference. We chose this methodology against real experiments to compare fairly the blacklisting techniques over the same radio conditions. We focus here on the efficiency of blacklisting in single hop topologies.

Several techniques have been proposed in the literature to construct a blacklist with a fixed [Chi+16; Shi+15] or variable (cf. chapter 4) size. For the sake of simplicity, we focus here uniquely on fixed-size blacklists, utilizing the two following strategies:

1. **k-Worst Channels:** This blacklisting technique excludes from the channel hopping sequence the k-worst radio channels with the poorest Packet Delivery Ratio (smoothed with a WMEWMA estimator [Shi+15]).

2. **Default:** We do not exclude any radio channel from the channel hopping sequence (equivalent to k=0).

3.5.2 Blacklist Efficiency

We first measure the Cumulative Distribution Function (CDF) of the ETX value for all the links when blacklisting a different number of radio channels (Fig. 3.9a). The ETX metric counts the average number of packets to transmit before receiving an acknowledgement. Without blacklisting (k=0), ETX is high, denoting retransmissions; some radio channels perform badly and impact significantly reliability. On the contrary, blacklisting automatically removes the bad radio channels from the frequency hopping sequence, thus, we need less retransmissions on average. This improvement has a counter-part: the network capacity is reduced since the network can only exploit a smaller number of radio channels.

3.5.3 Blacklist changes

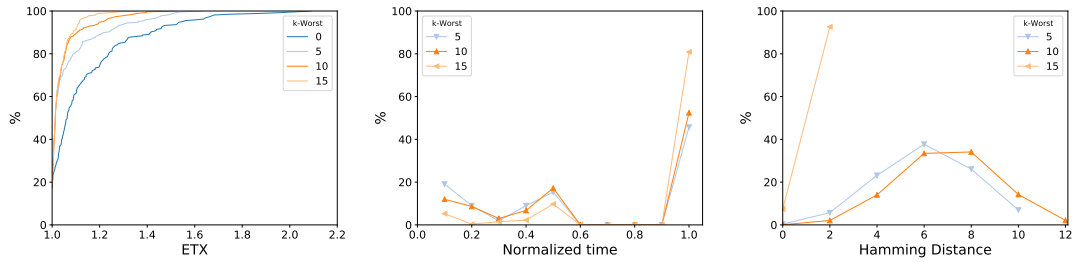
In a global blacklist scheme, the controller typically collects continuously the link quality metric to cope with time-variable conditions. If the radio channel quality changes significantly, the blacklist is updated and pushed to all the nodes. Similarly, per-link blacklists may change if the PDR per radio channel evolves. Then, the transmitter has to notify the receiver of the novel blacklist.

Thus, in Fig. 3.9b we plot the CDF of the average time duration before a blacklist changes. For instance, less than 20% of the links have an average blacklist duration below 9 min (10% of the experiments) when blacklisting the 5 worst radio channels. Longer blacklists tend to be more stable. Besides, most of the links have very stable blacklists which reduces the number of control packets to generate.

3.5.4 Location-based Heterogeneity

We explore the location-dependent characteristics of the different blacklists, by comparing pairwisely the blacklists of different links (see Fig. 3.9c). We use the Hamming Distance, counting the number of positions where the bits differ for a pair of binary strings. Here, we associate a 16-bit string to each link, the i^{th} bit being set to 1 if the radio channel i is blacklisted. In our case, the Hamming distance counts the number of radio channels which differ in the two blacklists.

We note that the Hamming distance is an absolute metric. In particular, two very long blacklists (e.g., 15) can only differ by one radio channel, leading to an Hamming distance at most equal to 2. However, a global blacklist would be inefficient even in that case: selecting randomly one pair of radio link, we have a 90% probability that the best radio channel differs.



(a) Efficiency: ETX value for a given radio link.

(b) Stability: average duration before a blacklist changes.

(c) Heterogeneity in the blacklists.

Figure 3.9: Impact of a blacklist which contains the k worst channels on the performance.

Blacklists with 5 radio channels are very different: 50% of pairs of links share less than one half of the radio channels. In other words, the blacklist is very location dependent and, thus, radio links have different blacklists. Consequently, relying on the same blacklist in the whole network would be suboptimal.

3.6 Conclusions and Perspectives

Our experimental research campaign has highlighted that the quality of the radio channels changes over time. More specifically, we can distinguish radio channels where their quality remains stable over time and radio channels that show intense fluctuations in their quality over time. Furthermore, the analysis of the experimental data showed a strong dependence between the quality of the radio channels and the spatial characteristics of their radio links. We have shown experimentally that there is no correlation between the RSSI value of the received packets with PDR, especially in the medium and low-quality radio channels. Therefore, RSSI is considered inappropriate for the estimation of the quality of a radio link's radio channels.

We then have studied the characteristics of a possible blacklist (i.e., global versus local) and of bad radio channels for indoor environments. Indeed, a slow channel hopping MAC helps to combat external interference, limiting consecutive packet drops. However, the radio channels that always perform *bad* should be blacklisted. Based on our experimental results, we highlighted local characteristics in which some radio channels perform poorly only for a subset of the radio links. The signal strength and the location of external interference impact significantly the list of radio channels that perform badly. In conclusion, the list of blacklisted radio channels should be probably localized, specifically for a zone or a radio link.

Our simulations depicted that even when using short blacklists, the improvement of reliability is significant. The long blacklists achieve the most considerable improvement in reliability and are more stable over time but sacrificing the network capacity. Another argument beyond developing a local blacklisting technique is the outstanding location-based heterogeneity observed between blacklists of any size. The observed stability of the blacklists indicates that a possible application of a blacklist technique will not require the dissemination of an excessive number of messages to adapt the blacklists to temporal variations. Consequently, the side effect of inconsistent blacklists will be limited.

In the future, we plan to extend our experimental evaluation by also considering outdoor testbeds and WSN's deployments in industrial environments.

Our research has shown that enhancing the channel hopping technique by applying a blacklisting technique is essential and improves the overall performance significantly by successfully tackling external interference and the multi-path fading effect. In particular, the blacklisting technique should be applied locally to each radio link taking into consideration the spatial characteristics of the radio links as highlighted in our research. It is also crucial that the blacklist be adapted quickly to the channels' quality variations over time. Finally, the blacklisted channels

of a radio link should be determined solely by the transmitter and the receiver of the radio link to avoid network congestion and deafness between the radio link's nodes. Taking into account all the above in the next chapter, we propose and evaluate a link-based adaptive blacklisting technique intended to IIoT.

Chapter 4

Distributed Link-based Blacklisting

In the previous chapter, we investigated the characteristics of a low power lossy network in an indoor environment. In particular, we highlighted the existence of radio channels which provide a very low reliability. Thus, in this chapter, we propose to investigate how we can use a blacklisting technique to improve the reliability and the energy efficiency. Blacklisting techniques identify the *bad* radio channels to avoid using them to transmit data packets. In this way, we reduce the number of transmissions, with a positive impact on both the reliability and the duty-cycle ratio.

Indeed, our experimental evaluation from Chapter 3 highlights a highly localized external interference. Thus, in this chapter, we need to propose a per radio link blacklisting technique.

The proposed distributed blacklisting technique is flexible to network dynamics, and is able to respond directly to variations in radio channel quality by adapting the blacklist accordingly, while the overhead to the network traffic is limited. In our distributed approach, the transmitter independently assesses the available radio channels and dynamically selects the best of them to be used in the channel hopping sequence. The transmitter and the receiver must also agree on a consistent blacklist to avoid deafness. Our thorough experimental evaluation based on OpenWSN and FIT IoT-LAB highlight the relevance of this approach: with a localized blacklisting strategy, we increase by 20% packet delivery rate for the worst links.

Contribution

This chapter presents the following contributions:

1. We provide an algorithm to determine dynamically which radio channels to blacklist. A set of *bad* radio channels is identified for each radio link. Since we do not exploit a *fixed* threshold value, we are able to identify bad channels even for weak links;
2. We present a method to passively probe the bad radio channels, while limiting their impact on the energy consumption and the reliability;
3. By exploiting 6P control packets, we detail techniques to maintain consistent blacklists for both the transmitter and the receiver and, thus, to avoid deafness (i.e., the receiver and the transmitter listen to different radio channels);
4. We propose a method to modify the frequency hopping sequence. In this way, we make the collisions not repetitive, when two radio links use the same timeslot with a different channel offset and different blacklists;
5. We experimentally validate our approach in the FIT IoT-LAB indoor testbed with the OpenWSN stack.

4.1 Problem Statement & Approach

External interference may severely affect some IEEE 802.15.4 channels [Wat+09], requiring to blacklist the *bad* channels. However, the performance of a given physical channel depends heavily on the geographical location, and even on the link's characteristics (cf. chapter 3).

We propose here to implement a *link-based* blacklisting algorithm, i.e., LABeL: the transmitter and the receiver have to agree on the blacklisted channels not to use for their transmissions. Different pairs of nodes would blacklist different channels resulting in increased frequency reuse. More specifically, each pair monitors the link quality across all the 16 available channels at 2.4 GHz, and decides which channels to utilize. Consequently, in this study, we focus on addressing the following challenges:

Overhead: We here implement a passive method to detect *bad* channels. No probing packets are required, increasing both the level of interference and the energy consumption. Instead, we use the data packets to continuously re-evaluate the quality of channels in order to appropriately insert or remove from the blacklist;

Time-variant: Under dynamic environments, the list of bad channels may change so frequently that blacklisting would have no effect on the performance (cf. chapter 3). Control packets have to be exchanged to update the blacklist, which would annihilate the benefit of reducing the number of (re)transmissions to deliver a data packet to the next hop. We experimentally verify that the PDR is actually improved with a localized adaptive blacklisting approach;

Inconsistency management: Two nodes agreeing on the list of bad channels, requires signaling (i.e., additional control packets). Since some control or acknowledgement packets may be lost, some inconsistencies may arise. As a result, they may operate with a different frequency hopping sequence, leading to potential deafness. We will propose robust mechanisms integrated to 6P in order to make the transactions reliable.

Minimization of collisions: When two interfering radio links use a different blacklist, they may collide even if they do not use the same channel offset, since Equation 4.2 depends on the blacklist's content (i.e., the number of available channels). We propose to modify the frequency hopping sequence to make the collisions less repetitive.

In this chapter, we both propose the mechanisms to implement a link-based blacklist, and we evaluate thoroughly the blacklisting technique in a realistic testbed to demonstrate the advantages of such approach.

4.2 Localized and Per-Link Adaptive Blacklisting under IEEE 802.15.4-TSCH

A global blacklist exploits a list of *bad* channels that provide a low reliability due to the presence of interference or a low channel gain (e.g., due to multi-path fading effect). However, this list is location and time-dependent(cf. chapter 3): while a channel may perform badly for some radio links, it may provide a close to perfect reliability for some other radio links. Moreover, the same radio channel may perform well during the afternoon and night, however, its performance may drop during the day-time, due to the Wi-Fi activity.

The impact of external interference depends on the Signal to Interference plus Noise Ratio (SINR) margin of the radio link [Gol05]. When the transmitter and the receiver are close to each other, or the channel gain is good, external interference has to be higher to impact the reliability. Thus, we here present an algorithm to incorporate a **localized** and **per-link** blacklist into IEEE 802.15.4-TSCH.

4.2.1 Deciding which channels to blacklist

In this study, we propose LABeL to identify the channels to blacklist, i.e., the set of channels that impact negatively the performance of the radio link and/or the network. According to our previous work, relying on RSSI or LQI metric is not representative of the channel quality (cf. chapter 3). Therefore, we focus on measuring the PDR performance, denoting accurately the ability of the link to deliver successfully the data packets.

To this aim, each node in a TSCH network computes the PDR of unicast data packets **independently** for each neighbor and channel. More precisely, a node counts the number of Acknowledgements (ACKs) and the number of packets transmitted to a particular neighbor N. Since we are interested in a per channel behavior, we compute this PDR value independently for each channel and neighbor:

$$PDR(N,c) = \frac{nb_{ack}(N,c)}{nb_{tx}(N,c)} \tag{4.1}$$

with $nb_{ack}[N,c]$ the number of ACKs received from N through the channel c, and $nb_{tx}[N,c]$ the number of packets transmitted to N.

Most of the proposals use a fixed threshold value (e.g., [Hän+11], [Sha+11]): any radio channel that provides a PDR inferior to a pre-defined threshold value is blacklisted. However, the *average* PDR is very radio link-dependent: when the received signal strength is low, packets may be dropped even if no external interference is present. Low quality links are frequent in many deployments, while high quality links are often not sufficient to maintain a connected topology [Liu+12]. We have consequently focus on an **adaptive** approach in which this threshold depends on the link, and is not fixed a priori globally.

The WMEWMA has been proved to accurately estimate the link quality [Bac+12]. Indeed, packet losses represent a stochastic variable and need to be *smoothen*. We consequently propose to use WMEWMA to independently measure the PDR for each channel. For this sake, a node counts the number of transmitted messages, and the number of acknowledgments received correctly. In

Algorithm 1: Blacklist construction

Data: *blacklist* (current blacklist),
$nb_{tx}[CH]$ and $nb_{ack}[CH]$ (nb. of transmitted packets and received ACKs over each channel)
α (WMEWMA's parameter)
$\mathcal{T}$ (threshold to consider a channel bad)
Result: *blacklist* (new list of bad channels)

1 // PDR for each channel
2 $best \leftarrow 0$;
3 **for** $c \in Channels$ **do**
4 | // WMEWMA of the PDR with the last 16 transmitted packets
5 | $PDR_{last16} \leftarrow \frac{nb_{ack}[c]}{nb_{tx}[c]}$;
6 | $PDR_{wmewma}[c] = \alpha PDR_{wmewma}[c] + (1-\alpha)PDR_{last16}$;
7 | // Remembers the PDR of the best channel
8 | **if** $best \leq PDR_{wmewma}[c]$ **then**
9 | | $best \leftarrow PDR_{wmewma}[c]$;
10 | **end**
11 **end**
12 // Adaptive Threshold Calculation
13 **repeat**
14 | $numch \leftarrow 0$;
15 | $weight \leftarrow weight - 0.01$;
16 | $\mathcal{T} \leftarrow best * weight$;
17 | **for** $c \in Channels$ **do**
18 | | **if** $PDR_{wmewma}[c] < \mathcal{T}$ **then**
19 | | | $numch \leftarrow numch + 1$;
20 | | **end**
21 | **end**
22 **until** $numch \geq 3$;
23 // threshold PDR to define which channels perform significantly worse than the best one
24 $PDR_{th} \leftarrow \mathcal{T} * best$
25 // For each channel, verifies it performs similarly to the best one (or not)
26 **for** $c \in Channels$ **do**
27 | // To blacklist
28 | **if** $PDR_{wmewma}[c] < PDR_{th}$ **and** $c \notin blacklist$ **then**
29 | | $blacklist \leftarrow blacklist + \{c\}$;
30 | **end**
31 | // To recover
32 | **if** $PDR_{wmewma}[c] > PDR_{th}$ **and** $c \in blacklist$ **then**
33 | | $blacklist \leftarrow blacklist - \{c\}$;
34 | **end**
35 **end**
36 **return** blacklist;

this chapter, each node computes the PDR for the last 16 transmitted packets for a given channel, and updates accordingly the smoothed PDR metric.

Algorithm 1 describes formally LABeL, our link-based and adaptive blacklisting approach. We first compute the average PDR of each channel independently, using the extended WMEWMA estimator (lines 3-4). Then, we identify the best channel, providing the highest PDR (lines 5-7), which allows us to define a dynamic PDR threshold value PDR_{th} to identify bad channels (lines 9-19). Note that we dynamically adapt PDR_{th} in order to maintain at minimum 3 whitelisted

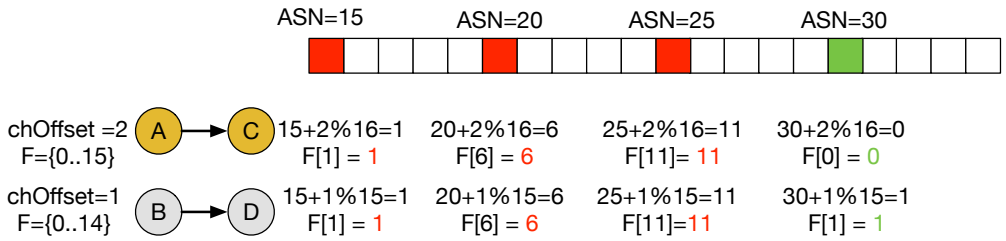


Figure 4.1: Colliding cells which use a different channel offset and different blacklists (F[] denotes the set of *good* channels).

channels on each wireless link. Then, we update the blacklist. In particular, a given channel is considered as bad if it provides a PDR lower than PDR_{th} (lines 21-23). Inversely, a channel is removed from the blacklist if its PDR metric significantly exceeds the threshold value (lines 24-26).

Note that constructing a link-based blacklist requires only for the transmitter to collect the ratio of acknowledged packets. In particular, the blacklist considers both directions, for respectively the data packet and the acknowledgement transmissions. Thus, computing the blacklist does not need to send explicit control and probe packets, and does not generate any overhead. Note that the blacklist is updated continuously, i.e., at each data transmission, while 6P control packet is exchanged, only when the blacklist is modified.

4.2.2 Modifying the frequency hopping sequence

After identifying the blacklisted radio channels, we next have to exploit this blacklisting mechanism with TSCH. In particular, the employed physical channel is decided at the beginning of each cell, using Equation 4.2.

$$frequency = F\Big(\big(ASN + channelOffset\big)\ \%\ nFreq\Big) \tag{4.2}$$

Note that ISA100.11a [PC11] proposes to use a localized blacklist. A node follows the frequency hopping sequence. However, when the transmitter detects that the physical channel associated to a cell is blacklisted, it postpones its transmission (i.e., for the following slotframe, 101 timeslots in TSCH). Since the number of channels and the slotframe length are mutually prime numbers, the physical channel associated with the same cell in the next slotframe will be different. However, such a technique presents two major drawbacks:

Delay: Since the transmission is postponed for the next slotframe, blacklisting would consequently increase the end-to-end delay. The jitter is also increased due to the fact that the delay increases if the channel offset leads to a blacklisted channel;

Bandwidth: Blacklisting a channel prevents to use the cell in all the corresponding slotframes. Thus, if X% of the channels are blacklisted, the radio link can only use (100-X%) of the radio bandwidth.

Let us assume that we adapt directly Equation 4.2, where $nFreq$ would be the number of non blacklisted channels, and $F()$ would map the values to the physical channels. Let us now consider two mutually interfering wireless links that use the same timeslot but a different channel offset. These links, would never collide, if they do not employ any blacklisting. However, if they use different blacklists, *different* channel offsets may map to the *same* physical channel.

Let's consider the scenario illustrated in Fig. 4.1. The radio link $A \rightarrow C$ has no blacklisted channel, while $B \rightarrow D$ blacklisted the channel 15, and they are assigned to them the channel offsets 2 and 1 respectively. Since the modulo changes, we may create several collisions in consecutive slotframes even when blacklisting only one channel.

Therefore, we propose to adapt the frequency hopping method, making the collisions non repetitive. We aim to minimize the number of collisions among interfering links that use a different channel offset if their blacklist differs slightly. To do so, we apply first the Equation 4.2 to compute the radio channel to use. Then, the algorithms makes the distinction between the following cases:

C1: **Good channel**: If the physical channel is not blacklisted, let's use it;

C2: **Blacklisted channel**: If the physical channel is blacklisted, let's select pseudo-randomly a good channel. The pseudo-random function must use a common knowledge between the receiver and the transmitter to avoid deafness. We propose to select the channel accordingly:

$$frequency = F\Big(\Big(ASN + channelOffset + k\Big)\ \%\ nFreq\Big) \tag{4.3}$$

with k the minimum integer value such that 'frequency' corresponds to a *good* channel. Since ASN, $channelOffset$, $nFreq$ and the blacklist are common to the receiver and the transmitter, they will lead to a consistent decision.

Since we keep the same modulo operator, two cells with different channel offsets will never collide if the channel hopping sequence leads to a good channel. A collision may occur probabilistically if at least one of the radio links leads to a blacklisted channel during the corresponding slotframe. The probability of collision is then uniformly distributed among all the channels. In other words, such repartition may be considered like external interference and over-provisioned cells should be already reserved for retransmissions to cope with this situation.

4.2.3 Modifying the Channel Hopping Sequence to Passively Monitor the Quality of Bad Channels

We continuously estimate the PDR performance for all channels, including the blacklisted ones. Indeed, since the radio conditions may change during the deployment (cf. chapter 3), we should recover a radio channel from a blacklist to whitelist, when its PDR performance exceeds the threshold value (Algorithm 1, line 24). However, dedicating resource (control packets) to probe bad channels is not recommended since it would be costly in terms of energy consumption and additional unnecessary traffic. Note that in such a case, the probe has to be done for each blacklisted channel for each radio link.

In this study, we rather propose to monitor the link quality using a passive method, exploiting directly the reliability statistics of data packets. However, a bad channel should be probed less frequently than a good channel since it has a negative impact on both the reliability and the energy consumption.

Therefore, we modify the previous second rule (C2) when computing the channel hopping sequence. More precisely, when Equation 4.2 returns a blacklisted channel:

C2.1: With the probability p, let's transmit the packet through this *bad* channel to keep on re-estimating the link quality for *all* channels;

C2.2: Otherwise, the transmitter and receiver select pseudo-randomly a good channel, applying the original C2 rule (cf. section 4.2.2).

A small p value means that the blacklisted channels will be probed infrequently. Re-estimating the quality consumes less resource, but requires a longer time to detect link quality change.

4.2.4 How to agree on a consistent blacklist in the transmitter and the receiver?

Recall, as previously detailed, each node calculates the number of ACKs received from a neighbor to compute the PDR. The transmitter then identifies the blacklisted channels according to their PDR by applying Algorithm 1. Hereafter, we should ensure that the transmitter and the receiver have the same blacklists, else they would use a different pseudo-random frequency hopping sequence, leading to a "deafness".

We focus here on providing a full blacklisting-enabled 6TiSCH stack. Thus, to this aim, the transmitter sends to the receiver its blacklist using a reliable method since the receiver is not aware of the actual statistics computed by the transmitter, and cannot construct the same blacklist. We here propose to exploit 6P to exchange the blacklists for each radio link (*e.g.*, A, B). More precisely, the transmitter A sends its blacklist in a 6P control packet. Note that 6P packets are transmitted through the shared cells and are prone to collisions: B needs to send an acknowledgement.

The IEEE 802.15.4 IEs are a convenient option to include the blacklist in the 6P packets. In our implementation, a node maintains for each of its active neighbors (i.e., to which it transmits packets) two blacklists:

1. `tx-tmp`: the last blacklist computed according to Algorithm 1, not yet acknowledged by the receiver;
2. `tx`: the last blacklist which was transmitted **and** acknowledged by the receiver.

Thus, we guarantee to use consistent blacklists for both sides. The list `tx-tmp` is used to construct a 6P IE. When the corresponding ACK is received, `tx-tmp` is copied in `tx` and then destroyed. Each node maintains different blacklists with each of its children. We thus achieve to define an adaptive, localized and per-link (per child) blacklisting algorithm.

We assume that the loss of ACKs when the packet is received can be neglected. If the ACK is lost, the blacklists may become inconsistent, and the transmitter at some time will try to update its blacklist.

4.3 Experimental Performance Evaluation

In this section, we present a thorough experimental campaign over the FIT IoT-LAB platform* that is part of the FIT†, an open large-scale and multiuser testing infrastructure for IoT-related systems and applications. Note that FIT IoT-LAB is a shared platform with potential concurrent experiments.

4.3.1 FIT IoT-LAB Platform

We conducted our study over the FIT IoT-Lab testbed, which belongs to the half real-world testbed category since several Wi-Fi APs are co-located. Thus, under such a realistic indoor environment, the nodes are subjected to external interference originated from Wi-Fi-based devices.

4.3.2 Experimental Setup and Parameters

In our experimental campaign, we employed M3 nodes, based on a STMicroelectronics 32-bit ARM Cortex-M3 micro-controller (ST2M32F103REY) that embeds an AT86RF231 radio chip, providing an IEEE 802.15.4 compliant PHY layer.

We focused on a 1-hop scenario with 10 M3 nodes to focus on the performance of a given radio link. We performed 120 experiments, while each experiment lasted for 120 *min*. The

*https://www.iot-lab.info/
†https://fit-equipex.fr/

Table 4.1: Experimental setup.

Topology	**Parameter**	**Value**
	Testbed site	Strasbourg site
	# of nodes	10
	# of Experiments	120
	Link Distance	$[2.0 - 14.3]$ *meters*
Experiment	**Parameter**	**Value**
	Duration	120 *min*
	Payload size	48 bytes
Protocol	**Parameter**	**Value**
CoAP	CBR (*Unicast*)	1 *pkts*/3 *sec*
RPL	DAO period	50 *s*
	DIO period	8.5 *s*
TSCH	Slotframe length	101
	NShared cells	5
	Timeslot duration	15 *ms*
	Maximum retries	3
Queues	Timeout	8 *s*
	Queue size	10 packets
	incl. data packets	Maximum 6 packets
Hardware	**Parameter**	**Value**
	Antenna model	Omnidirectional
	Radio propagation	2.4 *GHz*
	802.15.4 Channels	11 *to* 26
	Modulation model	AT86RF231 O-QPSK
	Transmission power	0 *dBm*

transmitter (leaf) node implements a CBR application model, by transmitting 1 data packet every 3 *seconds*, at 0 *dBm* transmission power, resulting in more than 2000 *pkts* transmissions in total per experiment. We chose a 48 *bytes* data size, which corresponds to the general information used by monitoring applications (e.g., node ID, packet sequence, sensed value). The details of the setup are exposed in Table 4.1. We systematically plotted the 95% confidence intervals (each radio link denoting a dataset).

To conduct our experiments, we employed OpenWSN[‡], an open-source implementation of a full protocol stack based on loT standards (IEEE 802.15.4-TSCH, IPv6, 6TiSCH, RPL, CoAP). In particular, we used the modified implementation of OpenWSN [§] for distributed scheduling with traffic isolation [TP16], to reserve a set of cells *per* flow.

4.3.3 Blacklisting Methods to Compare

We compared the following blacklisting methods:

- **Default:** TSCH network operates in standard mode and uses only channel hopping to defeat external interference;
- **Global Blacklisting**: We blacklist statically the three channels which are the most impacted by the interfering Wi-Fi networks — channel 12, 13 and 14;

[‡] `http://www.openwsn.com`

[§] branch *"track"* of `https://github.com/ftheoleyre/openwsn-fw/` and `https://github.com/ftheoleyre/openwsn-sw/`

- **Local-Fixed**: We blacklist all channels that exhibit a PDR lower than a fixed threshold value. This blacklist is then used locally to modify the pseudo-random channel hopping sequence. Note that if all 16 radio channels present a performance lower that the pre-defined threshold, we select the channel with the best PDR value.
- **Local-Adaptive: LABeL**: The blacklist is computed based on Algorithm 1. It is established as a per link basis, selecting the channels which perform significantly worse than the best ones. Thus, a channel is blacklisted not anymore only because it performs poorly, but more importantly if it exhibits a PDR significantly lower than the best channels for the *same* link. In other words, we avoid penalizing the links with a mediocre quality.

4.3.4 Studied Metrics

We measured the following metrics to evaluate the network performance:

- **Packet Delivery Ratio (PDR):** The ratio of the number of packets correctly acknowledged by the receiver and the number of packets transmitted by the transmitter. The PDR is measured at the MAC layer: one packet with one retransmission results a PDR of 50%;
- **Delay:** The average time between the generation of a packet and the reception of the corresponding acknowledgement. This average delay is computed only for the packets successfully delivered to the receiver;
- **Jitter:** The average difference for a given flow between its actual end to end delay and its average value;
- **Blacklist size:** The number of channels present in the blacklist;
- **ETX:** The average number of transmissions and retransmissions for each data packet. This metric is relative to the energy consumption: more cells and transmissions are required to deliver each data packet.

4.4 Performance Evaluation

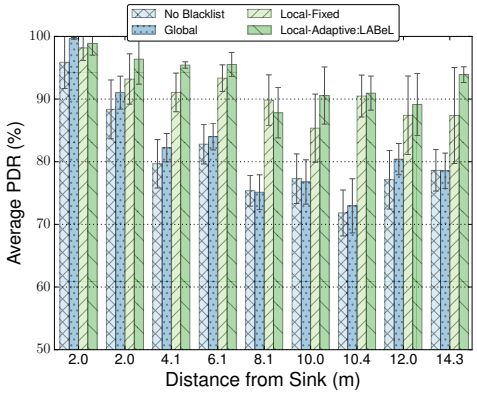
4.4.1 Reliability

We first focus on the reliability performance and measure the PDR provided by a given link (Fig. 4.2a). To investigate the impact of the signal strength by grouping together the links with approximatively the same geographical length (in our testbed, the signal strength and the geographical length are quite strongly correlated variables).

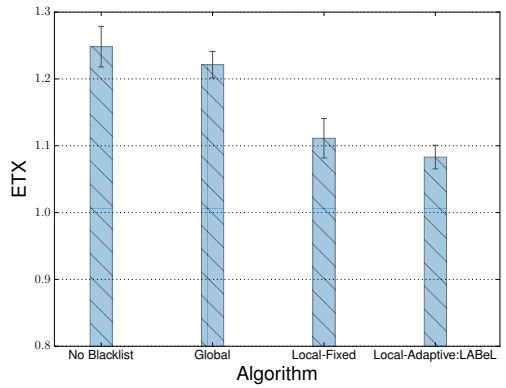
For short (and strong) links, PDR is very high (≈100%) whatever the employed blacklisting technique (Fig. 4.2a). However, blacklisting technique improves slightly the PDR, even for strong links.

Weaker links tend to be more sensitive to external interference since their SINR margin is smaller. The *bad* channels, with a large level of external interference, impact negatively the reliability. All the blacklisting techniques improve in some extent the PDR. The global blacklisting provides the lowest improvement: some channels perform *badly* only for *some* radio links while they are blacklisted globally. Local blacklisting with a fixed threshold value is also suboptimal: a weak radio link tends to exhibit a low average PDR for all its channels. Thus, a medium PDR does not mean that a channel should be blacklisted. LABeL, computing dynamically the threshold value for the PDR, according to the best channels, is more effective to blacklist only the less efficient channels.

Next, we measured the ETX in Fig. 4.2b. ETX is related to the energy efficiency since a node has less packets to transmit on average to deliver correctly a data packet. As can be observed, LABeL, the link-based adaptive scheme, provides an ETX below 1.1, making on average links more robust (14% less transmissions compared to without blacklisting).



(a) Packet Delivery Ratio.



(b) Average number of transmissions before receiving an ACK.

Figure 4.2: Per link reliability achieved with the different blacklisting methods.

4.4.2 Blacklist size

We measured the average number of channels present in the blacklist (Fig. 4.3). The global blacklist is not represented since we fixed statistically its composition, including the three channels most impacted by Wi-Fi.

Our results demonstrate that the stronger the links, the fewer the blacklisted channels. Besides, we can verify that using a fixed threshold is suboptimal and aggressive: it tends to blacklist also channels which are close to the best ones, but below the fixed threshold. It is straightforward that using weaker links means also blacklisting more channels, whatever the blacklisting method is.

4.4.3 Delay

We finally consider the delay (in number of timeslots) between the packet's generation and the reception of the acknowledgement from the receiver (Fig. 4.4a). The global blacklisting technique does not succeed to blacklist the worst channels: some keep on providing a low reliability and the packet has to be retransmitted. Indeed, it increases the average delay, while the standard deviation is much larger: some radio links are very negatively impacted by the non-blacklisted bad channels. On the contrary, local blacklisting allows to block the usage of the worst channels and to reduce the amount of retransmissions, thus, it reduces the delay.

In the IIoT, a deterministic and predictable performance is required. Therefore, we focus specifically in Fig. 4.4b on jitter. While the non-blacklisting technique provides the highest jitter

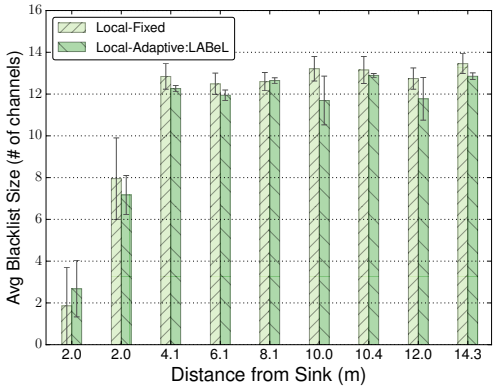
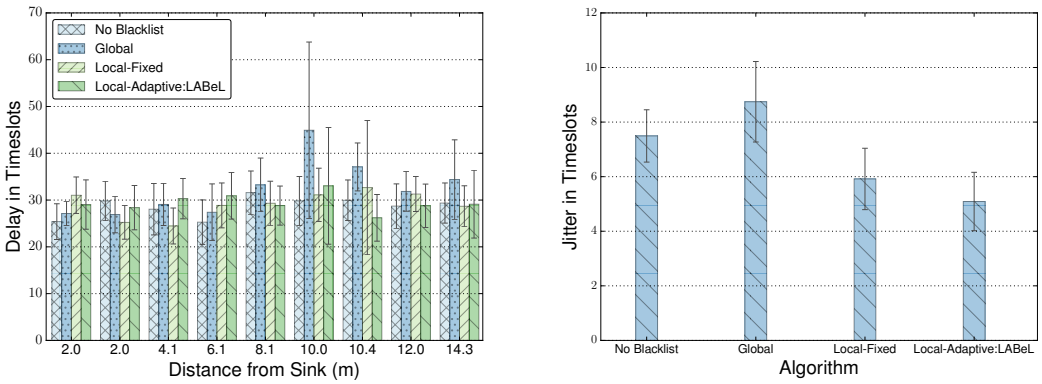


Figure 4.3: Average number of channels present in the blacklist.



(a) Delay (in timeslots).

(b) Jitter (in timeslots).

Figure 4.4: Time required for a given link to receive an ACK for a transmitted packet.

due to retransmissions, LABeL successfully identifies and exploits only the best channels and provides decreased jitter values.

4.5 Conclusions and Perspectives

In this Chapter, we proposed LABeL, a localized and link-based adaptive blacklisting technique. By employing a WMEWMA estimator paired with a dynamic PDR threshold, we are able to identify the bad radio channels, that a pair of nodes should avoid to use to improve the reliability. LABeL modifies the pseudo-random channel hopping sequence to use in priority the *good* ones. More precisely, we repeatedly apply the pseudo-random function until it provides a *good* radio channel. This, we make the collisions pseudo-random, and not repetitive among consecutive slotframes. In other words, they can be handled like usual external interference.

However, we still have to maintain the blacklist up-to-date since the external interference varies over time. Thus, LABeL keeps on integrating in the pseudo-random channel hopping sequence the bad radio channels. These *probes* are sufficiently unfrequent to limit the impact on the reliability. However, they maintain accurate estimations on the link quality over **all** the physical channels.

Our thorough experimental evaluation based on OpenWSN (implementation of 6TiSCH stack) and FIT IoT-LAB platform, exhibits that LABeL, an adaptive and link-based blacklisting tech-

nique, improves the reliability performance by 20% as well as it reduces the unnecessary traffic in the network, while improving the jitter performance.

In the future, we plan to extend our method to identify bad radio channels. In particular, blacklisting the channels providing a bad PDR may lead to a bias if only a few packets are forwarded through a given link. Thus, it would be interesting to study methods that do not rely directly on the PDR. Furthermore, we plan to develop a technique that will be able to determine for each radio link the optimal sampling rate of bad channels according to its characteristics. Thus, in case of stable in time *bad* radio channels, the sampling rate should be reduced, and in the opposite case, should be increased to achieve near to the optimal performance.

In the case of dense networks where nodes generate/transmit data at high rates, a significant increase in the rate of internal collisions is expected, thereby limiting the benefits of applying our proposed blacklisting technique. Therefore, the technique of scattering of internal collisions provided by LABeL is inadequate. For this reason, in the next chapter, we study the effect of internal collisions analytically, and we propose centralized blacklisting techniques where they eliminate collisions.

Chapter 5

Centralized Whitelisting Techniques

The enhancement of the channel hopping with a blacklisting technique improves significantly the reliability of a wireless network as highlighted in the previous chapter. However, the application of any local (per-link) blacklisting technique on multi-hop wireless sensor networks may create *internal collisions*. More precisely, *internal collisions* are caused by the interfering links scheduled at the same timeslot: because they use different blacklists, collisions may arise even if they receive different channel offsets.

The application of a distributed blacklisting technique to eliminate internal collisions requires that the nodes are aware of the interfering radio links scheduled in the same timeslot as well as the channel hopping sequence to use. However, the exchange of information between 1-hop neighbors is not sufficient to acquire the above information. Let's consider the network depicted in Fig. 5.1; radio links BS, CD, EF are scheduled in the same timeslot where BS, CD interfere with each other, and CD, EF interfere with each other. As a consequence, message exchange between radio links' nodes with hop distance greater than 1-hop is required, resulting in increased network traffic. Therefore, it seems more efficient to propose centralized approaches where a central entity is aware of the whitelist of each radio link, the central schedule, and the set of the interfering radio links.

Therefore, in this chapter, we propose a centralized approach to use heterogeneous blacklists while maintaining a fully collision-free schedule. We will use in this chapter the term **whitelisting** instead of the term blacklisting, to highlight the fact we use a "safe" radio channel allocation process (i.e., collision-free).

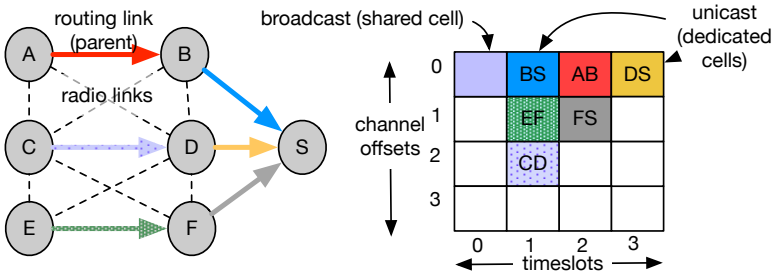


Figure 5.1: TSCH schedule for a 7 nodes topology.

Contribution

This chapter presents the following contributions:

1. We detail how probabilistic collisions may occur when exploiting link-dependent whitelists.
2. We propose three different schemes to avoid internal collisions:
   (a) We allocate different timeslots to the radio links that can cause an internal collision.
   (b) We force all links scheduled in the same timeslot to use the same whitelist. These whitelists are suboptimal, but no collision is introduced;
   (c) We apply our algorithm to reorder the whitelists to forbid any possible collision.
3. We evaluate the performance of these whitelisting techniques using an experimental dataset obtained in an indoor environment i.e., FIT IoT-LAB platform.
4. Finally, we provide additional open challenges in this research area.

5.1 Problem Formulation

To improve the reliability, only the best radio channels (whitelist) should be used. Improving the reliability means also less retransmissions and, thus, lower energy consumption; an idle timeslot consumes much less energy than both the reception and transmission [Vil+14]. Thus, we need to assess the quality provided by a radio channel with a certain link metric (e.g., RSSI, PDR), measured independently for each radio channel. Because the radio channel to use is derived pseudo-randomly from the channel offset, we need to wait for a sufficiently long time to obtain accurate measurements.

Whitelisting consists in identifying the best radio channels which should be used when transmitting the packets to optimize reliability. Unfortunately, whitelists are often link dependent (cf. chapter 3) ; the signal strength, and the location of the source of interference impact both the size of the whitelist as well as the set of the radio channels to include.

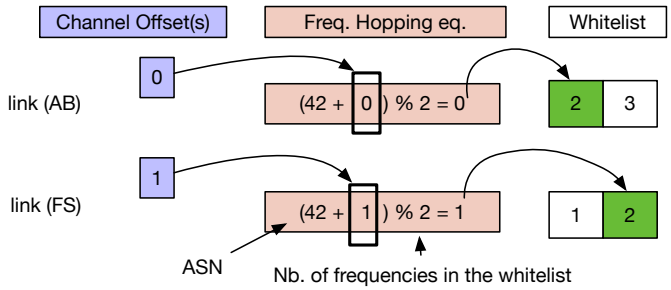


Figure 5.2: Radio channel computation with whitelists (ASN=42).

Table 5.1: Notation

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

5.1.1 Collisions with whitelists

A collision can occur only if two transmitters use the same radio channel to send their packet during the same timeslot. Thus, the scheduling algorithm does not create collisions when allocating different timeslots to the different links. Allocating two different channel offsets for two transmitters during the same timeslot is also safe if we do not use whitelists. Both transmitters will derive the radio channel to use with Equation 5.1, leading thus to different values (radio channels).

$$frequency = F\Big(\big(ASN + channelOffset\big)\ \%\ nFreq\Big) \tag{5.1}$$

However, collisions may occur when considering whitelist, when two transmitters use the same timeslot. Indeed, the Equation 5.1 may use a different modulo operator. The pseudo-random sequences may overlap (i.e., same radio channel at the same position for the two sequences), and create collisions.

Let us consider the example depicted in Fig. 5.1, with a schedule for a topology of 7 nodes. The links (A, B) and (F, S) have been scheduled in the same timeslot, but over a different channel offset (0 and 1, respectively).

Fig. 5.2 illustrates the frequency mapping process for a specific timeslot (ASN=42), using Equation 5.1. Let us assume that (A, B) and (F, S) have the whitelists {2, 3} and {1, 2}, respectively. The link (A, B) has to use its first whitelisted radio channel (i.e., 2) while the link (F, S) has to use the second one (also radio channel 2). In conclusion, a collision will be created in the cell with the ASN 42.

5.1.2 Formalization

Let us calculate the sequence of ASNs where collisions occur (see the example illustrated in Fig. 5.2). A collision is generated when two transmitters use different channel offsets during the same timeslot, but which leads to the same radio channel. Indeed, the radio channel is derived from the channel offset, according to Equation 5.1. Thus, the equation can lead to the same value, even with two different channel offsets, when using different whitelists (different modulo operator, or different positions of each radio channel in the whitelist, cf. section 5.1.1.)

Let us assume that the whitelists of the two radio links (A, B) and (F, S) are $\mathcal{W}_{AB}$ and $\mathcal{W}_{FS}$, respectively. A collision can occur only if both whitelists have at least one common radio channel. We denote by pos_{AB} (resp. pos_{FS}) the position of the common radio channel in $\mathcal{W}_{AB}$ (resp. $\mathcal{W}_{FS}$). A collision occurs if Equation 5.1 results in the same radio channel for the links (A, B) and (F, S):

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

$$\Leftrightarrow \exists (x, y) \in \mathbb{Z}, ASN + choff_{AB} = |\mathcal{W}_{AB}| \cdot x + pos_{AB}$$

$$\wedge \quad ASN + choff_{FS} = |\mathcal{W}_{FS}| \cdot y + pos_{FS} \tag{5.2}$$

$$\Leftrightarrow \exists (x, y) \in \mathbb{Z}, \quad |\mathcal{W}_{AB}| \cdot x - |\mathcal{W}_{FS}| \cdot y = c_1 \tag{5.3}$$

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

Equation 5.3 is a linear Diophantine equation [Joh60] of the general form $ax + by = c \quad x, y \in \mathbb{Z}$. It has an infinite number of solutions if the greatest common divisor (GCD) of a and b divides c ($GCD(a, b) \mid c$). Moreover, if (x_o, y_o) is a solution, then the other solutions have the following form:

$$x = x_o + \frac{b}{d}n, \quad y = y_o - \frac{a}{d}n \quad n \in \mathbb{Z} \tag{5.4}$$

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

Consequently, Equation 5.3 has an infinite number of solutions if $d = GCD(|\mathcal{W}_{AB}|, -|\mathcal{W}_{FS}|)$ divides c_1. Let's assume that (x_1, y_1) are possible solution of Equation 5.3. All the integer solutions are then:

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

Let lcm be the least common multiple of $|\mathcal{W}_{AB}|$ and $|\mathcal{W}_{FS}|$. We have also $|\mathcal{W}_{AB}| \cdot |\mathcal{W}_{FS}| = LCM(|\mathcal{W}_{AB}|, |\mathcal{W}_{FS}|) \cdot GCD(|\mathcal{W}_{AB}|, |\mathcal{W}_{FS}|)$. Thus, according to Equation 5.2:

$$ASN = |\mathcal{W}_{AB}| \left(x_1 - \frac{|\mathcal{W}_{FS}|}{d}n \right) + pos_{AB} - choff_{AB}$$

$$\Rightarrow \quad ASN = -lcm \cdot n + |\mathcal{W}_{AB}| \cdot x_1 + pos_{AB} - choff_{AB} \quad n \in \mathbb{Z} \tag{5.5}$$

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

$$ASN = -lcm \cdot x + |\mathcal{W}_{AB}| \cdot x_1 + pos_{AB} - choff_{AB} \quad \wedge \quad ASN = \mathcal{S} \cdot y + ts \quad x, y \in \mathbb{Z} \tag{5.6}$$

$$\Rightarrow lcm \cdot x + \mathcal{S} \cdot y = c_2$$

$$where \quad c_2 = |\mathcal{W}_{AB}| \cdot x_1 + pos_{AB} - choff_{AB} - ts \quad x, y \in \mathbb{Z} \tag{5.7}$$

Equation 5.7 is still a linear Diophantine equation so it has solutions iif $GCD(lcm, \mathcal{S}) \mid c_2$. So if Equation 5.7 has a solution (x_2, y_2) and $d = GCD(lcm, S)$, according to Equation 5.4, we have a collision for every timeslot with the following ASN:

$$\mathcal{S}\left(y_2 - \frac{lcm}{d}n\right) + ts = \mathcal{S} \cdot y_2 + ts - \frac{lcm \cdot \mathcal{S}}{d}n \quad n \in \mathbb{Z}$$

Since $ASN \geqslant 0$ we can rewrite Equation 5.1.2 as follows:

$$ASN = \mathcal{S} \cdot y_2 + ts + \frac{lcm \cdot \mathcal{S}}{d}n \quad n \in \mathbb{N}^* \tag{5.8}$$

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

In this chapter, we propose mechanisms to exploit whitelists without creating these collisions. More precisely, we re-arrange the whitelists when we detect collisions to avoid any inconsistent configurations.

5.2 Avoiding collisions when using whitelists

Existing per-link whitelists, independent of the schedule, often generate collisions pseudo-randomly. In particular, LABeL (cf. chapter 4) adopts a pseudo-random approach to use whitelists. However, it may generate collisions even among links which use different channel offsets. Here, we adopt rather a deterministic approach, where no collision is generated in the network. We rely on a centralized scheduling algorithm to assign the cells (timeslot and channel offset), and then to resolve collisions due to overlapping whitelists.

5.2.1 Whitelist-aware assignment

Since this approach has not been studied so far, we propose here the first whitelist-aware centralized scheduling algorithm. To the classical constraints (half-duplexity, interference, etc.) [Pal+13], we insert a set of constraints to deal with whitelists:

Same whitelist: If two radio links have the same whitelist, they cannot create collisions if they are allocated in the same timeslot with different channel offsets. The mapping function will never give the same result;

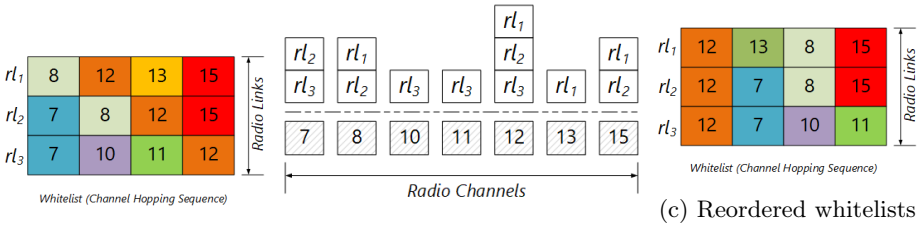
Disjoint whitelist: If two radio links use a disjoint whitelist, no collision can take place by definition. In all other cases, the scheduling algorithm considers a whitelist conflict and allocates different timeslots to the two links.

The centralized controller needs to know the whitelist of all the links, which may represent a large overhead. Adaptive whitelists mean that the schedule has to be probably changed accordingly, since new whitelisting constraints may arise.

Conflicting links are scheduled in different timeslots, and the schedule length tends to increase when whitelist are very different among the nodes. We need to explore how the centralized controller can also tune the whitelists to reduce the number of constraints. For instance, a channel may be removed by the centralized controller in a whitelist if it removes the conflict between a pair of links.

5.2.2 Common Whitelist per Timeslot

Since radio characteristics are variable across the wireless network, imposing the same whitelist for all radio links is suboptimal. This constraint causes an overall downgrading of network performance. Inversely, a per link whitelist may create collisions, as exposed previously. Thus,



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

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

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

Figure 5.3: Reordering process of the whitelists for a group of links scheduled during the same timeslot.

we propose here to group the radio links so that all the links in a group share the same whitelist, but different groups may have different whitelists.

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

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

Let us consider the TSCH schedule depicted in Fig. 5.1. Typically, the links *CD*, *EF* and *BS* must use the same whitelist to avoid collisions. Moreover, the whitelist has to contain at least 3 radio channels to support multiplexed transmissions, through 3 different channel offsets.

We construct a common whitelist as following:

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

5.2.3 Whitelist re-Ordering

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

We propose to enhance the previous solution by rearranging the whitelists to remove the collisions. Let us consider the example depicted in Fig. 5.2. If the whitelists of (A, B) and (F, S) are re-arranged into {3, 2} and {1, 2} respectively, a collision cannot anymore happen. Indeed, a collision occurs only if the Equation 5.1 results in the same integer value. However, this will never occur since the two links have different channel offsets.

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

This problem is closely related to the *University Course Timetabling Problem (UCTP)* [Bab+15], where a set of lectures have to be scheduled for a set of students, during the same timeslots. Similarly, the links correspond to the students, the channel offsets to the timeslots and the radio

channels to the lectures. Furthermore, a lecture should be given in a single timeslot, for all students who have to attend it. Similarly, a radio channel must be located at the same place in the different frequency hopping sequences.

The above problem is NP-complete and the research community has proposed many algorithms such as genetic, hybrid, tabu [Nan+12] approaches. In our case, the frequency hopping sequence length and the number of links during the same timeslot are reasonably small. Thus, a greedy approach seems acceptable to produce an efficient common whitelist.

Basically, the input of our algorithm is a matrix where each row corresponds to the whitelist of a link scheduled in a given timeslot (Fig. 5.3). The output of the algorithm is the same matrix, with the re-ordered whitelists to avoid collisions.

For instance, let us assume three links have the whitelists depicted in Fig. 5.3a. These whitelists have to be re-arranged, else, the radio channel 12 which is used by different links may create collisions. Thus, we first construct the list of links associated with each radio channel (Fig. 5.3b). Our greedy algorithm then picks the radio channels: it first picks the radio channel 12, placed at the beginning of all whitelists. Then, the radio channel 7 is selected, used only by two links. Since the first link does not have the radio channel 7 in its whitelist, it has to search for a radio channel only whitelisted by this link (i.e., radio channel 13). Finally, the algorithm finds a re-ordered solution, without collision, as illustrated in Fig. 5.3c.

Our re-ordering algorithm works in the following way (Algorithm 2):

1. We list the radio links which contain a given radio channel in their whitelist (fig. 5.3b) (lines 1-5).
2. We initialize the output matrix, the reordered whitelists (lines 6-8).
3. For each column (position in the whitelists) of the output matrix (lines 9-20) :
   (a) We look for a radio channel whose corresponding list is the largest subset of the column's list (line 11). It represents the most widely whitelisted radio channel;
   (b) We report this radio channel in the final re-ordered whitelists, and delete the common elements of the two lists (lines 12-18).
   (c) We repeat the above steps, i.e., **a)**, and **b)**, until the column is full, or there is no channel that meets the criterion of step **a)** (lines 10-19).
4. If some cells are still empty in the output matrix, it means that some radio channels are missing in the whitelists. Thus, we adopt here a best-effort approach, filling with suboptimal radio channels as follows:
   (a) We identify the incomplete whitelists (line 22);
   (b) We replace the empty cells by radio channels, not used in other whitelists or used by other whitelists at the same position (lines 23-28). In this way, collisions are impossible, at the price of a reduced reliability for these radio links.

5.3 Evaluation setup

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

5.3.1 Experimental Dataset

We rely on a dataset obtained from the FIT IoT-LAB* platform. This large-scale testbed mimics well an indoor, complex, environment. Other Wi-Fi or IEEE 802.15.4 networks are also deployed in the building, generating external interference. More precisely, we select a large set of radio

*https://www.iot-lab.info/

Algorithm 2: Whitelists Re-Ordering

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

```
// for each radio channel and each link
for l ∈ Links and ch ∈ [0, nch − 1] do
    if ch ∈ WL[l] then
        ListofLinksPerChannel[ch].add(l)
    end
end
// each column contains all links since WL_reord is empty
for l ∈ Links and ch ∈ [0, nch − 1] do
    Cols[ch].add(l)
end
// selects greedily the radio channels
for k ∈ [0, nch − 1] do
    repeat
        ch ← maxSubset(ListofLinksPerChannel, Cols[k]);
        // constructs the whitelists
        for l ∈ Links do
            // If the link has this radio channel in its whitelist, let's use it
            if ch ∈ WL[l] then
                WL_reord[l][k] ← ch;
                Cols[k] = Cols[k] − ListofLinksPerChannel[ch];
                ListofLinksPerChannel[k] ← ∅;
            end
        end
    until Cols[k] = ∅ or ch = −1;
end
// completes the whitelists by the isolated radio channels
for l ∈ Links and k ∈ [0, nch − 1] do
    // identifies all empty cells of WL_reord
    if WL_reord[l][k] = ∅ then
        for j ∈ [nch, 15] do
            ch ← ChRank[l][j];
            // if the channel has not been used so far or used by another link at the same
               column let's use it
            if ch ∉ WL_reord or ch ∈ WL_reord[∗][k] then
                WL_reord[l][k] ← ch;
            end
        end
    end
end
// we found a valid allocation, returns the reordered whitelists
return WL_reord;
```

links, which forward 1 data packet every 3 seconds. The radio links are scheduled in different timeslots without collision. We store the transmission failure/success for each data packet, for 90 $min^{\dagger}$.

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

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

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

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

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

5.3.2 Scheduling and whitelisting algorithm

In this study, we employed the Traffic Aware Scheduling Algorithm (TASA) [Pal+12] to construct the schedule. At the beginning of each slotframe, each node generates a random number of data packets per slotframe in the range $[1, 5]$. We consider a slotframe size of 293 timeslots with 16 channel offsets, to be able to forward all data packets. Because we focus on the efficiency of the whitelisting mechanism, and not on the scheduling process, we do not provision additional cells for the retransmissions. We repeat each experiment for twenty different random network topologies.

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

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

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

MABO-TSCH: The controller assigns a fixed number of channel offset per link [Gom+17] .

LABeL: Distributed per-link independent whitelists are implemented, where collisions may arise pseudo-randomly among interfering transmitters (chapter 4).

Whitelist-aware: our proposed centralized whitelist aware scheduling approach (section 5.2.1). Since we let each radio link to continuously update its whitelist dynamically, some collisions may arise among different channel offsets.

Common Whitelist per Timeslot: The scheduler assigns the same (common) whitelist for all links scheduled in a given timeslot (section 5.2.2), selecting the k radio channels which exhibit the highest average PDR;

Reordered Whitelists: We apply our re-ordering whitelist algorithm (section 5.2.3);

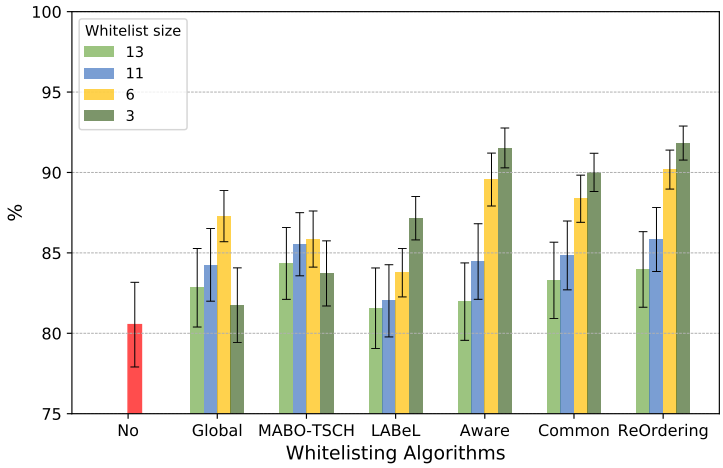


Figure 5.4: Link-level Packet Delivery Ratio.

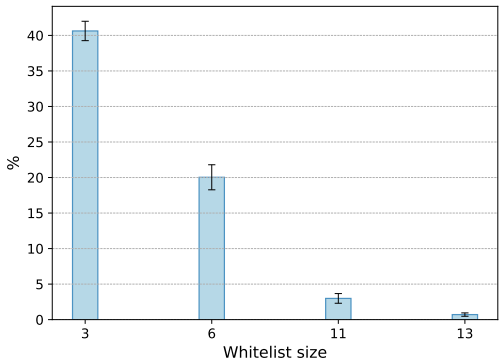


Figure 5.5: Percentage of transmissions which use a non-whitelisted radio channel with MABO-TSCH.

5.3.3 Reliability

We first measure the link-level PDR, i.e., the ratio of the number of data packets delivered by the receiver and the total number of data packets transmitted by the transmitter (Fig. 5.4). We also measure the impact of the whitelist size: a small whitelist means that only the best radio channels are selected, to reduce the number of retransmissions. TSCH without whitelisting achieves the worst reliability: many packets use possibly radio channels with a poor PDR, which negatively impacts the global reliability. Using whatever whitelisting algorithm improves the reliability. Moreover, smaller whitelists mean that only the best radio channels are used, reducing the number of retransmissions. This reliability improvement comes with a decrease of the network capacity: the load has to be spread across a smaller number of radio channels. MABO-TSCH seems less scalable: it does not handle small whitelists: an insufficient number of channel offset is assigned, and the nodes have to use also bad radio channels (Fig. 5.5). A global blacklist improves slightly the reliability by removing the worst radio channels. However, some of the whitelisted channels keep on providing a lower PDR for *some* links.

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

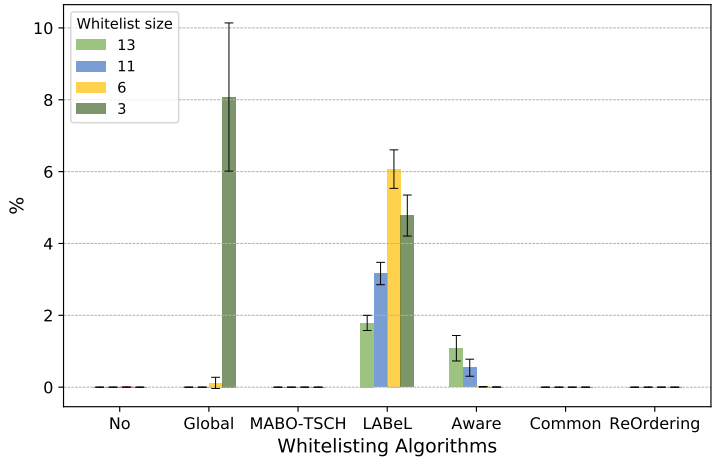


Figure 5.6: Percentage of Collisions (same radio channel with different channel offsets).

We also measured for MABO-TSCH the percentage of transmissions where the assigned channels doesn't give a whitelisted channel (Fig. 5.5). A small whitelist increases the proportion of transmissions over non whitelisted channels ($\approx$ 40% for 3 whitelisted radio channels). This inevitably impacts negatively the reliability, the non whitelisted channels exhibiting the worst PDR. On the other hand, large whitelists abolish the use of non whitelisted channels ($\approx$ 1% for 13 whitelisted radio channels), but integrate also channels with a lower reliability.

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

The whitelist-aware strategy, where the schedule is centrally adapted and the whitelists tuned locally, achieves reliability similar to reordering strategy, especially when the whitelist size is small.

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

5.3.4 Identification of Packet Drop reasons

We also identified the main reasons for packet losses:

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

LABeL adopts a non-deterministic approach, leading possibly to collisions (Fig. 5.6). Indeed, this strategy selects pseudo-randomly a good radio channel if the equation leads to a bad one.

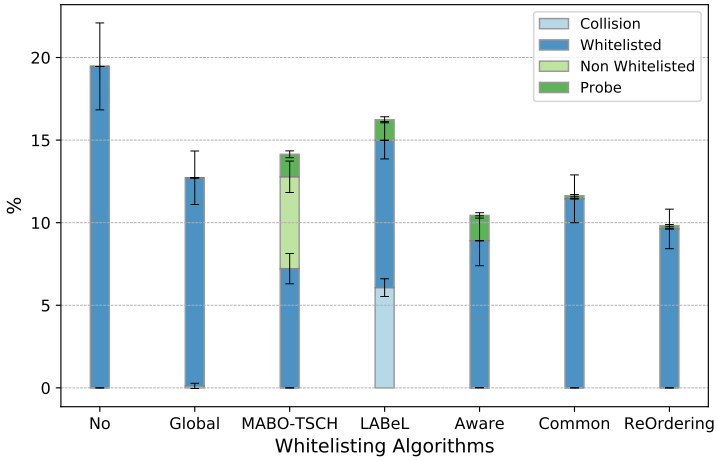


Figure 5.7: Classification of the packet drops, in ratio of the total number of data packet generated in the network (whitelist size of 6 radio channels).

If this radio channel is used by an interfering link, a collision is created. Some packets are also dropped because of probes through the bad radio channels. In the same way, using a small size of global whitelist has a negative impact on the network capacity (Fig. 5.6). Not enough cells are available: the scheduler cannot allocate all transmissions to different cells, leading to parallel transmissions, and thus, collisions.

With MABO-TSCH, a small number of packets (1.5%) are also dropped because they act as probes (Fig. 5.7). However, half of the packet drops are due to the bad radio channels. The number of channel offsets assigned to MABO-TSCH becomes often insufficient when the whitelist is too small. In that cases, the packets have to be transmitted through the bad radio channels, with a negative impact on the reliability.

The whitelist-aware algorithm produces only a small number of collisions since the allocation is done globally, but the whitelist is updated dynamically if required. Shorter whitelists are also more stable and create fewer collisions. Furthermore, as we can observe in Fig. 5.7, the adaptation of the local whitelists to the time-varying radio conditions has the side effect of the collisions on probe packets.

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

5.3.5 Efficiency

We aim here to quantify the gain of our whitelisting algorithms compared with a no whitelisting approach (Fig. 5.8). We measured here for each radio link its *gain*, i.e. ratio of the PDR provided without and with whitelisting. Formulated differently, the gain estimates the improvement of PDR when using a given whitelisting approach compared with the default TSCH behavior.

MABO-TSCH improves the reliability for all radio links. Since the gain is always superior than 1, this means that all links have a better PDR with MABO-TSCH. Obviously, whitelisting does not improve the reliability for perfect links: the gain is much larger for radio links with a medium or bad link quality. However, we can also notice that our reordering algorithm improves significantly the PDR compared with both TSCH without whitelisting and MABO-TSCH. More

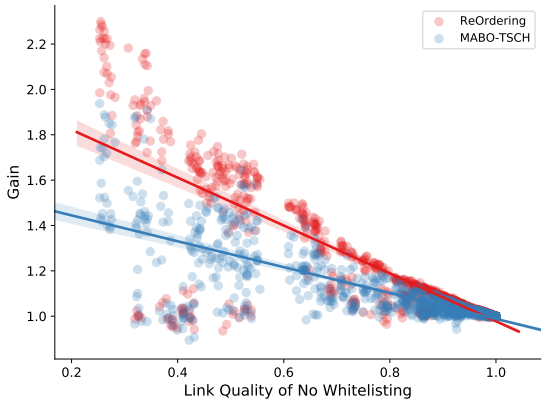


Figure 5.8: Comparing per link improvement of ReOrdering and MABO-TSCH against No Whitelisting (whitelist size of 6 radio channels).

precisely, almost all the links with the reordering approach exhibit a gain larger than any link which uses MABO-TSCH. MABO-TSCH tends to use bad radio channels when the whitelist is too small, a fallback strategy which is not adopted with our reordering whitelists.

5.4 Open Challenges

We have demonstrated the relevance of using blacklisting/whitelisting to reduce the number of transmissions, which impact negatively both the reliability and the energy consumption. However, there are still certain open issues when exploiting whitelists:

Joint-optimization: Considering the routing, scheduling, and whitelisting all together may help to improve performance. For instance, the scheduling algorithm may decide to change the blacklist to relax the scheduling constraints.

We have first to investigate theoretically the gap to fill between a disjoint and a conjoint optimization. Then, heuristics have to be proposed if this gap is significant;

Capacity reduction: With shorter whitelists, the nodes reduce the usage of bad radio channels and, thus, the number of retransmissions. However, transmissions have to be multiplexed through a smaller number of radio channels, increasing the probability of collisions in dense networks.

An adaptive whitelist size, which optimizes reliability while respecting a minimum network capacity has still to be proposed.

Co-located networks: Whitelisting has been designed for external interference using a static set of radio channels. If other co-located networks adopt a channel hopping strategy, whitelisting would be inefficient since the load is spread uniformly across all the radio channels [Yaa+17]. We should rather be able to detect interfering networks adopting the same strategy, for instance with a classification technique [Her+13]. Then, an heuristic to share the radio spectrum among the interfering networks shall be proposed.

5.5 Conclusions and Perspectives

We here proposed a whitelisting technique able to construct a collision-free centralized schedule, even using per-link whitelists. Rescheduling the interfering links to different timeslots minimizes the collisions and improves the reliability but increases the schedule length and, as a consequence,

increases the end-to-end delay. A common whitelist for all the links scheduled in the same timeslots allows the network to reduce the usage of bad radio channels.

We enhanced this algorithm to allow several links scheduled in the same timeslot to exploit different blacklists while still being collision free. More precisely, our algorithm reorders the whitelists so that even two links with different whitelists can be scheduled in the same timeslot without introducing collisions. Our simulation results, which used experimental dataset from FIT IoT-LAB platform, demonstrated the relevance of this approach to reduce the packet drops.

As future work, we should design a centralized scheduling algorithm able to adapt the schedule according to the whitelist. Indeed, it corresponds to an optimization problem where we have to maximize the minimum reliability, by both carefully selecting which radio channels to whitelist and how to re-order them.

While this whitelisting technique is efficient to improve the reliability, it adapts only very slowly to time variations. Indeed, changing a blacklist for a given radio link may possibly impact the blacklist of all the radio links scheduled in the same timeslot. Therefore, we propose in the next chapter a centralized scheduling approach tailored for this constraint.

Chapter 6

Hybrid Blacklisting Technique

In the previous chapter, we proposed whitelisting techniques that, on the one hand, eliminate internal collisions but, on the other hand, do not adapt the links' whitelists to the external interference's time variations.

In this chapter, we propose a hybrid approach extending MABO-TSCH. MABO-TSCH is a combination of a centralized algorithm, where some decisions are taken in distributed manner. Indeed, it computes the cells to allocate to each radio link in a centralized manner, and assigns a collection of channel offsets to each link. In this way, a radio link can decide distributively which channels should not be used for the data transmissions. While MABO-TSCH forbids any internal collision, it is inefficient for long blacklists, dense wireless networks, and heavy network traffic, as we will highlight.

Thus, we propose in this chapter an enhanced version of MABO-TSCH, that is able to allocate cells while using local blacklists.

More precisely, we allocate the channel offsets dynamically at each timeslot according to the number of parallel transmissions, while still avoiding internal collisions.

Contribution

This chapter presents the following contributions:

1. we propose to extend MABO-TSCH by proposing the Adaptive MABO (AM-ABO) algorithm, where the channel offsets are assigned dynamically at each timeslot according to the number of parallel transmissions;
2. we evaluate the performance of our solution relying on a real experimental dataset, highlighting the relevance of dynamic and per timeslot channel offset assignment for environments with high external interference.

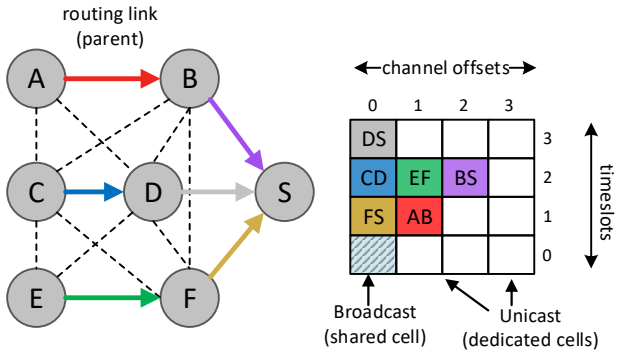


Figure 6.1: TSCH schedule for a 7 nodes topology.

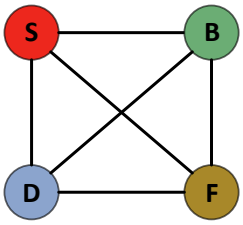


Figure 6.2: MABO-TSCH interference graph

6.1 Problem Statement

MABO-TSCH uses a receiver-based channel offset assignment (cf. subsection 2.3.1.0): a set of channel offsets is assigned to a receiver, and the transmitter has to use one of them for all the data packets to this receiver. MABO-TSCH applies a graph-coloring approach. The vertices are the nodes, the edges are the interfering links, and the colors are the 16 available channel offsets. It is worth mentioning that two nodes (vertices) are joined with an edge if they are neighbors or neighbors of their neighbors. Then, an extended version of the Welsh-Powell algorithm [WP67] is applied to assign multiple non-interfering radio channels to each node.

Let us consider the network topology and the schedule described in Fig. 6.1. Since MABO-TSCH is receiver oriented, we consider only the four different receivers (S, B, D, and F). The interference graph (which pairs of receiver mutually interfere) is represented in Fig. 6.2. In our case, all the receivers are neighbors of the sink S and interfere: this corresponds to a full graph.

Then, each receiver has to select a channel offset different from all its interfering nodes. MABO-TSCH assigns multiple channel offsets to receivers, as illustrated in Fig. 6.3. In particular, all radio links toward S use the channel offsets 0 to 3, and all different receivers use non overlapping channel offset ranges.

Besides, the number of channel offsets and the blacklist size impact directly the performance of a MABO-TSCH schedule. In particular, if the blacklist size exceeds the number of channel offsets, the radio link may not be able to aways use a non blacklisted radio channel. In that case, a blacklisted radio channel needs to be used, impacting negatively the reliability.

Thus, **assigning a fixed number of channel offset is inefficient**, and does not capture the whole network heterogeneity.

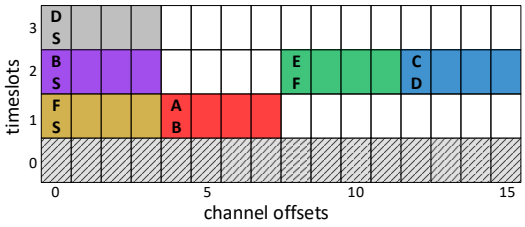


Figure 6.3: MABO-TSCH multiple channel offset assignment

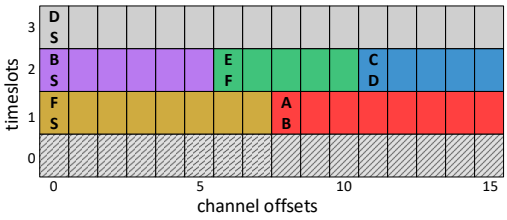


Figure 6.4: AMABO multiple channel offset assignment.

6.2 Proposition: Per timeslot heterogeneous channel offset assignment

We propose to change MABO-TSCH to **not** assign a fixed number of channel offsets per receiver. In this way, we optimize the probability to use a non blacklisted radio channel.

We propose to assign the channel offsets on a timeslot basis (Algorithm 3):

1. for each timeslot, we construct the interference graph corresponding to these links (line 2);
2. we assign fairly the set of channel offsets to each radio link of the clique in the interference graph (lines 6-15);

Indeed, having unused cells in the schedule has no practical interest for the network. While some radio bandwidth would be available, no radio link can exploit it, even if it has a long blacklist because of a high level of external interference.

6.2.1 Illustration

Let us consider the same topology as previously and apply Algorithm 3. To each timeslot corresponds a set of duplex-free but interfering radio links, i.e., a subgraph as illustrated in Fig. 6.5. For instance, during the first timeslot, are scheduled two different links: (AB) and (FS). To avoid wasting channel offsets, we assign half of the channels offsets to (AB), and the other part to the link (FS). On the contrary, the link (DS) is allocated during the third timeslot and receives all the channel offsets.

6.3 Evaluation setup

In this section, we evaluate the performance evaluation of our proposed enhancement on MABO-TSCH technique. To this aim, we emulate a network with 60 nodes by employing an experimental dataset to obtain realistic conditions as well as results.

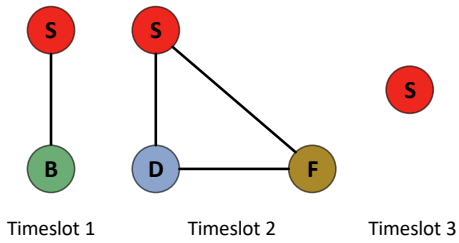


Figure 6.5: AMABO interference graphs per Timeslot

Algorithm 3: Multiple channel offset assignment per Timeslot

Input:
$CG(V', E')$- connectivity network graph
Sch- Common schedule
Output:
Coloring each timeslot's interference graph with multiple channel offsets

```
for all ts_i in SlotFrame do
    G(V,E) - network graph with receiver nodes of assigned radio links to timeslot ts_i as
     vertices and interfering links as edges
    C - list of 16 channel offsets
    Sort vertices v_1, v_2, ...v_n in V in non-increasing degree order
    colored ← true
    while colored=true do
        colored ← false
        for all v_i in V do
            find c_i as the minimal color in C not assigned to any vertex v_j connected to v_i
            if c_i exists then
                colored ← true
                Add c_i to the list of channel offsets of v_i
            end
        end
    end
end
```

6.3.1 Experimental Dataset

We run set of experiments on the FIT IoT-LAB* platform. We present a use-case with two M3 nodes, a transmitter and a receiver, respectively, positioned over various distances, from 0.6 to 17 m. To do so, in each experiment, we selected randomly two M3 nodes (out of 380) in the Grenoble site, where the transmitter transmits periodically 1 data packet every 3 seconds. We conducted more than 330 experiments, while each experiment lasted for 90 *min*.

We logged the following data in a dataset †

- The *distance* between the two devices.
- The *radio channel*, the ASN time and the *result* (success/failure) for each data packet transmission.

We employed the previously presented dataset as an input in a custom made simulator based on Python. We then emulated a wireless network of 60 devices and one root, randomly positioned

*https://www.iot-lab.info/
†The dataset is freely available for the research community at https://github.com/vkotsiou/grenoble-multichannel-dataset.git

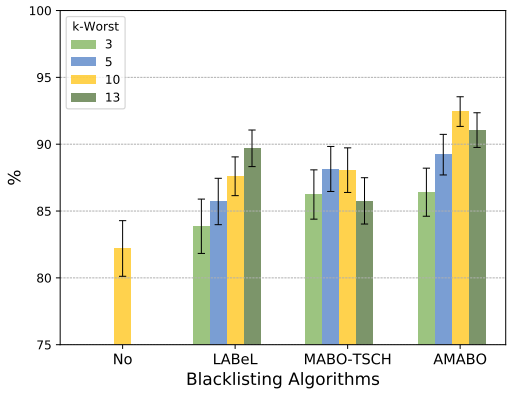


Figure 6.6: Link-level Packet Delivery Ratio.

in an area of 200 X 200 m^2, with a propagation range at 50 m. Thus, the average number of neighbors per node is 9.29, and 3.18 (maximum 6) the average number of hops. Each device choose its parent the neighbor *closest* to the root. Then, to emulate a realistic radio link quality behavior, by mapping each radio link of the simulated wireless network to a radio link of the dataset. Then, for each transmission, we search the most recent transmission in the dataset through the same radio channel and the same link. Its success or failure is then re-injected in the custom-based simulation.

6.3.2 Scheduling and Blacklisting algorithm

The construction of the schedule was carried out by employing the Traffic Aware Scheduling Algorithm (TASA) [Pal+12], where each node generates a random number of data packets to transmit per slotframe in the range $[1, 5]$. In this campaign, we consider a slotframe size of 293 timeslots with 16 channel offsets. Since we aim on the optimization of the channel offset assignment, and not on the scheduling process itself, we do not provision additional cells for the retransmissions. Finally, we execute each simulation for ten different random network topologies to obtain fair confidence intervals.

To objectively evaluate the channel offset assignment strategy, all radio links exclude from their transmissions the k worst channels. We compare the following four approaches:

Default (No Blacklisting): all 16 radio channel are available.

LABeL: a probabilistic approach, each radio link uses a set of radio channels with similar performance compared to $16 - k$ best radio channels, similar to [Kot+17].

MABO-TSCH: multiple channel offsets are assigned to each **radio link**, similar to [Gom+17].

AMABO: multiple channel offsets are assigned to each **timeslot** as described in section 6.2.

6.3.3 Simulation Results

Reliability

To measure the network reliability, we calculated the PDR at the link layer: ratio of the number of data packets delivered by the receiver and the total number of data packets transmitted by the transmitter. The PDR performance it is depicted in Fig. 6.6.

Our performance evaluation campaign shows that IEEE 802.15.4-TSCH without employing any blacklisting algorithm present the worst performance. This is because many transmissions took place over bad radio channels, which negatively affects the reliability. Next, as it can

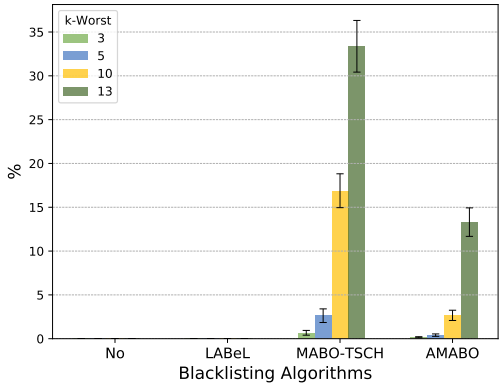


Figure 6.7: Percentage of packets that are transmitted trhough a *bad* radio channel.

be observed with LABeL algorithm, the smaller is the size of the blacklist, only the best radio channels are employed for transmission reducing thus the retransmissions. However, the reliability improvement of LABeL comes at the cost of reducing the network capacity, since eventually there are few (and only "good") radio channels available for communication. Furthermore, MABO-TSCH presents better performance than LABeL in scenarios with small blacklists, however, its performance degrades when the number of blacklist increases. Indeed, since MABO-TSCH assigns inefficiently the channel offset per radio link and, thus, the nodes have to employ "poor" radio channels to transmit, see Fig. 6.7. Finally, our proposed AMABO algorithm improves the MABO-TSCH scheme essentially, as it demonstrates network reliability over 90% by assigning in more intelligent manner the channel offsets. Indeed, it assigns dynamically per timeslot the channel offsets, instead of uniformly and statically per radio link. As a result, it reduced by more than 50% the selection of "poor" radio channels when compared against MABO-TSCH, see Fig. 6.7.

Collisions

Next, we evaluated the percentage of collision that are introduced when two or more interfering radio links are assigned at the same timeslot but have different blacklists. In Fig. 6.8 the obtained results are depicted. Our performance evaluation results show that default IEEE 802.15.4 TSCH do not present any collision. Indeed, with TSCH by design the collisions are impossible due to time-slotted approach. Furthermore, LABeL, the probabilistic approach, is the only algorithm that generates collisions. Longer the size of blacklists (i.e., 10 radio channels) indicates a smaller network capacity as well as higher probability to obtain collisions. Note that the reduction of collisions, as the blacklist size increases, can be explained by the increase of location-based heterogeneity in blacklists, thus, the devices that transmit at the same timeslot, employ "good" radio channels from disjoint whitelists. Finally, there are no collisions with MABO-TSCH and AMABO because they assign to each receiver a set of channel offsets that are different to each other. Thus, by design with MABO-TSCH and AMABO schemes it is impossible to have collisions.

6.4 Conclusions and Perspectives

In this chapter, we extended the MABO-TSCH mechanism to utilize more efficiently the whole range of 16 channel offsets. Indeed, instead of assigning statically a fixed number of channel offsets per radio link, we proposed here to handle a flexible number of channel offsets per timeslot. We managed to optimize the network capacity by sharing uniformly the channel offsets for all the radio links which share the same timeslot. Our experimental dataset demonstrate improvement

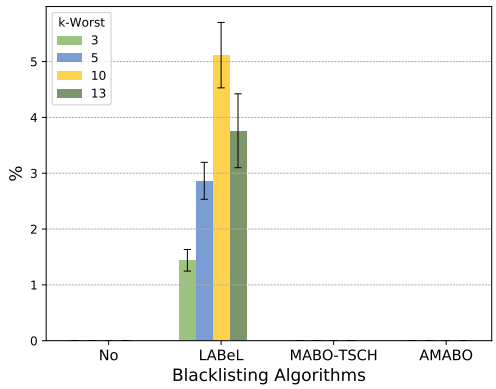


Figure 6.8: Percentage of Collisions due to parallel transmissions.

on PDR as well as highlights a reduction of the number of retransmissions due to “bad" radio channels, particularly for long blacklists.

In the future, we expect to study the scalability, with several collocated networks using the same frequency band. By controlling the volume of traffic (periodic vs. bursty), we expect to study how classification techniques may help to detect the type of interfering technologies (Bluetooth, TSCH, Wi-Fi) to apply the most accurate strategy.

The blacklisting techniques we have proposed so far, as demonstrated by their evaluation through experiments and simulations, significantly improve the reliability of radio links. Improving reliability decreases the number of retransmissions while reducing end-to-end latency. However, the above is inadequate to meet the requirements of IIoT for a low and deterministic end-to-end delay. More specifically, it is required to allocate the schedules' cells of the nodes on the path towards the sink in a daisy chain fashion to minimize the buffering delay and, therefore, the end-to-end delay. Furthermore, unreliable radio links should be considered by allocating (over-provisioning) an appropriate number of cells that can vary over time, due to the temporal characteristics of the radio links, causing rescheduling. The above two requirements, *reliability* and *low end-to-end delay*, in many cases, are contradictory, in the next chapter, we propose a distributed scheduling function for 6TiSCH networks, which satisfies both requirements.

Chapter 7

Low-latency Distributed Scheduling

Networked control automation is expected to rely more and more on wireless transmissions. The three previous chapters were exploring how blacklisting techniques may improve the reliability. However, the wireless infrastructure has to provide both high reliability and a bounded latency. Unfortunately, guaranteeing a bounded end-to-end latency is particularly challenging since transmissions have to be temporally chained. Even worse, potential degradation of the link quality may result in reconstructing the whole TSCH schedule along the path.

In this chapter, we propose the Low-latency Distributed Scheduling Function (LDSF) tailored to provide both high-reliability and a low end-to-end latency. LDSF is fully distributed: each device in the path decides by itself the cells to use. Our solution relies on the organization of the slotframe in smaller parts, called blocks. Each transmitter selects the right set of blocks, depending on its hop distance from the border router, so that retransmission opportunities are automatically scheduled. In this way, we can limit the impact on the end-to-end delay when the packet has to be retransmitted several times by a transmitter. The transmission opportunities are still chained further to limit the buffering delay.

Contribution

This chapter presents the following contributions:

1. We design a novel organization of the slotframes, divided into repetitive short blocks. Chaining the blocks reduces drastically the overall end-to-end delay. Moreover, a transmission opportunity is automatically reserved in consecutive blocks to deal with retransmissions;
2. We define the concepts of primary and ghost cells to save energy. While the primary cell corresponds to the earliest expected reception time, ghost cells are automatically reserved to deal with retransmissions, while limiting the impact on the delay;
3. We provide a mathematical analysis of the average end-to-end latency of the state-of-the-art scheduling algorithms defined to minimize latency, including our LDSF algorithm;
4. Simulations help us to investigate more complex scenarios and highlight the practical interest of our proposed scheduling function.

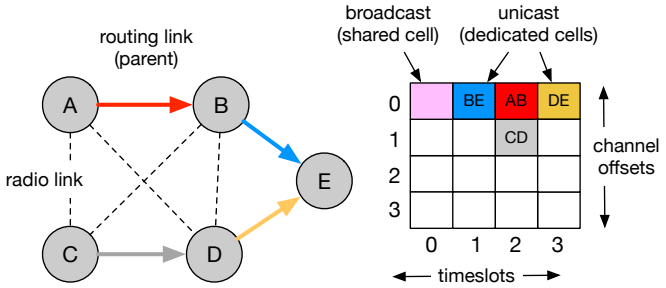


Figure 7.1: TSCH schedule for a 5 nodes topology.

7.1 Problem Formulation and Objectives

Centralized scheduling algorithms require the whole radio and interference topology knowledge, while they are not adaptive to changes. Thus, in this chapter, we rather propose a distributed Scheduling Function to optimize both the following performance metrics:

End-to-end delay: The packet has to be delivered to the border router before a certain deadline constraint;

Reliability: Even if links are unreliable, the system has to schedule retransmissions to guarantee a minimum end-to-end PDR.

Guaranteeing end-to-end delay is actually very challenging, since we have to also consider queuing delays. Let us consider Fig. 7.1. Typically the links CD and DE are scheduled in two consecutive cells, and the delay is minimal. On the contrary, the links BE and AB are scheduled inversely and the packet has to be enqueued in B until the next slotframe.

The slotframe may be very long since we can handle very infrequent transmissions. The slotframe duration should typically be equal to the inter-packet time. With heterogeneous periods, the slotframe length should be equal to the least common multiplier of all the packet's periods.

7.1.1 Providing High-Reliability

Unreliable links require retransmissions. However, most of the algorithms provision cells for the *average* case. Typically, most scheduling algorithms allocate 2 cells if the PDR is equal to 50%. If a packet needs more retransmissions, they will be scheduled in the next slotframe, which may be very long.

Let us assume that a link provides a PDR of $P_{ll}(link)$, i.e., ratio between the number of packets acknowledged by the receiver with respect to the total amount of sent packets. We assume here that k different cells are allocated, and thus, the transmitter can transmit at most k times this packet. The packet will be delivered from the transmitter A and correctly acknowledged by the receiver B within the current slotframe with the following probability:

$$P_{net}(A,B,k) = \sum_{i \in [1,k]} (1 - P_{ll}(A,B))^{i-1} * P_{ll}(A,B) \tag{7.1}$$

We assume here that all cells provide the same Packet Error Rate (PER) for a given link. Indeed, the frequency hopping scheme helps to mitigate external interference, so that the PER is the same for *any* cell [Cha+18].

Fig. 7.2 illustrates a numerical analysis of the network reliability ($P_{net}()$) achieved, depending on the link quality of the link ($P_{ll}(A,B)$) and the number of cells provisioned in the schedule for the (re)transmissions. Increasing the number of cells allows the transmitter to retransmit the packet. However, we need a very large number of cells to achieve a very high reliability. For a PDR of the link ($P_{ll}()$) of 50%, we need 7 cells to achieve a 99% delivery to the next hop, after the retransmissions.

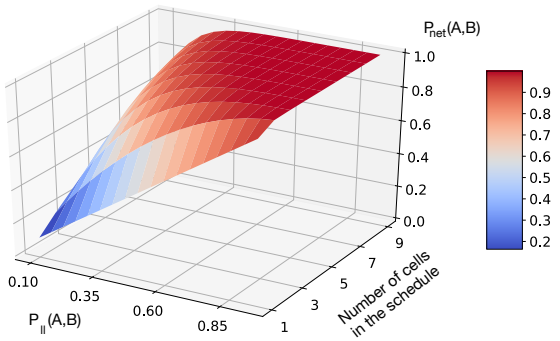


Figure 7.2: Probability to receive correctly the packet ($P_{net}()$) depending on the link reliability of the link ($P_{ll}()$) and the number of cells allocated in the schedule.

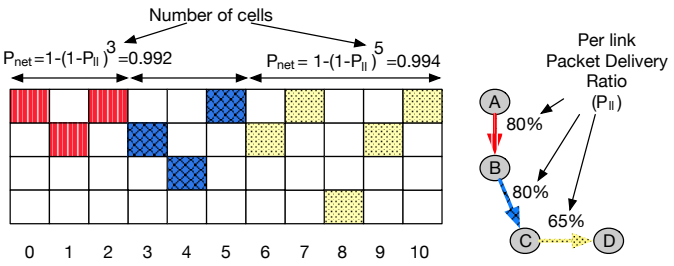


Figure 7.3: Scheduling consecutive ranges of cells to limit the end-to-end delay.

7.1.2 Delay Constraint with Dynamic Scheduling

Some scheduling algorithms reserve a *range* of consecutive cells for retransmissions to optimize the end-to-end delay [Gai+16b]. The number of cells in this range has to be sufficient to handle the worst case situation, with possibly a very large number of cells (Equation 7.1). Thus, the delay is proportional to the worst case, since it corresponds to the sum of the ranges along the path. Furthermore, negotiating cells is expensive since 6P control packets use shared cells, which are prone to collisions and, thus, increasing the convergence delay [TP16].

Even worse, such approach is inaccurate for dynamic network conditions. Indeed, a node may detect a change in the PDR, and has in that case to increase the number of cells with its parent. Unfortunately, the cells are chained along the path. Thus, inserting a novel cell requires to re-schedule all the cells until the border router. This renegotiation is time-consuming, and implies packet drops before the re-convergence.

Let us now consider the schedule illustrated in Fig. 7.3. The first two hops provide a reliability of 80% and 3 cells have to be provisioned to provide a per link reliability of 99%. However, inserting a novel cell for the radio link (AB) is expensive: we have to move all the subsequent cells. Otherwise, we have to place the retransmission cell after the 10^{th} cell, requiring to reserve novel cells for the rest of the path.

Stratum [Hos+16] tackles this reconfiguration problem by dividing the whole slotframe into stratums (one stratum per hop). However, this organization reduces the network capacity, and the end-to-end delay can only be the slotframe length.

7.1.3 Objectives of LDSF

Our proposed Low Latency Distributed Scheduling Function (LDSF) relies on the following features:

- Slotframe organization (in blocks): We reduce the end-to-end delay by chaining appropri-

Table 7.1: Notation.

$P_{ll}(A, B)$	Probability that A receives the acknowledgement from B (packet delivery ratio at the link layer)
$P_{net}(A, B, k)$	Probability that A receives at least one ACK from B after k retransmissions (packet delivery ratio at the network layer)
$MaxRetries$	Maximum number of retransmissions
$nbretx$	Current number of retransmissions for a given packet
$Cells(N_i, N_{i+1})$	number of cells from N_i to N_{i+1}
$\mathcal{S}$	Slotframe length (in timeslots)
$\mathcal{B}Length$	Block length (in timeslots)
$p = \{A..B\}$	path from A to B
hop_{delay}	hop delay, considering the link-layer retransmissions
$E2E_{delay}$	end-to-end delay
$Hops$	the hop distance between the forwarding node and the source of the packet
B_{id}	block id (number of blocks from the beginning of the slotframe)
$RxSlotId$	receiving timeslot (in the 6P request if received by the radio chipset, or given by the application layer if the packet is generated locally)
$NbGhostCells$	the number of ghost cells to add in the schedule for this flow
$\mathcal{NL}$	Network lifetime
Lf_i	Node's lifetime for the node i

ately the blocks. All nodes that are at an even number of hops (respectively odd) from the border router are scheduled during the even blocks (respectively odd). In this way, the packet progresses on average by one hop at the end of each block.

- Cell allocation in sequence: A node identifies automatically the expected earliest arrival of a packet to schedule cells to forward the packet. Then, the node reserves a cell in the next block, to minimize the buffering delay. Additionally, it also reserves the same cell every two blocks to cope with retransmissions. In this way, a node can proceed to a large number of retransmissions, to handle the worst case.

- Energy Saving: The receiver wakes-up only when a packet should be received (earliest time of arrival). We also keep a deterministic behavior, without any false negative.

7.2 Low-latency Distributed Scheduling Function

We here consider very long slotframes, where packets are generated infrequently but in a periodic manner. We propose to organize the long slotframe to reduce the end-to-end delay, by using smaller parts, that we call *blocks* (Fig. 7.4).

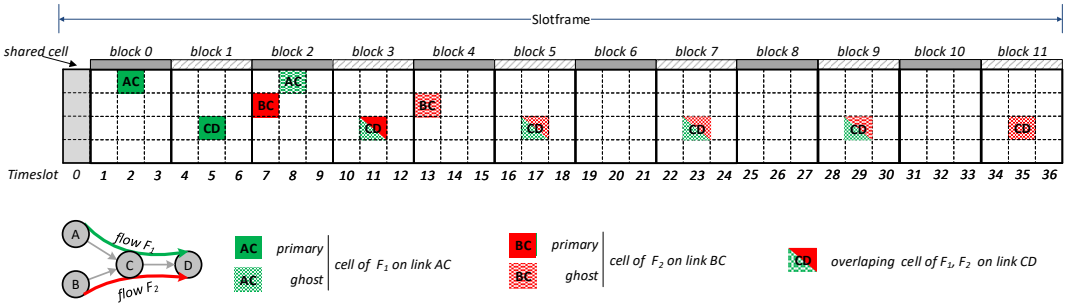


Figure 7.4: Slotframe organization in blocks, where $MaxRetries = 1$

.

7.2.1 Slotframe organization

We here consider sporadic traffic, where each sensor reports periodically its measurements to a border router. Thus, the slotframe length has to be equal to the least common multiplier of all the traffic periods. Consequently, in each long slotframe, shared cells are reserved for control traffic, such as 6P packets. One dedicated cell is also reserved for each node to send its unicast control packets corresponding to routing or synchronization. All data packets are transmitted through dedicated cells, that each pair of nodes has to reserve.

We propose to organize the long slotframe into small blocks that repeat over time.

To reduce the end-to-end delay, we have to limit the buffering delay when a packet is retransmitted. By reserving consecutive blocks for retransmissions, the buffering delay is proportional to the block size. Thus, we divide here the slotframe into small blocks with a few timeslots.

A packet is typically received during a block, and forwarded during the next one. Thus, a transmitter selects its block according to its hop count from the border router. More precisely, each block has a block id, that counts the number of blocks since the beginning of the slotframe. We have consequently even blocks (with an even block id), and odd blocks (with an odd block id). A transmitter has to select a block, so that the remainder of the Euclidean divisions of the hop count and of the block id by 2 are equal. More formally, a transmitter can select any cell in the blocks which respect the following property:

$$HC \pmod 2 = B_{id} \pmod 2 \tag{7.2}$$

where HC denotes the hop count from the transmitter to the border router, and B_{id} the block id.

Let us consider the topology and the LDSF schedule illustrated in Fig. 7.4. The node A is two hops away from the border router, and the packet is assumed to be generated in the timeslot 0. It must select a block with an even block id ($2 \pmod 2 = 0$). In our example, it selects the timeslot 2. The node C is 1 hop away from the border router, and considers only the blocks with an odd block id. It selects the block 1 (the consecutive block), and reserves one cell (here, the timeslot 5) to forward the packets from A. The blocks are daisy-chained, the packet received during the block i being forwarded in the block $i + 1$.

We make a distinction between the following types of cells:

- **Shared cell** for control packets in broadcast (EBs, routing advertisements), and control packets in unicast when no dedicated cell has been reserved (i.e., 6P requests/replies);
- **Primary** (dedicated) **cell** corresponds to the earliest expected reception time of the data packet from the previous hop (or from the application layer);

- **Ghost** (dedicated) **cells** correspond to the retransmission opportunities, that are automatically used by the transmitter if it did not receive an acknowledgment for its previous transmission. The ghost cells are scheduled in the same timeslot and channel offset as for the corresponding primary cell, but in the subsequent blocks.

When a node reserves a primary cell in a block, a fixed number of ghost cells is automatically reserved every two blocks. Thus, we can daisy chain the transmission opportunities along the path: a node is able to receive a packet during a block, and forward it during the subsequent blocks. In this way, we maintain a low end-to-end delay.

A link quality degradation means more retransmissions: in classical scheduling algorithms, we would need to reserve additional cells. Here, we pre-reserve a large number of cells, at the very beginning, to cope with the worst link qualities. Thus, the number of ghost cells (for retransmissions) is fixed, whatever the link quality.

Besides, the impact of the retransmissions on the end-to-end delay is limited since the blocks comprise a small number of timeslots. We use the Automatic Repeat reQuest (ARQ) feature of IEEE 802.15.4-TSCH: the transmitter schedules a retransmission in the next ghost cell only if it does not receive an acknowledgement for its previous transmission.

7.2.2 Number of Ghost Cells

We have now to compute the number of ghost cells to provision for the retransmissions.

Standard case

The delay induced by the retransmissions is cumulative along the path. Thus, we have to cope with the worst case: a packet may be retransmitted at most $MaxRetries$ times by each transmitter in the path. The latest time of arrival corresponds to the last RX ghost cell (e.g., C receives the packet from A at the latest during the timeslot 8 in Fig. 7.4).

We make here a distinction between the *source* of the flow that generates a data packet, and a *transmitter* that forwards this packet. The first transmitter in the path corresponds trivially to the source.

We can note that the number of ghost cells is proportional to the hop distance from the source. More precisely, a transmitter has to provision $(MaxRetries * (Hops + 1))$ ghost cells for the retransmissions, where $Hops$ denotes the hop distance from the source to the transmitter.

In Fig. 7.4, A is the source ($Hops = 0$) and provisions one primary cell (timeslot 2) and one ghost cell (timeslot 8). For the node C, it is one hop away from the source ($Hops = 1$). Thus, C allocates for the flow F_1 one primary cell (timeslot 5) and 2 ghost cells ($MaxRetries*(Hops+1)$). We can note that the node C can receive the packet through the primary or the ghost cells. Thus, even if it receives a packet during the last ghost cell (timeslot 8), it has still two transmission opportunities (timeslots 11 and 17) for one transmission, and one retransmission.

Overlapping case

Some flows may overlap, i.e., one relaying node uses the same ghost cells for two different flows. For instance, flows F_1 and F_2 are both forwarded by the node C, where some ghost cells are in common for both flows. A node can easily detect an overlap when receiving a 6P request: the primary cell corresponds to a ghost cell already reserved for another flow.

Even with this overlap, we must be sure to have enough ghost cells to handle the worst case. Let us consider the two following cases:

Case 1) the node receives a packet from the novel flow (F_2) while a packet from the previous flow (F_1) is already in the queue. By construction, the first flow F_1 has still enough ghost cells to handle $MaxRetries$ retransmissions. At the latest, the packet for the flow F_2 is received while only $MaxRetries+1$ ghost cells remain in its schedule (primary transmission + retransmissions). Thus, we need to provision $MaxRetries+1$ ghost cells for the novel flow F_2, after the ghost cells that would have been allocated to the flow F_2.

Case 2) the node receives a packet from the novel flow (F_2) while the packet from the other flow (F_1) was not yet received. For the same reason, the node has enough ghost cells for F_2 for $MaxRetries$ retransmissions. Thus, we have also to insert in that case $MaxRetries + 1$ additional ghost cells at the end of the range, but they will be used to forward the packet for the flow F_1.

In conclusion, it is sufficient to provision $MaxRetries + 1$ additional ghost cells when an overlap is detected, whatever the hop distance from the source.

7.2.3 Scheduling process

We now detail how the cells are reserved by each pair of nodes. Shared cells are only used for signaling, i.e., sending/receiving the 6P packets to negotiate which dedicated cells to use. A 6P request typically piggybacks a list of possible (dedicated) primary cells. The (dedicated) ghost cells are automatically derived from a primary cell. The receiver sends a 6P reply to the transmitter to validate the reservation. Since no dedicated cell is present in the schedule, the 6P packets use the shared cell.

A novel allocation is triggered when a node receives either a 6P request from the previous hop or directly the packet from the application layer. Thus, we propose the following procedure (see Algorithm 4):

1. First of all, we need to allocate the receiving cells if we receive a 6P request. We reserve in Receiving mode all the timeslots every 2 blocks (lines 1-7), located after the timeslot specified in the 6P request.

   We allocate one primary cell, and ($MaxRetries * Hops$) ghost cells for the retransmission. The value $Hops$ comes from the fact that the receiver for the 6P request is one hop farther from the source than the transmitter.

   Please note that the pseudo-random function `pseudoRandom()` is executed with the same argument by the receiver and the transmitter to derive the same channel offset (lines 3, and 16).

2. We identify the block which is directly located after the block of the first receiving timeslot (line 8). We will schedule the TX cells after this block;

3. We have to allocate the primary cell, and ($MaxRetries * Hops$) ghost cells (line 10);

4. We have then to make a distinction between the two possible cases:

   - **Overlapping** flows: A cell in this block is already assigned (line 11). Thus, the allocation will re-use the same timeslot and channel offset (line 12). We have also to allocate ($MaxRetries + 1$) additional ghost cells to handle the worst queuing delay with the overlapping flow (line 13).
   - **No-overlap:** We select randomly the timeslot and channel offset (line 15-16).

5. When we have determined the number of ghost cells, and the first timeslot to allocate, we can proceed to the allocation (lines 18-21). It is worth noting that with overlapping flows, some TX cells may be reserved by several flows (e.g., timeslot 11 in Fig. 7.4).

Let us illustrate this scheduling algorithm with Fig. 7.4. For the sake of better representation, we assume that the maximum number of retransmissions is equal to one ($MaxRetries = 1$):

- As explained previously, A selects a cell in the block 0. It also reserves automatically one associated ghost cell ($MaxRetries * 1$ *hop*) in block 2. After the 6P reservation, the primary and ghost cells are reserved for both the transmitter and the receiver.

Algorithm 4: Cell allocation process

Input:
`Schedule`: the current schedule
`Hops`: the hop distance between the node and the source of the packet
`RxSlotId`: receiving timeslot (in the 6P request if received by the radio chipset, or given by the application layer if the packet is generated locally)
$\mathcal{S}$: slotframe length
$\mathcal{B}Length$: block length
Output:
Schedule: the schedule updated with the novel cells

```
// Allocation of the slots in Receiving mode, only if the packet is received from the radio chipset
if packetNotReceivedFromApplication() then
    for nbretx ∈ [0, MaxRetries * Hops] do
        ChOff ← pseudoRandom(0, NbChannels − 1)
        // select the timeslots at regular intervals (every 2 blocks) after the timeslot present in the 6P request
        TsOffset ← (RxSlotId + nbretx * 2 * BLength) % S
        Schedule.AddCell(TsOffset, ChOff,' RX')
    end
end
// Identify the block which is located just after the first receiving timeslot (allocated in the previous loop)
TxSlotId ← (RxSlotId + (RxSlotId % BLength) ) % S
// Allocation of TxSlot (transmitter side)
if node ≠ BorderRouter then
    NbGhostCells ← MaxRetries * (Hops + 1)
    // If we have overlapping flows, pick-up the TX cell already allocated in the block
    if GetBusyTXSlotnBlock(TxSlotId) ≠ ∅ then
        {TxSlotId, ChOff} ← GetBusyTXSlotnBlock(TxSlotId)
        // ghost cells to handle queuing delays
        NbGhostCells ← NbGhostCells + MaxRetries + 1
    else
        // Select randomly one timeslot in the block
        TxSlotId ← TxSlotId + random(BLength)
        ChOff ← pseudoRandom(0, NbChannels − 1)
    end
    // Allocates the corresponding slots in the schedule (every 2 blocks)
    for nbretx ∈ [0, NbGhostCells] do
        TsOffset ← (TxSlotId + nbretx * 2 * BLength)% S
        Schedule.AddCell(TsOffset, ChOff,' TX')
    end
end
return Schedule
```

- The earliest time of arrival for the node C corresponds to the timeslot 2. Thus, it reserves a cell in the next block (1). It also reserves two ghost cells ($MaxRetries * 2hops$), in the blocks 3 and 5;

- We can note that C is also forwarding the flow F_2 (from B). Its primary cell for the flow F_2 is already reserved as ghost cell for the flow F_1: C detects an overlap. Thus, it first reserves one primary cell (timeslot 11) and 2 ghost cells (timeslots 17 and 23) for the flow F_2. Because, of the overlap, it has also to allocate additional ghost cells to consider the latest time of arrival (cf. section 7.2.2.0). Thus, it allocates two additional ghost cells (one primary cell + $MaxRetries$) after its last ghost cells. In conclusion, it selects the timeslots 29 and 35 as ghost cells.

Since we rely on a distributed random scheduling algorithm, two interfering links may select the same cell. A collision may occur if both transmitters select the same primary cells, or if the ghost and primary cells overlap. For instance, the links AB and CD in Fig. 7.1 may reserve the same timeslot and channel offset. Then, if both nodes A and C send the packet over the same

cell, there will be a collision. Since, in our scenarios, the packet generation period is sporadic, and because we consider long slotframes, the probability that two or more transmitters to transmit over the same cell is very low. In case of a collision, a relocation mechanism [Cha+18] will be applied.

7.2.4 Energy Savings using Ghost Cells

Reserving ghost cells allows the network to improve the reliability: LDSF can efficiently handle the fast link quality changes since ghost cells are a priori over-provisioned. Concerning the energy efficiency, the transmitter can safely sleep when it has no data packet to transmit. For the receiver side, we have to limit idle listening [Vil+14], forcing the node to wake-up at the beginning of the timeslot because it is not aware that the transmitter has nothing to transmit.

Under the LDSF algorithm, we configured a fixed number of ghost cells, based on the hop distance from the source, and a constant, whatever the link quality. A receiving node must wake-up at the beginning of each primary cell, to possibly receive a packet. Then, it must also wake-up for all the subsequent ghost cells until a packet has been received and correctly acknowledged. Once, a packet has been received or the last ghost cell is encountered, the receiver can safely save energy until the next primary cell. The receiver has then to forward the packet, and becomes a transmitter. It selects the corresponding cell in the next blocks, and starts to transmit the packet to the next hop.

Let us consider the scenario illustrated in Fig. 7.4. Let us assume that the node C has been able to decode the packet from the node A in the timeslot 2. It can stop listening to the ghost cell in timeslot 8. However, it will wake-up during the next block (timeslot 5) to forward the packet to the node D.

The primary cell corresponds to the earliest time arrival to optimize the end-to-end delay. Thus, we do not have any false negatives: the receiver is always awake when the transmission takes place, we thus keep the deterministic behavior of IEEE 802.15.4-TSCH.

7.3 Mathematical Analysis

Let us analyze next the end-to-end delay, i.e., average time required to deliver a data packet from a node A to the border router S via a path $p = \{N_i\}_{i \in [1,||p||]}$, where $||p||$ denotes the number of nodes in the path p.

7.3.1 Model

To obtain a fair evaluation, we use the same mathematical model to compare each scheduling approach (cf. notation in Table 7.1). We have made the following considerations:

- We consider the worst case delay, when the data packet is enqueued at the beginning of the slotframe;
- We assume that the inter packet time is sufficiently large, and the queue for the nodes is empty at the beginning of the slotframe.

We apply here the same methodology as in [Hos+16] to compute the average time before a packet is delivered to the border router (i.e., the end-to-end delay).

MSF

Since cells are randomly scheduled, we consider a uniform distribution of the cells in the slotframe.

To be generic, we compute the normalized delay in number of timeslots. We have to multiply the normalized delay by the timeslot duration T_{slot} to compute the actual delay. Compared

with [Hos+16], we have the following normalized delay:

$$hop_{delay}(N_i, N_{i+1}) = \frac{\mathcal{S}}{2 * Cells(N_i, N_{i+1})} \tag{7.3}$$

where $\mathcal{S}$ denotes the slotframe length, i.e., number of timeslots.

If the radio link between N_i and N_{i+1} is unreliable, the transmitter needs $\frac{1}{P_{ll}(N_i,N_{i+1})}$ transmissions before the packet is successfully decoded and acknowledged. So using Equation 7.3:

$$hop_{delay}(N_i, N_{i+1}) = \frac{\mathcal{S} * \frac{1}{P_{ll}(N_i,N_{i+1})}}{2 * Cells(N_i, N_{i+1})} \tag{7.4}$$

Finally, the average end-to-end delay to deliver a packet from a source node to the border router along the routing path $p = \{N_i\}_{i\in[1,||p||]}$ is:

$$E2E_{delay}(p) = \mathcal{S} * \sum_{i=1}^{||p||-1} \frac{1/P_{ll}(N_i, N_{i+1})}{2 * Cells(N_i, N_{i+1})} \tag{7.5}$$

Stratum

If we consider the length of each stratum is sufficient to handle retransmissions, the maximum end-to-end delay corresponds to the slotframe length ($\mathcal{S}$). In this case, the packet is delivered successfully before the end of the stratum:

$$E2E_{delay}(p) = \mathcal{S} \tag{7.6}$$

If the number of cells is insufficient, the retransmissions may be scheduled during the subsequent slotframe, increasing the end-to-end delay by one slotframe length. We may use the distribution of the packet delivery success according to Equation 7.1 to derive the distribution of this additional delay. It depends also on the length of each stratum, since it upper bounds the number of cells we can allocate. While Equation 7.6 corresponds to an optimistic case, we consider that the stratum length is correctly sized to handle the worst case situation.

LDSF

The cells are pseudo-randomly allocated in a block. Besides, the node N_{i+1} will forward a packet it received from N_i in the just immediately consecutive block (by construction).

Thus, the average buffering time is equal to the block length:

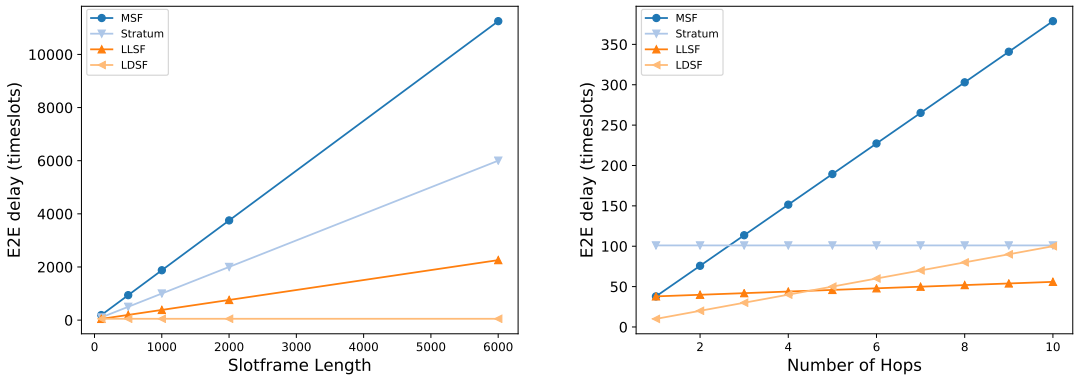
$$hop_{delay}(N_i, N_{i+1}) = \mathcal{B}Length \tag{7.7}$$

If the radio link between N_i and N_{i+1} is unreliable, they require $1/P_{ll}(N_i, N_{i+1}) - 1$ retransmissions. Since consecutive primary/ghost cells are interspaced by $2 * \mathcal{B}Length$ timeslots, we have:

$$hop_{delay}(N_i, N_{i+1}) = \mathcal{B}Length + \mathcal{B}Length \left(\frac{2}{P_{ll}(N_i, N_{i+1})} - 1 \right) \tag{7.8}$$

Finally, the average end-to-end delay is:

$$E2E_{delay}(p) = \mathcal{B}Length \sum_{i=1}^{||p||-1} \left(\frac{2}{P_{ll}((N_i, N_{i+1})} - 1 \right) \tag{7.9}$$



(a) Flow through a route of 5 hops.

(b) Slotframe with 101 timeslots.

Figure 7.5: Impact of the slotframe length and number of hops on the end to end delay with a PDR per link of 66%.

LLSF

the approach daisy-chains the cells along the path, relying on relocation when they are not contiguous because of retransmission cells. Thus, we will here focus on the steady-state, after the convergence, i.e. we neglect the effect of suboptimal relocations.

The first hop delay is obtained similarly to MSF. Since the nodes select the transmission cells randomly on the first hop, the delay of the first hop is:

$$hop_{delay}(N_1, N_2) = \frac{\mathcal{S} * \frac{1}{P_{ll}(N_1,N_2)}}{2 * Cells(N_1, N_2)} \tag{7.10}$$

Then, the end-to-end delay is similar to LDSF. Indeed, two cells for the retransmissions are now consecutive instead of being separated by $2 * \mathcal{B}Length$ timeslots. The average end-to-end delay is thus:

$$E2E_{delay}(p) = hop_{delay}(N_1, N_2) + \sum_{i=2}^{||p||-2} \left(\frac{2}{P_{ll}(N_i, N_{i+1})} - 1 \right) \tag{7.11}$$

7.3.2 Numerical results

We consider here a scenario, with unreliable radio links (i.e., $P_{ll} = 66\%$). Thus, two cells are allocated for each link ($Cells = 2$) for possible retransmissions, and the block length is fixed to 5 timeslots ($\mathcal{B}Length = 5$).

Fig. 7.5a illustrates the impact of the slotframe length. The end-to-end delay of both MSF and Stratum is linear with the slotframe length. Indeed, Stratum can only guarantee a delay equal to the slotframe length. As can be observed, the performance of LLSF is affected by the slotframe length due to the selection of the transmission cells randomly at the first hop. On the contrary, the delay of LDSF is independent of the slotframe length: the cells are chained along the path.

Similarly, we can see that the delay increases with the number of hops for MSF (Fig. 7.5b): the TX and RX cells are uniformly distributed, and a node has to buffer the packet for a long time before being able to forward it. On the contrary, Stratum provides a delay independent of the hop length. However, this hop distance cannot exceeds the number of stratums, which is 10 in this case. LDSF achieves to provide a much smaller delay, increasing only slightly for long routes. More precisely, the delay is increased by one block length per hop, i.e., 5 timeslots. Finally, the LLSF achieves low delay too especially when the number of hops increases.

7.4 Performance Evaluation

We now assess the performance of our distributed scheduling function in a more realistic environment, where packets can be dropped because of collisions and low link qualities.

7.4.1 Simulation Setup

We use here the 6TiSCH Simulator [Mun+19], a discrete-event simulator written in Python. We consider the following scenarios:

1. CBR flows;
2. Topologies with a variable number of nodes (by default 39 nodes and one border router) to assess the scalability of our Scheduling Function.

We generate random topologies, where each node is randomly located in an area of 2000×2000 m^2. Thus, each node has at least 3 neighbors. The propagation model of 6TiSCH Simulator is based on the Pister-Hack model [Le+09]. Table 7.2 regroups all the values of the different parameters.

We extended the 6TiSCH Simulator with three additional scheduling functions (LDSF, LLSF [Cha+16] and Stratum [Hos+16]). For the LDSF and Stratum algorithms, each node has to know its hop distance from the border router. We implement in Stratum the opportunistic deallocation of cells described in [TP16], where an unused cell is removed after a long timeout to avoid schedule's inconsistencies.

The node's lifetime is:

$$Lf_i = \frac{Battery_{Capacity} \times SF_{duration}}{Q_{slotframe} * 3600 * 24 * 365} \tag{7.12}$$

Where $SF_{duration}$ is the duration of the slotframe (in seconds), $Battery_{Capacity}$ is the node's battery capacity (in μC), $Q_{slotframe}$ is the average energy drawn during a slotframe (in μC), and $(3600 * 24 * 365)$ allows to convert seconds into years. To compute the lifetime of each node, we rely on the energy model described in [Vil+14] that relies on real measurements. The model provides the power of the node in each state:

TxDataRxAck: the transmitter sends a packet and waits for an acknowledgement;

TxData: the transmitter sends a packet without waiting for an ACK (e.g., a broadcast packet);

RxDataTxAck: the receiver decodes a packet and sends an ACK;

RxData: the receiver decodes the packet, and switches directly to sleeping mode, without sending an ACK;

Idle: the receivers listens to the medium but does not sense anything;

Sleep: the nodes turns off its radio chipset.

Thus, the simulator [Mun+19] has just to count the time spent in each state, combined with the values measured in [Vil+14] and reported also in Table 7.2 to compute the lifetime metric.

We measure here the network lifetime as the time until the first node dies because it runs out of energy (n-of-n lifetime in [DD09]). We assume that the system reaches a steady state, and we just have to identify the node with the largest energy consumption. Finally, the network's lifetime is:

$$\mathcal{NL} = min\{Lf_i\}_{i \in [1, ||Nodes|| - 1]} \tag{7.13}$$

We exclude from our estimation the border-router since we assume that is not powered by a battery.

Table 7.2: Simulation setup.

Topology	**Parameter**	**Value**
	# of nodes	39 + 1 border router
	# of Experiments	20 per algorithm
Simulation		
	Duration	60 min
	Payload size	90 bytes
Protocol		
CoAP	CBR (*Unicast*)	1 $pkt/[5, 10, 20..120]sec$
RPL	DAO period	60 s
	DIO period	8.5 s
TSCH	NShared cells	1
	Timeslot duration	10 ms
	Maximum retries	5
MSF, Stratum & LLSF	Slotframe length	101 timeslots
LDSF	Slotframe length	$CBR * 101$ timeslots
	Block length	5 timeslots
Queues	Timeout	10 s
	Queue size	10 packets
Energy		
	Energy Model	[Vil+14]
	$Battery_{Capacity}$	10157.4×10^6 μC
	$Idle$	6.4 μC
	$Sleep$	0 μC
	$TxDataRxAck$	54.5 μC
	$TxData$	49.5 μC
	$RxDataTxAck$	32.6 μC
	$RxData$	22.6 μC

7.4.2 Scheduling Algorithms

We compare the following approaches:

MSF: [Cha+19] is the default scheduling function of 6TiSCH, where autonomous (pseudo-random) cells are used for control traffic and dedicated cells are used for data packets;

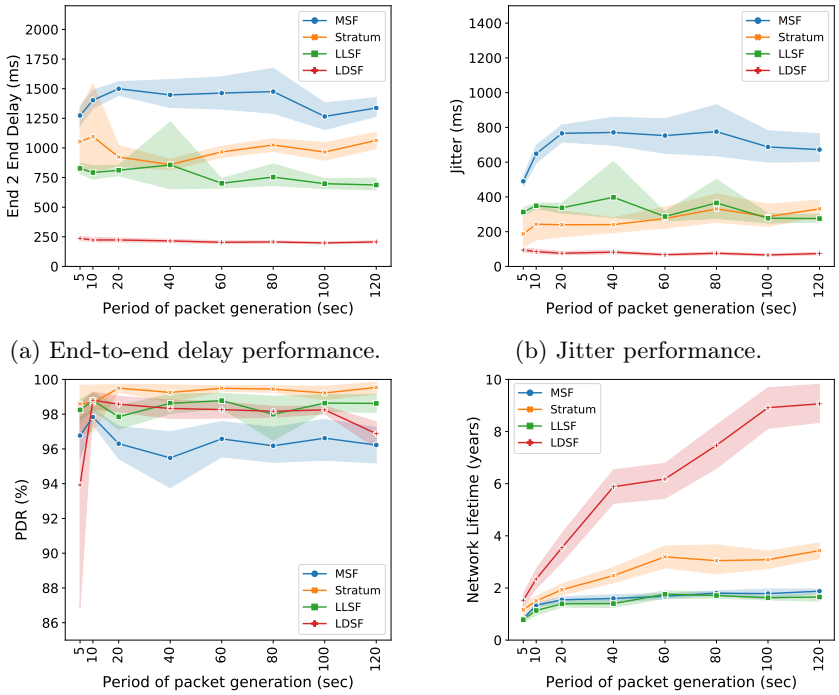
Stratum: [Hos+16] divides the slotframe in blocks (i.e., stratums). Each node selects a block according to its hop distance, so that a packet is delivered in the current slotframe;

LLSF: [Cha+16] aims to reduce the end-to-end latency by allocating receiving and transmitting cells as close as possible in the schedule.

LDSF: Our scheduling function described in section 7.2.

Stratum uses a slotframe length of 101 timeslots, to be able to provide an end-to-end delay equal to 1010 ms (=101 * 10ms). MSF and LLSF also uses the default slotframe length (101). LDSF uses rather a slotframe length proportional to the maximum flow rate, since it was designed for this purpose. The same cell is used every two blocks for transmissions and retransmissions. Since each block comprises 5 timeslots in LDSF, the transmitter has to wait on average 10ms * 5 timeslots * 2blocks = 100ms.

Our implementation (simulation code, scripts, and raw data) is freely available (`https://github.com/vkotsiou/Scheduling` for the implementation, and `https://doi.org/10.5281/zenodo.3748712` for our dataset).



(a) End-to-end delay performance.

(b) Jitter performance.

(c) End-to-end packet delivery ratio.

(d) Network lifetime.

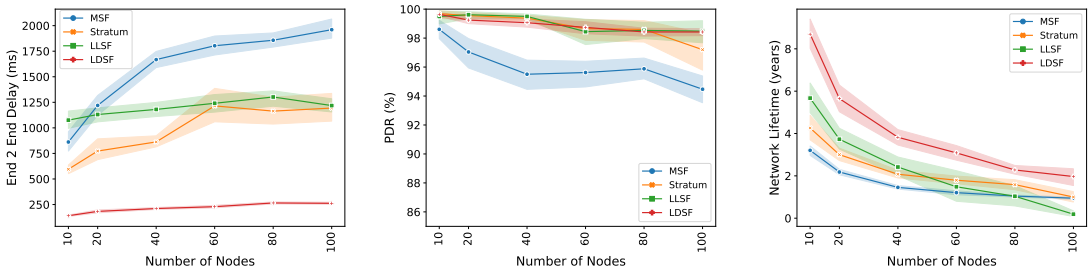
Figure 7.6: Impact of the traffic rate (i.e., inter packet time).

7.4.3 Traffic Rate

We first measure the average end-to-end delay (Fig. 7.6a). LDSF is very robust to large traffic rates: it keeps on providing a very low delay. Stratum presents a very stable end-to-end delay: packets are delivered at the end of the slotframe exactly (101 timeslots * 10ms). MSF presents the highest end-to-end delay because it does not have any cell allocation strategy to minimize the delay: it picks randomly the cells. LLSF provides also a very stable delay. While the cells are reserved consecutively along the path, the first cell is picked randomly and generates a large buffering delay (half of the slotframe = 505ms).

We can make the same remarks concerning the jitter (Fig. 7.6b). LDSF provides the lowest jitter performance which is less than 150 ms, even for very high traffic rates. Collisions are accurately handled, and the packets are retransmitted efficiently in the subsequent blocks to minimize the buffering delay. LLSF achieves a larger jitter: the schedule is modified as soon as some retransmission cells have to be inserted. However, this optimization has a cost since the whole schedule has to be modified along the path. Stratum provides jitter performance similar to LLSF, corresponding to the length of the last *stratum* (i.e., block). Indeed, a packet is randomly scheduled in the last *stratum*to be received by the border router. Since this *stratum* is typically much larger than the LDSF's block, the jitter is mechanically increased. Finally, MSF provides the worst jitter since retransmission cells can create a cumulative effect along the path, since they can be allocated after the cells of the next hop.

Fig. 7.6c focuses on the reliability. The three schemes are able to guarantee end-to-end reliability above 96% in most cases. Stratum achieves the highest reliability for low traffic rates since the blocks are large to avoid collisions. However, the number of collisions starts to increase for high traffic rates (inter packet time $<$ 10s). LDSF is able to provide an end-to-end packet delivery ratio higher than 98%. LLSF provides also a good reliability, except for high-traffic rates: many collisions arise and are particularly challenging to resolve since the cells are contiguous. Moreover, the scheduling process needs to solve the collisions for each cell, while LDSF is more robust since the same cell is pre-reserved also for the retransmissions.



(a) End-to-end delay (b) End-to-end packet delivery ratio (c) Network lifetime.

Figure 7.7: Impact of the number of nodes

.

Fig. 7.6d illustrates the network lifetime. We extrapolate the average energy consumption for the most loaded node to derive the network lifetime. MSF generates a large number of control packets with many (de)allocations, which impact negatively the lifetime. Stratum increases slightly the lifetime, by reducing the renegotiation of cells. LLSF achieves the same characteristics since it uses a short slotframe (101 timeslots) and that shared cells consume energy. Finally, LDSF is very efficient to handle unreliability: ghost cells are automatically reserved after a 6P transactions, minimizing the amount of control traffic. Thus, the lifetime increases for very low traffic conditions.

7.4.4 Scalability

We then measure the scalability of these distributed scheduling functions by increasing the number of nodes up to 100. LDSF is very scalable; even a large number of nodes does not increase significantly the collisions. The delay remains below 200 ms. The delay of MSF and Stratum is much higher, while the delay of LLSF scales smoothly with respect to the number of nodes. MSF provides the lowest end-to-end reliability (Fig. 7.7b): some packets are dropped because of an excessive number of retransmissions, or because of a buffer overflow. Stratum, LLSF and LDSF achieve to still provide a very high reliability even with 100 nodes generating one packet every 20 seconds. More than 98,5% of the packets are delivered to the border router.

LDSF achieves the higher network lifetime whatever the number of nodes (Fig. 7.7c). The ghost cells are pre-reserved, but the transmitter and the receiver do not have to wake-up in every ghost cells. More precisely, they will wake-up only if no acknowledgement was received correctly. In this way, LDSF provides the larger lifetime compared against MSF, LLSF and Stratum approaches.

7.4.5 Slotframe occupation

Finally, Fig. 7.8 illustrates the percentage of the slotframe occupied, i.e., the cells are allocated to at least one transmitter. We can note that MSF and LLSF use short slotframes, thus the ratio of shared cells is higher compared to long slotframes. Thus, idle listening will consume a large quantity of energy. Stratum may generate more collisions since interfering links have often to select their cell in the same stratum. Thus, the number of allocated cells is reduced, but keeps larger than LDSF. LDSF organizes the cells appropriately and can exploit long slotframes (as long as the CBR period). While LDSF reserves ghost cells for the retransmissions, they are used only if the transmission has failed. If the packet was acknowledged, neither the transmitter nor the receiver will wake-up during the next ghost cells (cf. amount of unused cells in Fig. 7.8).

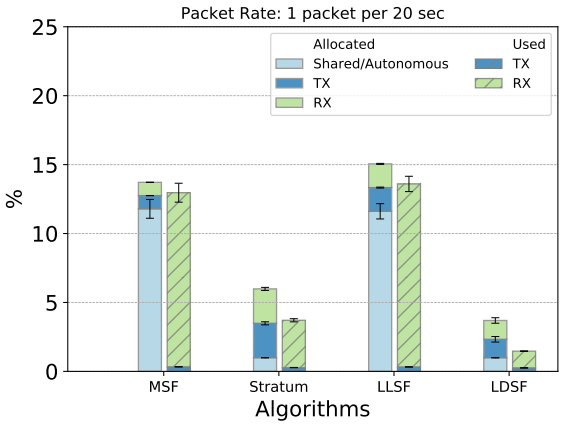


Figure 7.8: Percentage of the slotframe occupied (i.e., cells allocated), with one packet generated by each node every 20 seconds.

7.5 Conclusion and Perspectives

We proposed here LDSF, a scheduling function able to minimize the end-to-end delay by chaining the cells along the path. Instead of classical approaches, LDSF does not need to reschedule all the cells toward the destination when the quality of a specific link changes. LDSF is rather designed to handle the worst case scenario, provisioning enough cells for retransmissions. It divides the long slotframe into small blocks that repeat over time. Each node selects the right block corresponding to its hop distance from the border router to minimize the delay. Moreover, ghost cells are automatically reserved in the slotframe to cope with retransmissions. Besides, a device stays awake during these ghost cells only if the previous transmission has failed, to save energy. Our simulation results demonstrate that LDSF can achieve a low latency and jitter with high reliability, even for multi-hop topologies.

In a future work, we aim at exploring how our scheduling solution may be combined with the blacklisting techniques. We have still to provide collision-free schedules, daisy-chained along the path to the border router, while exploiting per radio link blacklists. Furthermore, we will study how LDSF can be extended to support a network where nodes have not the same packet generation period. Finally, in the future, we aim to extend LDSF so that it implements the concept of tracks as introduced in 6TiSCH. A track is a set of cells along a path towards the border router that is mapped to a given data packet flow; we consider that each application generates its own data packet flow.

Chapter 8

Conclusions and Perspectives

This Chapter concludes the manuscript, reminding the addressed problems, highlighting the contributions, and opening up perspectives.

8.1 Conclusions

The goal of this dissertation was to provide reliable communications to IIoT. Our research campaign focused on two sub-objectives. First of all, we focused on improving the reliability of radio communications, which are subject to external interference and the multi-path fading effect. We proposed techniques to improve the slow channel hopping mechanisms, a widely adopted technique for coping with unreliable radio links, by the application of a Blacklisting/Whitelisting technique. The other objective was to provide low and bounded end-to-end delay and high PDR performance at the same time by proposing a distributed scheduling function. Since IEEE 802.15.4-TSCH and 6TiSCH became the defacto protocols for IIoT, our research campaign was based on them.

We started by conducting series of experiments in the FIT IoT-LAB, an indoor testbed, to characterize the IEEE 802.15.4 radio channels. The ultimate goal of characterizing the radio channels was to exhibit the necessity of a blacklisting technique to address the unreliability of radio links as well as to specify the features that such a technique should have. The analysis of our experiments showed that a significant portion of radio channels exhibit variations in their quality over time. Moreover, they highlighted that the spatial characteristics of a radio link, such as its location and its geographical distance, significantly affect the quality of the different radio channels. Our experiments also showed that the assessment of the radio channels should not be based exclusively on RSSI since the correlation between RSSI and PDR is weak. Taking advantage of the above findings of our research, we conclude that it is necessary to develop a blacklisting technique since the channels of radio links present diversity in their quality. The blacklisting technique should be adaptive and local per link due to the spatial and temporal characteristics of the radio links.

In Chapter 4, motivated by the key findings of the previous chapter, we implemented a distributed adaptive per link blacklisting technique called LABeL. LABeL exploits our proposed dynamic threshold algorithm to classify the channels as *bad* or *good*, overcoming the drawbacks of applying a fixed threshold or a pre-defined number of channels classification technique (cf. section 2.3.1.0). LABeL addressed the issue of blacklists' inconsistency by implementing a handshake protocol where 6P control packets are exchanged. We also modified the channel hopping sequence to keep on probing the bad radio channels to recover without burdening the network traffic with extra packets. Furthermore, LABeL applied a modification technique of channel hopping sequence that disperses the internal collisions across the spectrum, making them less repetitive to be considered as external interference. Our experimental study showed that LABeL improves the network's reliability by 20% in comparison to a *blind* (all radio channels are used

evenly) channel hopping technique (e.g., IEEE 802.15.4-TSCH). Finally, LABeL achieves superior performance compared to the technique where the network administrator manually blacklists the low-performance channels globally, as applied by WirelessHART.

Then, in Chapter 5, we proposed centralized whitelisting algorithms to cope with the problem of internal collisions. First of all, the formalization of the above problem, highlighted that the internal collisions are repetitive, and the frequency of their occurrence depends on whitelists' size of the interfering links. Our first approach (Whitelist-Aware) was to exploit a centralized scheduling algorithm (e.g., TASA), where the links that can cause internal collisions are scheduled in separate timeslots. While it avoids internal collisions, such an approach also increases the slotfame length, and thus, the delay. Then, we proposed a whitelisting algorithm (Commom Whitelist per Timeslot) that forces all the radio links scheduled in the same timeslot to use the same whitelist. Our latest approach (Whitelist Re-ordering) reorders the whitelists of the radios links that are scheduled at the same timeslot in such a way that do not cause internal collisions. The evaluation performance of our proposals through simulations highlighted that the strategies (Whitelist Aware, Whitelist Re-Ordering) with local per link whitelists, present the highest reliability.

In Chapter 6, we proposed AMABO, an adaptive per-link blacklisting technique that eliminates internal collisions by exploiting the multiple channel offset assignment technique of MABO-TSCH. Moreover, AMABO assigns the multiple channels offsets to the radio links dynamically in per timeslot basis, in contrast to MABO-TSCH, which assigns a fixed number of channel offsets per link. The AMABO's technique achieves to exploit the full range of the available channels at any given time, thus increasing the number of channels assigned per radio links. As a consequence, it increases the probability to use a good channels for the data transmissions. Our evaluation performance campaign demonstrates AMABO's improvement on PDR, particularly for long blacklists compared to MABO-TSCH.

Algorithm	Temporal Variability	Spatial Variability	Blacklist Scope	Blacklist Construction	Scheduling Algorithm	Internal Collisions
LABeL	✓	✓	per Link	Local/Sender	Distributed	Probabilistic
Whitelist Aware	✓	✓	per Link	Local/Sender	Centralized	Limited
Common Whitelist per Timeslot	✗	✗	Common per Timeslot	Centralized/PCE	Centralized	No
Re-Ordering	✗	✓	per Link	Centralized/PCE	Centralized	No
AMABO	✓	✓	per Link	Local/Sender Channel Offsets/PCE	Centralized	No

Table 8.1: An overview of our proposed blacklisting/whitelisting techniques.

The main characteristics of our proposed blacklisting/whitelisting algorithms are presented in Table 8.1. As it can be observed, each of our solutions has specific advantages and limits. More specifically, our distributed proposal LABeL is best suited for industrial networks where the main concern is scalability and flexibility, and where the traffic is sparse and without a large number of parallel transmissions to cause extensive internal collisions. On the other hand, when the network traffic is intense, compact schedules are employed, there is a requirement for optimal reliability, and the external interference is somehow steady over time. Our centralized proposals achieve the best performance. Finally, AMABO, our hybrid proposal, seeks to reconcile scalability, high reliability, and immediate adaptation to time variations of the external interference by achieving near to optimal performance.

Finally, in Chapter 7, we proposed LDSF, a scheduling function intended to 6TiSCH networks,

which achieves a low end-to-end delay, high reliability, and low power consumption at the same time. Thus LDSF, divides the slotframe into repetitive short blocks. There are two kinds of blocks, in the first one the node listens for incoming packets from its children, and in the second one, it transmits one packet to its parent. By exchanging the types of blocks repeatedly, a chain of transmitting/receiving blocks is created towards the sink, reducing the end-to-end delay. This slotframe organization reserves automatically consecutive blocks for possible retransmissions to cope with the lossy links and to preserve the raise of the end-to-end delay to a low level. LDSF, based on the periodic traffic pattern of a typical WSN, allocates cells in the schedule only when traffic expected, taking into account also, all possible retransmissions along the path to the sink, thus saving energy. Our evaluation campaign through the 6TiSCH simulator showed that LDSF achieves low and deterministic end-to-end delay without sacrificing the reliability, and it even increases the network lifetime.

8.2 Perspectives

The contributions of this thesis can be extended in several directions. Let us now present some of them.

8.2.1 Experiments

In this manuscript, we evaluated and validated our contributions using a variety of methods such as indoor testbed's experiments, simulations using our custom-made trace-driven python simulator, and the 6TiSCH simulator. We have chosen to use simulations in our evaluation campaign to judge fairly our proposals and the state of the art algorithms under the same radio conditions. However, it is necessary to use real experiments to evaluate the algorithms we have proposed so far, and we have only assessed them through simulations. The above will allow us to validate the correctness of our proposals and to study them in real conditions.

Our intentions in the future are to expand our experimental research campaign of the characterization of IEEE 802.15.4 radio channels to other indoor testbeds (e.g., FIT IoT-LAB's testbeds) to investigate the qualitative and quantitative differences between the testbeds. It would also be ideal to comprise in this campaign industrial testbeds where different environmental conditions prevail, such as large metal reflective surfaces, dust, noise from machinery, high-temperature, etc. Finally, the expansion of the performance evaluation of LABeL, in a multitude of testbeds (indoor, outdoor and industrial) would give us the opportunity to investigate: a) how we can adapt the parameters of WMEWMA to the specificities of each testbed, b) the effectiveness of our dynamic threshold algorithm and c) what is the appropriate probing rate of the bad channels.

8.2.2 Co-located Networks

In section 1.3, we emphasize the urgent need for cooperation between the co-located wireless networks since the rapid growth of the wireless devices that operate in the ISM band, is expected to lead to extensive interference. Motivated by the above finding, we intend to propose cooperation techniques between the co-located networks, which exploit the slow channel hopping technique (e.g., IEEE 802.15.4-TSCH) to cope with the external interference and multi-path fading effect.

A significant challenge is to propose techniques where channel hopping networks are able to detect the presence of other channel hopping networks. The techniques that monitor the reliability of the radio channels, such as blacklisting, are inefficient in this case. Since the exploitation of the channel hopping technique spreads the collisions in all radio channels uniformly. One possible solution would be to detect a pattern of collisions over time, i.e., the reliability of some cells is very different from the average reliability of other cells, as in [Cha+18]. In this case, the scheduling function re-schedules the cells to avoid collisions.

8.2.3 Improving the probing strategy of the *bad* channels

The exploitation of a software-based LQE has the drawback that the bad radio channels are no longer evaluated since they are on the blacklist. However, the radio conditions change over the time, and it is necessary to evaluate the bad radio channels since their quality maybe change. In our proposals with adaptive local per link blacklists (LABeL, Whitelist-Aware, and AMABO), we tackle the above issue by modifying the channel hopping sequence in such a way to use less frequently the bad radio channels in transmissions and therefore do not stop evaluating them. (cf. section 4.2.3). Although such an approach does not require to use additional control packets to probe the bad radio channels, data packets may have to be retransmitted more often when they are transmitted through bad channels (Fig. 5.7). The increase of the collisions is not negligible and, without a doubt, depends on the specific radio conditions of the deployment of the IIoT. In the future, we intend to focus our efforts on two directions.

In the first direction, we intend to adjust the frequency of probing of the bad radio channels according to their quality fluctuations [Li+15]. We also intend to go one step further by characterizing the source/cause of the interference (e.g., Wi-Fi, Bluetooth, multi-path fading effect, etc.) [Her+13]. Then, we can adjust the probing frequency accordingly (e.g., channels subjected to deep fade it is worthless to assess them continuously).

In the other direction, we would like to combine our link quality estimator based on PDR with the RSSI metric. The value of RSSI could be retrieved either from the received data packets or from the noise measurements during inactive periods of the network [Tav+15; Els+17]. Therefore, the assessment of bad radio channels could be based on the measurement of the RSSI, to eliminate collisions arising from the partial use of the bad radio channels.

8.2.4 sub-GHz band

In recent years there has been a lot of interest in developing communication technologies for IIoT that use the unlicensed sub-GHz band such as LoRa [All15] and SigFox. The communication technologies are intended to provide wireless connectivity to limited power devices over long distances, i.e., Low-Power Wide Area Network (LPWAN). As in the 2.4 GHz ISM band, the issue of interference is intense due to the concentration of a plethora of devices, in the same space that shares the wireless medium [Orf+17; Lau+17]. Thus, it would be interesting to adapt the blacklisting/whitelisting techniques that we developed to the specifics of the sub-GHz band to meet the requirement for high reliability of IIoT. More specifically, we could focus our research efforts on protocol LoRa/LoRaWAN, which has adopted the TSCH mechanism [Riz+17; Hau+20].

The adaption of our proposed blacklisting algorithms, designed and developed for TSCH networks, to LPWANs that operate in the sub-GHz band is not a straightforward process. Since the following challenges will have to be addressed:

- *What is the most effective technique for estimating the quality of the available channels?* The estimation of channels' quality may be a time-consuming process since sub-GHz wireless technologies might be using more radio channels.
- *What are the most effective blacklisting techniques, local or global [Šol+19]?* LPWANs cover a large geographical area, so there may be diversity in the quality of channels due to their spatial characteristics. Therefore, local blacklisting techniques seem more attractive.

Bibliography

[Acc+11] Nicola Accettura, Luigi Alfredo Grieco, Gennaro Boggia, and Pietro Camarda. "Performance analysis of the RPL routing protocol". In: *2011 IEEE International Conference on Mechatronics*. IEEE. 2011, pp. 767–772.

[Acc+13] Nicola Accettura, Maria Rita Palattella, Gennaro Boggia, Luigi Alfredo Grieco, and Mischa Dohler. "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things". In: *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2013, pp. 1–6.

[Acc+15] Nicola Accettura, Elvis Vogli, Maria Rita Palattella, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. "Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation". In: *IEEE Internet of Things Journal* 2.6 (2015), pp. 455–470.

[Adj+15] Cedric Adjih et al. "FIT IoT-LAB: A large scale open experimental IoT testbed". In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 459–464.

[AI+11] Mohammed H Almarshadi, Saleh M Ismail, et al. "Effects of precision irrigation on productivity and water use efficiency of alfalfa under different irrigation methods in arid climates". In: *Journal of Applied Sciences Research* 7.3 (2011), pp. 299–308.

[Ale+12] Roger Alexander et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. Mar. 2012.

[All12] ZigBee Alliance. *New ZigBee PRO feature: green power*. 2012.

[All15] LoRa Alliance. "White paper: A technical overview of LoRa and LoRaWAN". In: *The LoRa Alliance: San Ramon, CA, USA* (2015), pp. 7–11.

[An09] ISA An. "Standard Wireless Systems for Industrial Automation: Process Control and Related Applications". In: *ISA Standard* 100 (2009).

[Anc+13] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. "The role of the RPL routing protocol for smart grid communications". In: *IEEE Communications Magazine* 51.1 (2013), pp. 75–83.

[Aro+04] Anish Arora et al. "A line in the sand: a wireless sensor network for target detection, classification, and tracking". In: *Computer Networks* 46.5 (2004), pp. 605–634.

[Ass+12] IEEE Standards Association et al. "IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)—Amendment 1: MAC Sublayer; IEEE Std 802.15. 4e-2012 (Amendment to IEEE Std 802.15. 4-2011)". In: *IEEE Computer Society: New York, NY, USA* (2012).

[Atz+10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[Azm+14] Noraini Azmi, Latifah Munirah Kamarudin, Massudi Mahmuddin, Azizi Zakaria, AYM Shakaff, S Khatun, K Kamarudin, and MN Morshed. "Interference issues and mitigation method in WSN 2.4 GHz ISM band: A survey". In: *2014 2nd International Conference on Electronic Design (ICED)*. IEEE. 2014, pp. 403–408.

[Bab+15] Hamed Babaei, Jaber Karimpour, and Amin Hadidi. "A survey of approaches for university course timetabling problem". In: *Computers & Industrial Engineering* 86 (2015), pp. 43 –59.

[Bac+12] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco Antonio Zúñiga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. "Radio link quality estimation in wireless sensor networks: A survey". In: *ACM Transactions on Sensor Networks (TOSN)* 8.4 (2012), p. 34.

[Ban+18] Jyotirmoy Banik, Il Han Kim, Jianwei Zhou, Xiaolin Lu, and Andrea Fumagalli. "SmartHop: A Cloud-Driven Channel Hopping Algorithm for Improved IoT Network Connectivity and Stability". In: *2018 IEEE International Conference on Communications (ICC)*. IEEE. 2018, pp. 1–6.

[Ben+09] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. "Pearson correlation coefficient". In: *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[Blu10] SIG Bluetooth. *Specification of the Bluetooth System-Covered Core Package version: 4.0*. 2010.

[Boa+11] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer, and Marco Zúñiga. "Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation". In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE. 2011, pp. 175–186.

[BS15] Cory Beard and William Stallings. *Wireless communication networks and systems*. Pearson, 2015.

[Bur+04] Jenna Burrell, Tim Brooke, and Richard Beckwith. "Vineyard computing: Sensor networks in agricultural production". In: *IEEE Pervasive computing* 3.1 (2004), pp. 38–45.

[Cam+07] Alberto Camilli, Carlos E Cugnasca, Antonio M Saraiva, André R Hirakawa, and Pedro LP Corrêa. "From wireless sensors to field mapping: Anatomy of an application for precision agriculture". In: *Computers and Electronics in Agriculture* 58.1 (2007), pp. 25–36.

[CC05] Wensong Chu and Charles J Colbourn. "Optimal frequency-hopping sequences via cyclotomy". In: *IEEE Transactions on Information Theory* 51.3 (2005), pp. 1139–1141.

[Cer+05a] Alberto Cerpa, Jennifer L Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin. "Statistical model of lossy links in wireless sensor networks". In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE. 2005, pp. 81–88.

[Cer+05b] Alberto Cerpa, Jennifer L Wong, Miodrag Potkonjak, and Deborah Estrin. "Temporal properties of low power wireless links: modeling and implications on multi-hop routing". In: *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM. 2005, pp. 414–425.

[Cha+16] Tengfei Chang, Thomas Watteyne, Qin Wang, and Xavier Vilajosana. "LLSF: Low latency scheduling function for 6TiSCH networks". In: *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2016, pp. 93–95.

[Cha+18] Tengfei Chang, Thomas Watteyne, Xavier Vilajosana, and Qin Wang. "CCR: Cost-aware cell relocation in 6TiSCH networks". In: *Transactions on Emerging Telecommunications Technologies* 29.1 (2018). e3211 ett.3211, e3211.

[Cha+19] Tengfei Chang, Mališa Vučinić, Xavier Vilajosana, Simon Duquennoy, and Diego Dujovne. *6TiSCH Minimal Scheduling Function (MSF)*. Internet-Draft draft-ietf-6tisch-msf-06. Work in Progress. Internet Engineering Task Force, Aug. 2019. 19 pp.

[Che16] Michael Cheffena. "Propagation channel characteristics of industrial wireless sensor networks [wireless corner]". In: *IEEE Antennas and Propagation Magazine* 58.1 (2016), pp. 66–73.

[Chi+16] Francesco Chiti, Romano Fantacci, and Andrea Tani. "Performance evaluation of an adaptive channel allocation technique for cognitive wireless sensor networks". In: *IEEE Transactions on Vehicular Technology* 66.6 (2016), pp. 5351–5363.

[Dag+07] Serhan Dagtas, Yuri Natchetoi, and Huaigu Wu. "An integrated wireless sensing and mobile processing architecture for assisted living and healthcare applications". In: *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. 2007, pp. 70–72.

[Dan+18] Glenn Daneels, Bart Spinnewyn, Steven Latré, and Jeroen Famaey. "ReSF: Recurrent low-latency scheduling in IEEE 802.15. 4e TSCH networks". In: *Ad Hoc Networks* 69 (2018), pp. 100–114.

[Dan+19] Glenn Daneels, Steven Latré, and Jeroen Famaey. "Efficient Recurrent Low-Latency Scheduling in IEEE 802.15. 4e TSCH Networks". In: *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE. 2019, pp. 1–6.

[DD09] Isabel Dietrich and Falko Dressler. "On the lifetime of wireless sensor networks". In: *ACM Transactions on Sensor Networks (TOSN)* 5.1 (2009), pp. 1–39.

[Dec05] Jean-Dominique Decotignie. "Ethernet-based real-time and industrial communications". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1102–1117.

[DG+16] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. "IEEE 802.15. 4e: A survey". In: *Computer Communications* 88 (2016), pp. 1–24.

[DP+16] Marc Domingo-Prieto, Tengfei Chang, Xavier Vilajosana, and Thomas Watteyne. "Distributed pid-based scheduling for 6tisch networks". In: *IEEE Communications Letters* 20.5 (2016), pp. 1006–1009.

[DR12] Peng Du and George Roussos. "Adaptive time slotted channel hopping for wireless sensor networks". In: *2012 4th computer science and electronic engineering conference (CEEC)*. IEEE. 2012, pp. 29–34.

[DSP19] Carlos Alexandre Gouvea Da Silva and Carlos Marcelo Pedroso. "MAC-Layer Packet Loss Models for Wi-Fi Networks: A Survey". In: *IEEE Access* 7 (2019), pp. 180512–180531.

[Duq+15] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. "Orchestra: Robust mesh networks through autonomously scheduled TSCH". In: *Proceedings of the 13th ACM conference on embedded networked sensor systems*. ACM. 2015, pp. 337–350.

[Duq+17] Simon Duquennoy, Atis Elsts, Beshr Al Nahas, and George Oikonomo. "Tsch and 6tisch for contiki: Challenges, design and evaluation". In: *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2017, pp. 11–18.

[EKM11] Melike Erol-Kantarci and Hussein T Mouftah. "Wireless multimedia sensor and actor networks for the next generation power grid". In: *Ad Hoc Networks* 9.4 (2011), pp. 542–551.

[Els+17] Atis Elsts, Xenofon Fafoutis, Robert Piechocki, and Ian Craddock. "Adaptive channel selection in IEEE 802.15. 4 TSCH networks". In: *2017 Global Internet of Things Summit (GIoTS)*. IEEE. 2017, pp. 1–6.

[Els16] Atis Elsts. "Source-node selection to increase the reliability of sensor networks for building automation". In: (2016).

[FT00] Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*. Vol. 7. University of California, Irvine Irvine, 2000.

[Fun06] HART Comunication Fundation. "HART field communication protocol specification". In: *HFC_SPEC12, Revision* 6 (2006).

[Gai+16a] Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, and Fabrice Valois. "High-reliability scheduling in deterministic wireless multi-hop networks". In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2016, pp. 1–6.

[Gai+16b] Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, and Fabrice Valois. "Kausa: KPI-aware Scheduling Algorithm for Multi-flow in Multi-hop IoT Networks". In: *Ad-hoc, Mobile, and Wireless Networks (ADHOCNOW)*. Springer, 2016, pp. 47–61.

[Git79] John C Gittins. "Bandit processes and dynamic allocation indices". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 41.2 (1979), pp. 148–164.

[GK12] Olfa Gaddour and Anis Koubâa. "RPL in a nutshell: A survey". In: *Computer Networks* 56.14 (2012), pp. 3163–3178.

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[Gom+16] Pedro Henrique Gomes, Ying Chen, Thomas Watteyne, and Bhaskar Krishnamachari. "Insights into frequency diversity from measurements on an indoor low power wireless network testbed". In: *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2016, pp. 1–6.

[Gom+17] Pedro Henrique Gomes, Thomas Watteyne, and Bhaskar Krishnamachari. "MABO-TSCH: Multi-hop And Blacklist-based Optimized Time Synchronized Channel Hopping". In: *Transactions on Emerging Telecommunications Technologies* e3223 (2017), pp. 1–20.

[Gri+17] Simone Grimaldi, Aamir Mahmood, and Mikael Gidlund. "An SVM-based method for classification of external interference in industrial wireless sensor and actuator networks". In: *Journal of Sensor and Actuator Networks* 6.2 (2017), p. 9.

[Gun+17] Dolvara Gunatilaka, Mo Sha, and Chenyang Lu. "Impacts of channel selection on industrial wireless sensor-actuator networks". In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[Ham+] Sarra Hammoudi, Zibouda Aliouat, and Saad Harous. "Enhanced time-slotted channel hopping". In: *Transactions on Emerging Telecommunications Technologies* (), e3638.

[Hau+20] Martin Haubro, Charalampos Orfanidis, George Oikonomou, and Xenofon Fafoutis. "TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things". In: *Internet Technology Letters* (2020).

[Her+13] Frederik Hermans, Olof Rensfelt, Thiemo Voigt, Edith Ngai, Lars-Åke Norden, and Per Gunningberg. "SoNIC: classifying interference in 802.15. 4 sensor networks". In: *Proceedings of the 12th international conference on Information processing in sensor networks*. 2013, pp. 55–66.

[Her+17] Rodrigo Teles Hermeto, Antoine Gallais, and Fabrice Theoleyre. "Scheduling for IEEE802. 15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey". In: *Computer Communications* 114 (2017), pp. 84–105.

[Hit+14] Anwar Hithnawi, Hossein Shafagh, and Simon Duquennoy. "Understanding the Impact of Cross Technology Interference on IEEE 802.15.4". In: *International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*. ACM. Maui, Hawaii, USA, 2014, pp. 49–56.

[Hos+16] Inès Hosni, Fabrice Théoleyre, and Noureddine Hamdi. "Localized scheduling for end-to-end delay constrained low power lossy networks with 6tisch". In: *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE. 2016, pp. 507–512.

[HT17] Inès Hosni and Fabrice Théoleyre. "Self-healing distributed scheduling for end-to-end delay optimization in multihop wireless networks with 6TiSCh". In: *Computer Communications* 110 (2017), pp. 103 –119.

[Hwa+17] Ren-Hung Hwang, Chih-Chiang Wang, and Wu-Bin Wang. "A distributed scheduling algorithm for ieee 802.15. 4e wireless sensor networks". In: *Computer Standards & Interfaces* 52 (2017), pp. 63–70.

[Hän+11] Markku Hänninen, Jukka Suhonen, Timo D Hämäläinen, and Marko Hännikäinen. "Link Quality-Based Channel Selection for Resource Constrained WSNs". In: Springer. 2011.

[Ieea] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)". In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016).

[Ieeb] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN)". In: *IEEE Std 802.15.4-2003* (2003), pp. 1–680.

[Itu] *ITU, internet reports 2005: The internet of things*. tech. report. ITU, 2005.

[JC17] Kihoon Jeon and Sanghwa Chung. "Adaptive channel quality estimation method for enhanced time slotted channel hopping on wireless sensor networks". In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE. 2017, pp. 438–443.

[Jin18] Vandana Jindal. "History and architecture of wireless sensor networks for ubiquitous computing". In: *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 7.2 (2018), pp. 214–217.

[Joh60] Selmer Martin Johnson. "A linear diophantine problem". In: *Canadian Journal of Mathematics* 12.3 (1960), pp. 390–398.

[Jon+10] Val Jones, Valerie Gay, and Peter Leijdekkers. "Body sensor networks for mobile health monitoring: Experience in europe and australia". In: *2010 Fourth International Conference on Digital Society*. IEEE. 2010, pp. 204–209.

[KL09] Albert Ko and Henry YK Lau. "Robot assisted emergency search and rescue system with a wireless sensor network". In: *International Journal of Advanced Science and Technology* 3 (2009), pp. 69–78.

[Kot+17] Vasileios Kotsiou, Georgios Z. Papadopoulos, Periklis Chatzimisios, and Fabrice Theoleyre. "LABeL: Link-based Adaptive BLacklisting Technique for 6TiSCH Wireless Industrial Networks". In: *Proc. of the 20th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM. 2017.

[Kov+11] Matthias Kovatsch, Simon Duquennoy, and Adam Dunkels. "A low-power CoAP for Contiki". In: *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE. 2011, pp. 855–860.

[Kru+19] Leo Krueger, Lotte Steenbrink, and Andreas Timm-Giel. "Avoiding Local Interference in IEEE 802.15. 4 TSCH Networks using a Scheduling Function with Distributed Blacklists". In: *Mobile Communication-Technologies and Applications; 24. ITG-Symposium*. VDE. 2019, pp. 1–6.

[Kus+07] Nandakishore Kushalnagar, Gabriel Montenegro, Christian Schumacher, et al. "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals". In: (2007).

[Lan+06] Koen Langendoen, Aline Baggio, and Otto Visser. "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture". In: *Proceedings 20th IEEE international parallel & distributed processing symposium*. IEEE. 2006, 8–pp.

[Lau+17] Mads Lauridsen, Benny Vejlgaard, István Z Kovács, Huan Nguyen, and Preben Mogensen. "Interference measurements in the European 868 MHz ISM band with focus on LoRa and SigFox". In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2017, pp. 1–6.

[Le+09] Hanh-Phuc Le, Mervin John, and Kris Pister. "Energy-aware routing in wireless sensor networks with adaptive energy-slope control". In: *EE290Q-2 Spring* (2009).

[Li+15] Peishuo Li, Tom Vermeulen, Hong Liy, and Sofie Pollin. "An adaptive channel selection scheme for reliable TSCH-based communication". In: *2015 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE. 2015, pp. 511–515.

[Lim+10] Yujin Lim, Hak-Man Kim, and Sanggil Kang. "A design of wireless sensor networks for a power quality monitoring system". In: *Sensors* 10.11 (2010), pp. 9712–9725.

[Lin+04] Ruizhong Lin, Zhi Wang, and Youxian Sun. "Wireless sensor networks solutions for real time monitoring of nuclear power plant". In: *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*. Vol. 4. IEEE. 2004, pp. 3663–3667.

[Liu+12] Yunhao Liu, Yuan He, Mo Li, Jiliang Wang, Kebin Liu, and Xiangyang Li. "Does wireless sensor network scale? A measurement study on GreenOrbs". In: *IEEE Transactions on Parallel and Distributed Systems* 24.10 (2012), pp. 1983–1993.

[Lon+17] Stefano Longo, Tingli Su, Guido Herrmann, and Phil Barber. *Optimal and robust scheduling for networked control systems*. CRC press, 2017.

[Mun+19] Esteban Municio et al. "Simulating 6TiSCH networks". In: *Transactions on Emerging Telecommunications Technologies* 30.3 (2019), e3494.

[Nan+12] Anirudha Nanda, Manisha P Pai, and Abhijeet Gole. "An algorithm to automatically generate schedule for school lectures using a heuristic approach". In: *International Journal of Machine Learning and Computing* 2 (2012).

[Nat+16] Radhakrishnan Natarajan, Pouria Zand, and Majid Nabi. "Analysis of coexistence between IEEE 802.15. 4, BLE and IEEE 802.11 in the 2.4 GHz ISM band". In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 6025–6032.

[NK06] Mikael M Nordman and Taneli Korhonen. "Design of a concept and a wireless ASIC sensor for locating earth faults in unearthed electrical distribution networks". In: *IEEE transactions on power delivery* 21.3 (2006), pp. 1074–1082.

[Oh+18] Sukho Oh, DongYeop Hwang, Ki-Hyung Kim, and Kangseok Kim. "Escalator: An Autonomous Scheduling Scheme for Convergecast in TSCH". In: *Sensors* 18.4 (2018), p. 1209.

[Ols14] Jonas Olsson. "6LoWPAN demystified". In: *Texas Instruments* 13 (2014).

[OP99] Bob O'Hara and Al Petrick. *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network IEEE Press, 1999.

[Orf+17] Charalampos Orfanidis, Laura Marie Feeney, Martin Jacobsson, and Per Gunningberg. "Investigating interference between LoRa and IEEE 802.15. 4g networks". In: *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2017, pp. 1–8.

[Pal+12] Maria Rita Palattella, Nicola Accettura, Mischa Dohler, Luigi Alfredo Grieco, and Gennaro Boggia. "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks". In: *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*. IEEE. 2012, pp. 327–332.

[Pal+13] Maria Rita Palattella, Nicola Accettura, Luigi Alfredo Grieco, Gennaro Boggia, Mischa Dohler, and Thomas Engel. "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802. 15.4 e TSCH". In: *IEEE Sensors Journal* 13.10 (2013), pp. 3655–3666.

[Pal+14] Maria Rita Palattella, Pascal Thubert, Xavier Vilajosana, Thomas Watteyne, Qin Wang, and Thomas Engel. "6tisch wireless industrial networks: Determinism meets ipv6". In: *Internet of Things*. Springer, 2014, pp. 111–141.

[Pal+16] Maria Rita Palattella, Mischa Dohler, Alfredo Grieco, Gianluca Rizzo, Johan Torsner, Thomas Engel, and Latif Ladid. "Internet of things in the 5G era: Enablers, architecture, and business models". In: *IEEE Journal on Selected Areas in Communications* 34.3 (2016), pp. 510–527.

[Pap+] Georgios Z Papadopoulos, Fabrice Theoleyre, and Pascal Thubert. "Operations, Administration and Maintenance (OAM) features for Reliable and Available Wireless (RAW) Networks". In: *Internet Technology Letters* ().

[Pap+16a] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noel. "Importance of Repeatable Setups for Reproducible Experimental Results in IoT". In: *PE-WASUN*. ACM. 2016.

[Pap+16b] Georgios Z Papadopoulos, Antoine Gallais, Guillaume Schreiner, and Thomas Noël. "Importance of repeatable setups for reproducible experimental results in IoT". In: *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. 2016, pp. 51–59.

[Pap+17] Georgios Z Papadopoulos, Antoine Gallais, Guillaume Schreiner, Emery Jou, and Thomas Noel. "Thorough IoT testbed characterization: From proof-of-concept to repeatable experimentations". In: *Computer Networks* 119 (2017), pp. 86–101.

[PC11] Stig Petersen and Simon Carlsen. "WirelessHART Versus ISA100. 11a: The Format War Hits the Factory Floor". In: *IEEE Industrial Electronics Magazine* 5.4 (2011), pp. 23–34.

[PD08] Kris Pister and Lance Doherty. "TSMP: Time Synchronized Mesh Protocol". In: *Parallel and Distributed Computing and Systems*. 2008.

[PH06] Daniele Puccinelli and Martin Haenggi. "Multipath fading in wireless sensor networks: measurements and interpretation". In: *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. ACM. 2006, pp. 1039–1044.

[Que+18] Diego V Queiroz, Ruan D Gomes, Cesar Benavente-Peces, Iguatemi E Fonseca, and Marcelo S Alencar. "Evaluation of Channels Blacklists in TSCH Networks with Star and Tree Topologies". In: *Proceedings of the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*. ACM. 2018, pp. 116–123.

[Raz+19] Saleem Raza, Muhammad Faheem, and Mesut Guenes. "Industrial wireless sensor and actuator networks in industry 4.0: Exploring requirements, protocols, and challenges—A MAC survey". In: *International Journal of Communication Systems* 32.15 (2019), e4074.

[Riz+17] Mattia Rizzi, Paolo Ferrari, Alessandra Flammini, Emiliano Sisinni, and Mikael Gidlund. "Using LoRa for industrial wireless networks". In: *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*. IEEE. 2017, pp. 1–4.

[San+10] Lifeng Sang, Anish Arora, and Hongwei Zhang. "On link asymmetry and one-way estimation in wireless sensor networks". In: *ACM Transactions on Sensor Networks (TOSN)* 6.2 (2010), p. 12.

[Sef+20] Amina Seferagić, Jeroen Famaey, Eli De Poorter, and Jeroen Hoebeke. "Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things". In: *Sensors* 20.2 (2020), p. 488.

[Sha+06] Kewei Sha, Weisong Shi, and Orlando Watkins. "Using wireless sensor networks for fire rescue applications: Requirements and challenges". In: *2006 IEEE International Conference on Electro/Information Technology*. IEEE. 2006, pp. 239–244.

[Sha+11] Mo Sha, Gregory Hackmann, and Chenyang Lu. "ARCH: Practical channel hopping for reliable home-area sensor networks". In: *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE. 2011, pp. 305–315.

[She+14] Zach Shelby, Klaus Hartke, and Carsten Bormann. "The constrained application protocol (CoAP)". In: (2014).

[Shi+15] Chao-Fang Shih, Ariton E Xhafa, and Jianwei Zhou. "Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks". In: *Proc. of the IEEE International Conference on Communications (ICC)*. 2015.

[Sis+18] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. "Industrial internet of things: Challenges, opportunities, and directions". In: *IEEE Transactions on Industrial Informatics* 14.11 (2018), pp. 4724–4734.

[Son+08] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control". In: *RTAS*. IEEE. 2008.

[Sou+16] Ridha Soua, Pascale Minet, and Erwan Livolant. "Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15. 4e TSCH networks". In: *Transactions on Emerging Telecommunications Technologies* 27.4 (2016), pp. 557–575.

[Spe08] WirelessHART Specification. "75: TDMA Data-Link Layer". In: *HART Communication Foundation Std., Rev* 1 (2008).

[Sri+10] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. "An empirical study of low-power wireless". In: *ACM Transactions on Sensor Networks (TOSN)* 6.2 (2010), p. 16.

[Sta+13] David Stanislowski, Xavier Vilajosana, Qin Wang, Thomas Watteyne, and Kristofer SJ Pister. "Adaptive synchronization in IEEE802. 15.4 e networks". In: *IEEE Transactions on Industrial Informatics* 10.1 (2013), pp. 795–802.

[Sur+12] NK Suryadevara, Anuroop Gaddam, RK Rayudu, and SC Mukhopadhyay. "Wireless sensors network based safe home to care elderly people: Behaviour detection". In: *Sensors and Actuators A: Physical* 186 (2012), pp. 277–283.

[Tan+06] Vanessa WS Tang, Yuan Zheng, and Jiannong Cao. "An intelligent car park management system based on wireless sensor networks". In: *2006 First International Symposium on Pervasive Computing and Applications*. IEEE. 2006, pp. 65–70.

[Tav+15] Rasool Tavakoli, Majid Nabi, Twan Basten, and Kees Goossens. "Enhanced Time-Slotted Channel Hopping in WSNs Using Non-intrusive Channel-Quality Estimation". In: *Proc. of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2015.

[Tav+18] Rasool Tavakoli, Majid Nabi, Twan Basten, and Kees Goossens. "Dependable interference-aware time-slotted channel hopping for wireless sensor networks". In: *ACM Transactions on Sensor Networks (TOSN)* 14.1 (2018), p. 3.

[Tho05] J-P Thomesse. "Fieldbus technology in industrial automation". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1073–1101.

[TP16] Fabrice Theoleyre and Georgios Z Papadopoulos. "Experimental validation of a distributed self-configured 6TiSCH with traffic isolation in low power lossy networks". In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM. 2016, pp. 102–110.

[Vil+14] Xavier Vilajosana, Qin Wang, Fabien Chraim, Thomas Watteyne, Tengfei Chang, and Kristofer SJ Pister. "A Realistic Energy Consumption Model for TSCH Networks". In: *IEEE Sensors Journal* 14.2 (2014), pp. 482–489.

[WA+06] Geoffrey Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees, and Matt Welsh. "Deploying a wireless sensor network on an active volcano". In: *IEEE internet computing* 10.2 (2006), pp. 18–25.

[WA+08] Geoffrey Werner-Allen, Stephen Dawson-Haggerty, and Matt Welsh. "Lance: Optimizing High-Resolution Signal Collection in Wireless Sensor Networks". In: ACM, 2008.

[Wan+16] Jiafu Wan, Shenglong Tang, Zhaogang Shu, Di Li, Shiyong Wang, Muhammad Imran, and Athanasios V Vasilakos. "Software-defined industrial internet of things in the context of industry 4.0". In: *IEEE Sensors Journal* 16.20 (2016), pp. 7373–7380.

[Wan+18] Qin Wang, Xavier Vilajosana, and Thomas Watteyne. *6TiSCH Operation Sublayer (6top) Protocol (6P)*. RFC 8480. Nov. 2018.

[Wat+09] Thomas Watteyne, Ankur Mehta, and Kris Pister. "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense". In: *PE-WASUN*. ACM. 2009.

[Wat+10] Thomas Watteyne, Steven Lanzisera, Ankur Mehta, and Kristofer SJ Pister. "Mitigating multipath fading through channel hopping in wireless sensor networks". In: *2010 IEEE International Conference on Communications*. IEEE. 2010, pp. 1–5.

[Wat+12] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven Glaser, and Kris Pister. "OpenWSN: a standards-based low-power wireless development environment". In: *Transactions on Emerging Telecommunications Technologies* 23.5 (2012), pp. 480–493.

[Wat+15a] Thomas Watteyne, Cédric Adjih, and Xavier Vilajosana. "Lessons Learned from Large-scale Dense IEEE802.15.4 Connectivity Traces". In: *CASE*. IEEE. 2015.

[Wat+15b] Thomas Watteyne, Maria-Rita Palattella, and Luigi Alfredo Grieco. *Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement*. RFC 7554. 2015.

[WC03] Alec Woo and David E Culler. *Evaluation of efficient link reliability estimators for low-power wireless networks*. Computer Science Division, University of California Oakland, Calif, USA, 2003.

[Wei91] Mark Weiser. "The Computer for the 21 st Century". In: *Scientific american* 265.3 (1991), pp. 94–105.

[Win+12] Tim Winter et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. rfc 6550. IETF, 2012.

[WJ16] Quan Wang and Jin Jiang. "Comparative examination on architecture and protocol of industrial wireless sensor network standards". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2197–2219.

[Wol+17] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0". In: *IEEE industrial electronics magazine* 11.1 (2017), pp. 17–27.

[WP67] Dominic JA Welsh and Martin B Powell. "An upper bound for the chromatic number of a graph and its application to timetabling problems". In: *The Computer Journal* 10.1 (1967), pp. 85–86.

[Wu+08] Yafeng Wu, John A Stankovic, Tian He, and Shan Lin. "Realistic and efficient multi-channel communications in wireless sensor networks". In: *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE. 2008, pp. 1193–1201.

[Xu+04] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. "A wireless sensor network for structural monitoring". In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004, pp. 13–24.

[Yaa+16] Sahar Ben Yaala, Fabrice Théoleyre, and Ridha Bouallegue. "Performance study of co-located IEEE 802.15. 4-TSCH networks: Interference and coexistence". In: *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE. 2016, pp. 513–518.

[Yaa+17] Sahar Ben Yaala, Fabrice Theoleyre, and Ridha Bouallegue. "Cooperative Resynchronization to Improve the Reliability of Colocated IEEE 802.15.4-TSCH Networks in Dense Deployments". In: *Ad Hoc Networks* 64 (2017), pp. 112 –126.

[Yan+15] Jiachen Yang, Jianxiong Zhou, Zhihan Lv, Wei Wei, and Houbing Song. "A real-time monitoring system of industry carbon monoxide based on wireless sensor networks". In: *Sensors* 15.11 (2015), pp. 29535–29546.

[Zha+04] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi. "Hardware Design Experiences in ZebraNet". In: *SenSys*. ACM, 2004.

[Ziv16] Natasa Zivic. *Modern Communications Technology*. Walter de Gruyter GmbH & Co KG, 2016.

[Zor+18a] Dimitrios Zorbas, Georgios Z. Papadopoulos, and Christos Douligeris. "Local or Global Radio Channel Blacklisting for IEEE 802.15.4-TSCH Networks?" In: *Proc. of the IEEE International Conference on Communications (ICC)*. 2018.

[Zor+18b] Dimitrios Zorbas, Vassilis Kotsiou, Fabrice Théoleyre, Georgios Z Papadopoulos, and Christos Douligeris. "LOST: Localized Blacklisting Aware Scheduling Algorithm for IEEE 802.15.4-TSCH Networks". In: *Proc. of the 10th Wireless Days (WD)*. 2018.

[Cis] Cisco. *The Internet of Things Infographic*. `https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/iot-aag.pdf`.

[ISAy] ISA-100.11a-2011: "Wireless Systems for Industrial Automation:Process Control and Related Applications". In: *International Society of Automation (ISA) Std.* 1 (May 2011).

[Šol+19] Tomaž Šolc, Halil Yetgin, Timotej Gale, Mihael Mohorčič, and Carolina Fortuna. "Whitelisting in RFDMA Networks". In: *IEEE Access* 7 (2019), pp. 159284–159299.

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	Low-Power Wireless Personal Area Networks.
6P	6top Protocol.
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e.
6top	6TSCH Operation Sublayer.
ADV	Advertisement.
AP	Access Point.
ARQ	Automatic Repeat reQuest.
ARR	Acknowledgment Reception Ratio.
ASN	Absolute Sequence Number.
BER	Bit Error Rate.
BLE	Bluetooth Low Energy.
CBR	Constant Bit Rate.
CCA	Clear Channel Assessment.
CDF	Cumulative Distribution Function.
CHS	Channel Hopping Sequence.
CoAP	Constrained Application Protocol.
CSMA	Carrier Sense Multiple Access.
DAO	Destination Advertisement Object.
DIO	DODAG Information Object.
DODAG	Destination-Oriented Directed Acyclic Graph.
DSN	Data Sequence Numbers.
EB	Enhanced Beacon.
ED	Energy Detection.
ETF	Expected number of Transmissions over Forwarding radio links.
ETX	Expected Transmission Count.
EWMA	Exponential Weighted Moving Average.
FHS	Frequency Hopping Sequence.
ICMP	Internet Control Message Protocol.
IE	Information Element.
IETF	Internet Engineering Task Force.
IIoT	Industrial Internet of Things.

IoT	Internet of Things.
IP	Internet Protocol.
IPv6	Internet Protocol version 6.
ISA	International Society of Automation.
ISM	Industrial, Scientific and Medical radio bands.
LLN	Low-power and Lossy Network.
LPWAN	Low-Power Wide Area Network.
LQE	Link Quality Estimator.
LQI	Link Quality Indicator.
LR-WPAN	Low-Rate Wireless Personal Area Network.
MAC	Medium Access Control.
MTU	Maximum Transmission Unit.
NF	Noise Floor.
PAN	Personal Area Network.
PCE	Path Computation Element.
PDR	Packet Delivery Ratio.
PER	Packet Error Rate.
PHY	Physical.
PRR	Packet Reception Rate.
REST	Representational State Transfer.
RNP	Required Number of Packets.
RPL	IPv6 Routing Protocol for Low-power and Lossy Networks.
RPT	additional auxiliary RePorT.
RR	Reception Rate.
RSSI	Received Signal Strength Indicator.
SES	Simple Exponential Smoothing.
SF	Scheduling Function.
SINR	Signal to Interference plus Noise Ratio.
SNR	Signal-to-Noise Ratio.
TCP	Transmission Control Protocol.
TDMA	Time Division Multiple Access.
TSCH	Timeslotted Channel Hopping.
TSMP	Time Synchronized Mesh Protocol.
UCTP	University Course Timetabling Problem.
UDP	User Datagram Protocol.
WMEWMA	Window Mean Exponentially-Weighted Moving Average.
WSAN	Wireless Sensor and Actuator Network.
WSN	Wireless Sensor Network.

Communications fiables pour l'Internet des Objets Industriels

Résumé

l'asbtract en français

Index Terms

Internet des Objets Industriels ; fiabilité ; interférences externes ; ordonnancement

I. Introduction

Les réseaux de capteurs représentent une technologie révolutionnaire ayant émergé durant la guerre froide [Jin18]. Dans les années 1950, l'armée américaine a déployé le système de surveillance sonore (SOSUS) qui détecter et traque les sous-marins silencieux soviétiques en utilisant des capteurs sonores. De nos jours, les réseaux de capteurs sans-fil (WSN) sont devenus une technoologie fondamentale vers la dématérialisation suivant la vision de Mark Weiser pour l'informatique ubiquitaire [Wei91].

Un ensemble de capteurs forment un réseau appelé WSN. Un noeud sans-fil est un noeud contraint, ce qui crée des limites dans la puissance de transmission, la mémoire, les capacités de calcul et l'énergie. De plus, les WSN sont déployés de façon dense, et dans de nombreux, requièrent des communications multisaut, exacerbant les problèmes créés par des liens radio non fiables, sujets aux interférences externes. Ainsi, les WSN peuvent être considérés comme des réseaux basse-puissance, avec pertes (LLNs).

Dans une application typique pour les WSN, les noeuds mesurent leur environnement naturel et encapsulent leurs mesures dans des paquets de données. Ces paquets de données sont reçus par d'autres noeuds WSN qui les relaient souvent vers un puits, en charge de collecter les paquets de données. Ainsi, un noeud peut générer ou relayer ou traiter des paquets. Il les relaie aussi de et vers le puits, en agrégeant possiblement plusieurs paquets afin de réduire le volume de données à transférer.

Nous identifions trois grands types de trafic dans les WSN

— *multipoint-to-point ou convergecast* : les noeuds envoient leurs paquets vers un petit nombre de puits ;
— *point-to-multipoint* : le puits envoie des données (ex : commandes, mises à jour logicielles, informations pour rejoindre le réseau) aux noeuds ;
— *point-to-point* : un noeud envoie directement ces données à un autre noeud (ex : boucle de contrôle).

Le noeud source, selon l'application qu'il exécute, envoie les données au puits : i) périodiquement (Constant Bit Rate, à intervalles de temps constants), ii) sporadiquement après qu'un évènement a été détecté (Event-triggered, Event-driven), iii) réactivement, pour répondre à une requête reçue du puits ou d'un autre noeud (Query-driven).

L'évolution des WSN de telle sorte qu'ils soient connectés à Internet a permis d'étendre leur intérêt, en créant l'Internet des Objets (IoT). Les WSN sont devenus une technologie clé pour l'IoT, servant d'interface entre le monde physique et numérique, et augmentant les capacités de l'environnement.

A. Internet des Objets

Dans la dernière décennie, nous avons connu la pénétration très rapide de l'Internet dans la vie de tous les jours de biens des humains. Nous avons radicalement changé la façon de laquelle les gens travaillent, s'amusent, communiquent, s'informent, s'instruisent. L'Internet a pour but de connecter des ordinateurs, des supercalulateurs, des tablettes, et maintenant des smartphones. Récemment, un nouveau concept de communication, appelé *Internet des Objets*, est devenu clé. L'IoT se réfère à un ensemble d'objets identifiables de façon unique, connectés en un réseau d'objets [AIM10]. Les objets intelligents sont

des dispositifs embarqués, avec des capacités de mesure, de traitement, et de communications. L'expansion de l'infrastructure et des services dans cet Internet, dans lequel des objets hétérogènes collaborent et communiquent, sert la vision "tout le temp, partout, avec n'importe quoi" [Itu].

L'expansion de l'IoT est rapide, les prévisions de 2020 prévoyaient 20 milliards d'objets connectés [Cis] tandis que les prévisions de 2025 en prévoient 100 milliars.

Un nombre croissant d'applications reposent sur l'IoT, avec par exemple :

— *Applications environnementales* : suivi de populations animales [Zha+04], mesure d'activités sismiques [WADHW08], mesure de l'humidité et de la température [LBV06], msue de l'activité volcanique [WA+06], mesure de la santé d'un batiment [Xu+04] ;
— *Santé :* suivi de santé de patients [JGL10 ; DNW07] avec des body area networks, aide aux personnes âgées dans leur vie quotidienne [Sur+12] ;
— *Applications urbaines* : contrôle du trafic routier [Aro+04], parking intelligent [TZC06] ;
— *Grille intelligente* : gestion du système électrique [LKK10], surveillance de fermes éoliennes [EKM11], gestion automatique des pannes électriques [NK06] ;
— *Applications industrielles* : agrilctulture de précision [AI+11], surveillance d'une centrale nucléaire [LWS04], surveillance du monoxyde de carbone [Yan+15]
— *Applications terrestres* : mesure de température, de l'humidité , du pH de sol, de pesticides et insecticides dans lenvironnement [BBB04 ; Cam+07] ;
— *Urgences* : détection de feu de forêt, opération de secours en cas de feu [SSW06] ou de tremblement de terre [KL09].

Les technologies sans-fil préexistantes, telles que le standard très populaire IEEE 802.11 [OP99], sont considérées comme inappropriées dans le cas des réseaux de capteurs sans-fil. En effet, de petits objets embarqués ne peuvent exécuter un protocole aussi couteux en énergie. Ainsi, des technologies radio taillées sur mesure pour les WSN ont été proposées. ZigBee [All12] et le standard IEEE Std 802.15.4 [Ieeb] sont en particulier très populaires. Le standard IEEE Std 802.15.4 définit l'Accès au Medium (MAC) et la couche Physique pour les réseau personnels sans-fils à bas-débit (LR-WPANs). Le standard IEEE Std 802.15.4 se distingue par sa faible puissance de transmission pour économiser l'énergie, un débit d'au plus 250 kbit/s, L'usage de bande ISM (Industrial Scientific Medical, $2.4\ GHz$) et une taille maximale de paquets (MTU) de 127 octets

L'adoption d'IPv6 comme protocole d'accès à Internet est sans conteste une solution obligatoire pour l'IoT, étant donné le nombre colossal d'équipements à connecter. Cependant, il reste des défis sciginficatifs, tels que la différence de MTU entre IPv6 et IEEE 802.15.4. L'Internet Engineering Task Force (IETF) a standardisé des protocoles conçus pour les LLNs et donc également les WSN. Ainsi, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) est en charge du transport des larges paquets IPv6 à traver les couches basses de IEEE 802.15.4, convertissant les adresses, et compressant les en-têtes. Comme un réseau WSN est bien souvent multisaut, un protocole de routage est également souvent utilisé, afin que chaque noeud possède un prochain saut dans sa table de routage pour toutes les destinations du réseau. Ainsi, IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) est le protocole de routage standardisé actuellement conçu pour ces WSN, utilisant en priorité un trafic de type convergecast.

B. L'internet des Objets Industriels

L'utilisation des technologies de l'IoT aux processus industriels représente un élément clé de la transition vers a $4^{ième}$ révolution industrielle (Industry 4.0) [WSJ17]. Par le terme de Industry 4.0, nous désignons la fusion entre les paradigmes de l'IoT et des Syctèmes Cyber Physiques (CPS) [Sis+18]. L'Internet des Objets Industriels (IIoT) comprend un large ensemble de capteurs et actionneurs sans-fil, utilisées par des applications industrielles. Typiquement, les réseaux de contrôle comprennent des capteurs, actionneurs et des contrôleurs [Lon+17]. Les capteurs envoient leurs mesures régulièrement au contrôleur, qui, grâce à ces retours, peut déclencher des actionneurs afin de corriger le comportement su l'environnement. Cette boucle de contrôle doit souvent fonctionner en temps-réel.

Cependant, les communications pour un grand nombre de systèmes industriels (sécurité, supervision, alerte, collecte d'informations) doivent respecter des critères précis. Ainsi, l'infrastructure réseau doit pouvoir fournir une fiabilité élevée avec un délai de bout en bout borné. Malheureusement, ces contraintes sont complexes à respecter dans des environnements radio tels que l'IIoT.

Les réseaux industriels câblés respectent déjà de telles contraintes, fournissant débits élevés, sur de longues distances. Depuis quelques années, l'industrie a déployé de nombreux réseaux reposant par exemple sur Highway Addressable Remote Transducer (HART) [Fun06], FieldBus [Tho05], ou Ethernet temps-réel (RTE) [Dec05]. Cependant, ces infrastructures filaires demandent un cout élevé en cables, et une maintenance chère. L'installation de cables dans des environnements dangereux (flames, explosifs, chaleur) est complexe, et même impossible pour certains noeuds mobiles (robots, véhicules autonomes) [Sef+20].

Du fait de ces limites, les réseaux sans-fil de capteurs et actionneurs (WSAN) font partie intégrante de l'IIoT, qui commence à remplacer de façon graduelle les réseaux industriels filaires. Les bénéfices d'une telle adoption sont nombreux :

— *Reconfiguration flexible de la chaine de production* : par exemple, des robots industriels peuvent avec bénéfice utiliser des communications radio, les cables étant sujets à cassure après des milliers de flexions [PTT]. La topologie réseau peut également être changé facilement, permettant une reconfiguration rapide des outils de production [Wan+16].
— *Réduction des couts* : le déploiement sans-fil est souvent moins cher, plus facile à maintenant, facile à réparer par rapport aux réseaux filaires.
— *Sécurité* : L'exploitation de l'IIoT peut améliorer la sécurité des travailleurs. Par exemple, le remplacement de capteurs sans-fil dans des environnements dangereux peut être réalisée sans intervention humaine.

Dans un déploiement IIoT classique, les paquets de données sont envoyés à travers le réseau à une destination, tout en respectant des contraintes de temps strictes. Formulé autrement, nous avons besoin de communications temps-réel. La couche MAC est la couche principale de la pile en charge de respecter ces contraintes de temps, gérant la contention entre tous les équipements.

La couche MAC doit coordonner les transmissions dans un médium partagé, spécifiant comment et quand un noeud doit essayer de transmettre un paquet. L'objectif principale de la couche MAC dans un réseau IIoT est d'économiser de l'énergie, en coupant l'alimentation du noeud quand elle n'est pas nécessaire. La couche MAC est également responsable de plusieurs mécanismes gâchant l'énergie, tels que l'overhearing, les collisions, l'écoute active, et la surdité. Par ailleurs, la couche MAC doit aussi assurer une livraison déterministe et respectant un certain nombre de contraintes en orchestrant les transmissions dans le réseau. Les techniques MAC basées sur un accès aléatoire telles que slotted Aloha et Carrier Sense Multiple Access (CSMA), sont inappropriées pour les contraintes des réseaux industriels. Cette classe de protocoles MAC doit présenter des performances strictes en termes de fiabilité, latence, même en cas de fort trafic.

Les protocoles MAC (synchronisés) basés sur l'ordonnancement des transmissions représentent les meilleurs candidats pour respecter de telles contraintes. Le standard IEEE 802.15.4-2015 [Ieea] orienté dans cette direction a proposé en 2016 le mode Time-Slotted Channel Hopping (TSCH) pour le standard IEEE 802.15.4. TSCh est basé sur les protocoles déjà existant Time Synchronized Mesh Protocol(TSMP) [PD08], WirelessHART [Spe08], et ISA100.11a [ISAy]. TSCH exploite un découpage en temps de l'accès (TDMA) pour éviter les collisions entre émetteurs, et limiter l'impact des interférences externes et du multipath fading grâce à une technique de saut de fréquence lente.

Les protocoles MAC IIoT qui ont été proposés jusque là, exploitent l'IIoT principalement pour les systèmes industriels non critiques. Pour étendre la palette d'applications, et afin de matérialiser l'Industrie 4.0, les défis suivants ont besoin d'être résolus [RFG19] :

— **Fiabilité** : le médium radio est, de par sa nature, partagé, non fiable, et de qualité fluctuante dans le temps. Les pertes sont exacerbées par l'environnement industriel qui contient de nombreuses surfaces métalliques réfléchissantes, le bruit électromagnétique des machines, des interférences radio. Une solution évidente consiste a gérer ce manque de fiabilité en augmentant le nombre de retransmissions

au niveau MAC. Cependant, cette technique augmente à la fois la latence et la consommation d'énergie, tout en réduisant la bande passante disponible.

— **Performances temps-réel** : la couche MAC doit respecter des contraintes en termes de latence par exemple. Des délais maximum de 10 ms pour les applications critiques et de 100 ms pour les applications de supervisions sont courants [WJ16]. Résoudre ces défis est d'autant plus complexe que la qualité des liens radio peut se dégrader soudainement, causant des fluctuations en termes de délai.
— **Efficacité énergétique** : les noeuds doivent pouvoir économiser l'énergie qu'ils consomment, d'autant plus que le remplacement de leurs piles est quelquefois même impossible. Cependant, la demande de fiabilité et de performances temps-réel augmentent leur consommation d'énergie, rendant nécessaire la recherche d'un compromis entre ces objectifs contradictoires.
— **Passage à l'échelle** : nous devons pouvoir adapter la topologie réseau, ajouter ou supprimer des noeuds, sans avoir besoin de changer le protocoles MAC, ni de le reconfigurer. De même, nous devons pouvoir faire évoluer le réseau, vers de larges topologies, très denses. Cependant, des protocoles tels que TSCH et WirelessHART sont basés sur le TDMA, dans lequel gérer un grand nombre de noeuds est particulièrement complexe.
— **Coexistence et Interoperabilité** : la concentration d'un grand nombre de noeuds, utilisant tous la bande ISM, avec de multiples technologies radio possibles, rend l'accès au médium radio particulièrement complexe. Les différentes technologies radio peuvent interférer entre elles, et créer potentiellement des cas pathologiques. Ainsi, il est nécessaire que les objets détectent, classifient, et trouvent des solutions de réduction pour de telles interférences. Il existe deux grandes alternatives : chacun essaie de combattre les interférences de façon individuelle (par exemple avec des techniques de type saut de fréquence, CCA, codes correcteurs), soit en essayant de proposer des mécanismes de coopération qui partagent équitablement la bande passante entre tous.

C. Motivation

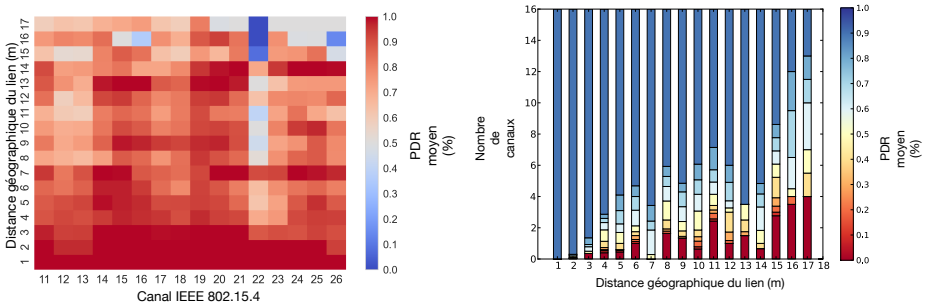
L'IIoT présente plusieurs défis qui représentent actuellement un obstacle à son adoption généralisée dans le monde industriel. Plus spécifiquement, l'exploitation de l'IIoT dans les applications critiques représente encore un champs non résolu.

L'utilisation du médium radio partagé, et plus particulièrement de la bande ISM des $2.4\ GHz$ crée des difficultés significatives pour obtenir de la haute fiabilité. En effet, cette bande représente une portion du spectre radio que tout un chacun peut utiliser sans payer pour des applications industrielles, médicales ou scientifiques. L'usage de cette bande est souvent préféré, du fait de l'absence de royalties à payer, et la facilité de déploiement d'un grand nombre de technologies du commerce. En particulier, l'IIoT souffre de sa faible puissance de transmission, comparativement aux autres technologies telles que le WiFi. Ainsi, les réseaux co-localisés comme le WiFi créent une interférence externe très importante, pénalisant la fiabilité.

La technique de saut de fréquence lent représente une solution prometteuse pour réduire l'impact des interférences externes. Les noeuds changent de façon synchrone le canal radio utilisé, telle que dans IEEE 802.15.4-TSCH, WirelessHART, et ISA100.11a. Les performances du saut de fréquence peuvent même être encore améliorés par des techniques de blacklistng, permettant de ne pas utiliser les fréquences radio les plus chargées.

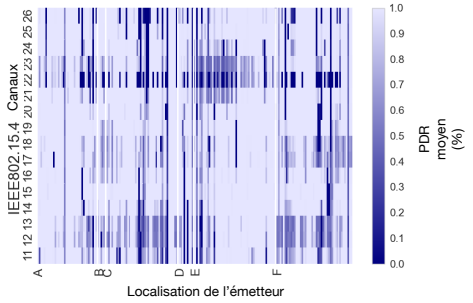
Le rôle des techniques de Blacklisting/Whitelisting correspond à évaluer la qualité des canaux radio dans la séquence de saut de fréquence, identifiant ceux offrant une faible fiabilité. Après avoir décidé des *mauvais* canaux radio, le réseau peut ainsi modifier la séquence de saut de fréquence, permettant d'utiliser en priorité les meilleurs canaux, c-a-d. ceux les moins bruités.

Typiquement, les réseaux IIoT supportent les communications multisaut, du fait de la faible puissance de transmission et de la surface importante de déploiement. Offrir une haute fiabilité ainsi qu'un faible délai de bout en bout représente un prérequis essentiel. Cependant, le multisaut crée un délai de buffering, qui s'ajoute le long de la route vers la destination. De plus, des paquets doivent également être retransmis, du fait de la non fiabilité des liens radio. Les techniques de Blacklisting/Whitelisting permettent ainsi d'exploiter



(a) Heatmap de la qualité moyenne de chaque canal selon sa distance euclidienne.

(b) Classification de la qualité de liens selon leur distance euclidienne.



(c) Heatmap du PDR pour chaque canal, les liens étant groupés selon leur proximité géographique.

FIGURE 1: Variabilité de la liste des mauvais canaux radio.

des liens plus fiables, et donc de réduire les délais de bout en bout. Il est également nécessaire de proposer des algorithmes d'ordonnancement conçus en fonction de cette contrainte de délai et de fiabilité.

II. CARACTÉRISATION EXPÉRIMENTALE EN ENVIRONNEMENT INTÉRIEUR

La croissance de l'Internet des Objets, qui exploitent des technologies radio utilisant la même bande de fréquence des $2.4GHz$ résulte en une large concentration de petits objets dans une même aire géographique, générant des interférences entre eux. Par ailleurs, le multipath fading accentue le manque de fiabilité des réseaux radio. En effet, le signal peut arriver au récepteur, en ayant emprunté plusieurs chemins radio possibles, rendant le signal plus instable. Le multipath fading peut être exacerbé dans des environnements industriels, présentant des structures métalliques qui réfléchissent les ondes électromagnétiques [Che16].

Nous avons donc commencé par quantifier le gain de performance des protocoles de saut de fréquence pour combattre les interférences externes. Pour ce faire, nous avons réalisé une étude expérimentale qui permet de caractériser la radio utilisant le standard IEEE 802.15.4 channels, et étudiant la connectivité en générale d'un environnement indoor.

Notre campagne de recherche expérimentale a mis en exergue que la qualité des liens radio est très variable dans le temps. Plus spécifiquement, nous avons distingué des canaux dont la qualité est stable au cours du temps, et d'autres pour lesquels la qualité subit d'intenses fluctuations au cours du temps. De plus, l'analyse des données expérimentales a montré une forte dépendance entre qualité et caractéristiques spatiales des liens. En particulier, il n'existe pas de corrélation entre RSSI et Taux de Livraison (PDR), particulièrement pour les liens de qualité moyenne. Ainsi, le RSSI semble être un indicateur inapproprié de la qualité.

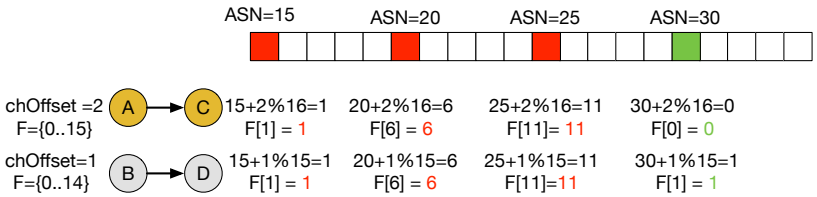


FIGURE 2: cellules en collision cells qui utilisent un channel offset et des blacklist différents (F[] dénote l'ensemble des bons canaux radio).

Nous avons également étudié les caractéristiques d'une possible blacklist (global versus local). En effet, plus de canaux veut dire plus de diversité, mais les moins bon canaux devraient ne pas être utilisés pour autant. Nous avons mis en exergue des propriétés locales, certains canaux étant mauvais pour un petit nombre de liens radio. De façon logique, la force du signal impacte la taille de la blacklist. En conclusion, une blacklist doit être locale, spécifique à une zone, ou même à une paire de noeuds.

Notre évaluation a montré qu'utiliser même des blacklists courtes permet d'améliorer significativement la fiabilité. De longues blacklists permettent d'améliorer la stabilité, au sacrifice de la capacité du réseau. Nous avons observé une stabilité certaine des blacklists, qui laisse présager que des solutions adaptatives sauraient réagir suffisamment rapidement, sans générer un trafic de contrôle excessif.

III. ALGORITHME DISTRIBUÉ DE BLACKLISTING PAR LIEN

Du fait des conclusions précédents, nous avons cherché à proposer une technique de blacklisting par lien radio. Nous avons donc proposé LABeL, pour Link Based Blacklisting. LABeL est flexible, adaptant les blacklists à la dynamique du réseau, répondant rapidement aux variations de qualité qui peuvent survenir. Nous nous efforçons également de limiter l'overhead généré, source de consommation d'énergie, et de réduction de la bande passante.

Nous avons employé un estimateur WMEWMA combine à un seuil dynamique de taux de livraison (PDR), afin de pouvoir identifier les mauvais canaux, et ainsi augmenter la fiabilité. Ainsi, LABeL modifie la séquence pseudo-aléatoire de saut de fréquence pour utiliser en priorité les bons canaux. Plus précisément, nous répétons la fonction pseudo-aléatoire jusqu'à obtenir un canal radio autorisé. Ainsi, nous créons des collisions de façon pseudo-aléatoire avec les autres émetteurs partageant le meme timeslot. En d'autres termes, ces collisions peuvent être gérées de la même manière que de l'interférence externe (Fig. 2).

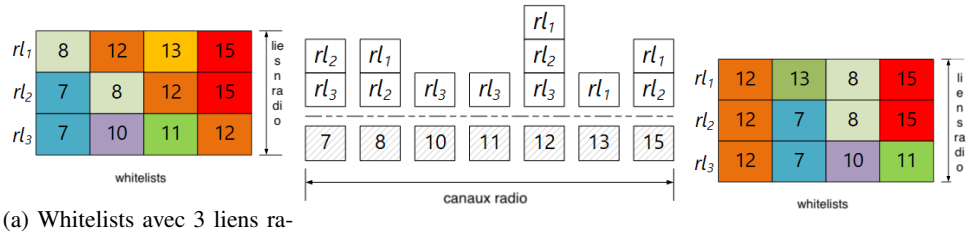
Pour maintenir des estimateurs de qualité à jour, même pour les mauvais canaux, LABeL intègre un mécanisme permettant d'utiliser les mauvais canaux afin de mettre à jour leurs indicateurs de qualité. Ces tests sont suffisamment peu fréquents pour limiter l'impact sur la fiabilité, tout en maintenant une estimation mise à jour en continu.

Notre évaluation expérimentale utilisant OpenWSN ainsi que FIT IoT-LAB montre que LABeL est adaptatif, et permet d'augmenter la fiabilité (de 20% dans certaines des situations), réduisant le trafic du aux retransmissions, et réduisant la gigue.

IV. ALGORITHME CENTRALISÉ DE WHITELISTING

Les technique de saut de fréquence permettent de réduire l'impact des interférences externes. LABeL adopte une technique de blacklisting par lien, permettant d'utiliser en priorité les meilleurs canaux. Cependant, LABeL reste pseudo-aléatoire : chaque lien radio décide du canal à utiliser selon une fonction pseudo-aléatoire dérivée de l'id de l'émetteur ainsi que du channel offset affecté. Cela se traduit en des collisions probabilistes, potentiellement créées au sein du réseau.

Nous avons donc également proposé un algorithme centralisé, capable de construire des blacklists, tout en interdisant la possibilité de créer des collisions. Dans un premier temps, nous devons identifier les mauvais



(a) Whitelists avec 3 liens radio dans le même timeslot (b) Pile des liens par canal radio (c) Whitelists réordonnées

FIGURE 3: Processus de réordonnancement des whitelists partageant le même timeslot.

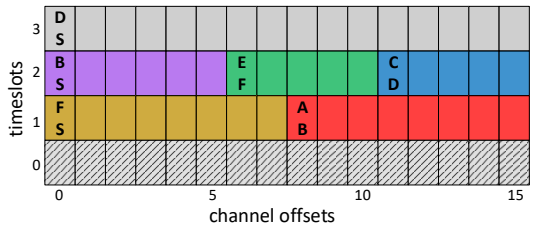


FIGURE 4: AMABO : assignation multiples uniform des channel offsets.

canaux. Nous utilisons directement la métrique de taux de livraison, ayant montré que le RSSI est un mauvais indicateur.

Ensuite, nous avons proposé une technique d'assignation de whitelist à chaque lien radio tout en évitant les collisions. Chaque whitelist comprend en priorité les meilleurs canaux (ceux présentant pour chaque lien radio le meilleur PDR). Pour éviter les collisions, nous groupons les whitelists par timeslot : deux liens radio ne partageant pas le même timeslot ne peuvent bien évidemment pas collisionner. Nous forçons ainsi chaque groupe de liens partageant le même timeslot à utiliser la même whitelist. Pour une assignation équitable, nous sélectionnons les canaux qui offrent le meilleur *rang moyen*.

Nous avons ensuite étendu cette technique, en proposant un algorithme ré-ordonnant les canaux présents dans les différentes whitelists pour des liens radio ordonnancés dans le même timeslot. En changeant la place des canaux dans les whitelist, nous nous efforçons qu'ils ne puissent jamais collisionner. Pour ce faire, nous avons défini les conditions suffisantes à respecter dans l'assignation.

Nos résultats de simulation, reposant sur des données expérimentales que nous avons collectés, montrent l'intérêt d'une telle technique quand les conditions sont stables. Elles permettent de discriminer les canaux peu fiables, tout en individualisant les whitelists.

V. TECHNIQUE DE BLACKLISTING HYBRIDE

Nous avons ensuite proposé d'adapter un algorithme hybride existant dans la littérature : MABO-TSCH . MABO-TSCH combine un algorithme centralisé, dans lequel des cellules sont assignées à chaque lien radio, avec une approche distribuée, où chaque lien décide de la fréquence à utiliser parmi les cellules qui lui ont été octroyées. Bien que MABO-TSCH interdise les collisions entre liens radio, il montre ses limites pour les blacklists longues.

Nous avons donc proposé d'étendre MABO-TSCH en concevant Adaptive MABO (AMABO). Nous utilisons plus efficacement tous les channel offsets disponibles (par défaut 16 dans IEEE 802.15.4). En effet, une assignation statique d'un nombre fixe de channel offsets est sous-optimale : certains channel offsets ne sont pas du tout utilisés alors qu'ils permettraient d'augmenter la fiabilité.

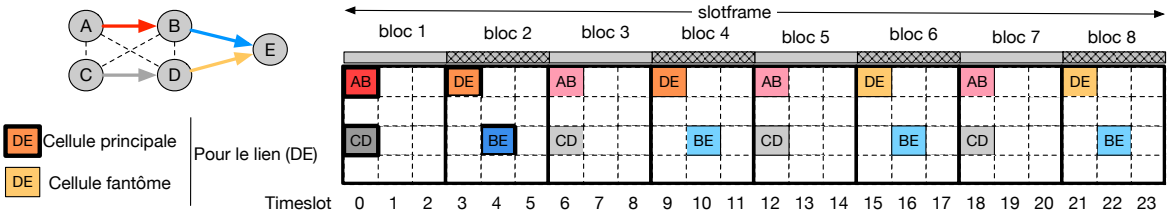


FIGURE 5: Ordonnancement en blocs de LDSF.

Nous avons donc proposé une solution adaptative, assignant tous les channel offsets disponibles dans un certain timeslot (Fig. 4). La matrice d'ordonnancement n'est ainsi plus creuse, et nous augmentons les choix disponibles à un lien radio lorsqu'il a peu de *compétiteurs* dans la matrice (cad. partageant le même timeslot).

Notre évaluation exploitant le jeu de données expérimentales montre que cette approche permet bien d'augmenter la fiabilité, en assignant en moyenne plus de channel offsets par lien radio.

VI. ORDONNANCEMENT DISTRIBUÉ À FAIBLE LATENCE

Les réseaux de contrôle pour l'automatisme sont supposés offrir une très haute fiabilité et une latence bornée. Nous avons précédemment expliqué comment les techniques de blacklisting aident à atteindre un tel but. Cependant, offrir un délai de bout en bout borné est un défi particulièrement complexe, puisque les transmissions sont chainées le long de la route. Ainsi, une seule retransmission peut décaler entièrement les retransmissions le long de la slotframe, créant un effet en cascade sur le délai de buffering.

Nous avons donc proposé Low-latency Distributed Scheduling Function (LDSF) conçue spécifiquement pour offrir un délai faible de bout en bout, même en cas de retransmissions. LDSF est entièrement distribué : chaque lien radio choisit les cellules à utiliser pour échanger des paquets. Notre solution repose sur une organisation de la slotframe en sous-parties, que nous avons appelés blocs.

Chaque émetteur choisit sont bloc en fonction de sa distance en nombre de sauts de la destination. Lorsqu'il doit émettre un paquet, il calcule tout d'abord l'arrivée au plus tôt de ce paquet, s'il n'a subit aucune retransmission. Ensuite, il réserve automatiquement une cellule tous les deux blocs. Si le paquet a subi une retransmission, une nouvelle cellule est disponible juste après, n'augmentant le délai que de la longueur d'un bloc, et plus d'une slotframe.

Nous avons également proposé un mécanisme d'endormissement. Un émetteur sait qu'il n'a rien à émettre et s'endort dés que son buffer est vide. Le récepteur lui connait le temps d'arrivée au plus tôt d'un paquet. Il peut donc s'endormir dés qu'il a reçu son paquet, sachant qu'aucune retransmission ne viendra ultérieurement (Fig. 5).

Nous avons montré par simulation que LDSF est très efficace pour offrir une très faible gigue, la réception n'étant décalée que de quelques blocs. Par ailleurs, LDSF est parfaitement adaptative : si un lien voit sa qualité se dégrader, des opportunités de retransmissions sont automatiquement disponibles, et il n'est nul besoin de tout modifier le long de la route dans la matrice d'ordonnancement.

VII. LISTES DES PUBLICATIONS

Cette contribution a donné lieu aux publications suivantes :

A. *Journaux et Magazines internationaux*

— *"LDSF : Low-latency Distributed Scheduling Function for Industrial Internet of Things"*
Kotsiou, V., Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
In IEEE **Internet of Things Journal**, **accepted**.

— *"Whitelisting without Collisions for Centralized Scheduling in Wireless Industrial Networks"*
Kotsiou, V., Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
In IEEE **Internet of Things Journal**, March 2019.
— *"Blacklisting-Based Channel Hopping Approaches in Low-Power and Lossy Networks"*,
Kotsiou, V, Papadopoulos, G. Z. Zorbas, D., Chatzimisios, P., & Theoleyre, F.
In IEEE **Communications Magazine**, February 2019.

B. Conférences et Workshops internationaux

— *"Adaptive Multi-Channel Offset Assignment for Reliable IEEE 802.15.4 TSCH Networks"*
Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P.,& Théoleyre, F.
In *2018 Global Information Infrastructure and Networking Symposium (**GIIS**)* (pp. 1-5). IEEE, Thessaloniki, October 2018.
— *"Label : Link-based adaptive blacklisting technique for 6tisch wireless industrial networks"*
Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
In *Proceedings of the 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (**MSWiM**)* (pp. 25-33), ACM, Miami, November 2017.
— *"Is local blacklisting relevant in slow channel hopping low-power wireless networks ?"*
Kotsiou, V, Papadopoulos, G. Z. Chatzimisios, P., & Theoleyre, F.
In 2017 *International Conference on Communications (**ICC**)* (pp. 1-6), IEEE, Paris, May 2017.

RÉFÉRENCES

[AI+11] Mohammed H ALMARSHADI, Saleh M ISMAIL et al. "Effects of precision irrigation on productivity and water use efficiency of alfalfa under different irrigation methods in arid climates". In : *Journal of Applied Sciences Research* 7.3 (2011), p. 299-308.

[AIM10] Luigi ATZORI, Antonio IERA et Giacomo MORABITO. "The internet of things : A survey". In : *Computer networks* 54.15 (2010), p. 2787-2805.

[All12] ZigBee ALLIANCE. *New ZigBee PRO feature : green power*. 2012.

[Aro+04] Anish ARORA et al. "A line in the sand : a wireless sensor network for target detection, classification, and tracking". In : *Computer Networks* 46.5 (2004), p. 605-634.

[BBB04] Jenna BURRELL, Tim BROOKE et Richard BECKWITH. "Vineyard computing : Sensor networks in agricultural production". In : *IEEE Pervasive computing* 3.1 (2004), p. 38-45.

[Cam+07] Alberto CAMILLI, Carlos E CUGNASCA, Antonio M SARAIVA, André R HIRAKAWA et Pedro LP CORRÊA. "From wireless sensors to field mapping : Anatomy of an application for precision agriculture". In : *Computers and Electronics in Agriculture* 58.1 (2007), p. 25-36.

[Che16] Michael CHEFFENA. "Propagation channel characteristics of industrial wireless sensor networks [wireless corner]". In : *IEEE Antennas and Propagation Magazine* 58.1 (2016), p. 66-73.

[Cis] CISCO. *The Internet of Things Infographic*. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/iot-aag.pdf.

[Dec05] J-D DECOTIGNIE. "Ethernet-based real-time and industrial communications". In : *Proceedings of the IEEE* 93.6 (2005), p. 1102-1117.

[DNW07] Serhan DAGTAS, Yuri NATCHETOI et Huaigu WU. "An integrated wireless sensing and mobile processing architecture for assisted living and healthcare applications". In : *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. 2007, p. 70-72.

[EKM11] Melike EROL-KANTARCI et Hussein T MOUFTAH. "Wireless multimedia sensor and actor networks for the next generation power grid". In : *Ad Hoc Networks* 9.4 (2011), p. 542-551.

[Fun06] HART Comunication FUNDATION. "HART field communication protocol specification". In : *HFC_SPEC12, Revision* 6 (2006).

[Ieea] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)". In : *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016).

[Ieeb] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 15 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN)". In : *IEEE Std 802.15.4-2003* (2003), p. 1-680.

[ISAy] ISA-100.11A-2011 : "Wireless Systems for Industrial Automation :Process Control and Related Applications". In : *International Society of Automation (ISA) Std.* 1 (May 2011).

[Itu] *ITU, internet reports 2005 : The internet of things*. tech. report. ITU, 2005.

[JGL10] Val JONES, Valerie GAY et Peter LEIJDEKKERS. "Body sensor networks for mobile health monitoring : Experience in europe and australia". In : *2010 Fourth International Conference on Digital Society*. IEEE. 2010, p. 204-209.

[Jin18] Vandana JINDAL. "History and architecture of wireless sensor networks for ubiquitous computing". In : *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 7.2 (2018), p. 214-217.

[KL09] Albert KO et Henry YK LAU. "Robot assisted emergency search and rescue system with a wireless sensor network". In : *International Journal of Advanced Science and Technology* 3 (2009), p. 69-78.

[LBV06] Koen LANGENDOEN, Aline BAGGIO et Otto VISSER. "Murphy loves potatoes : Experiences from a pilot sensor network deployment in precision agriculture". In : *Proceedings 20th IEEE international parallel & distributed processing symposium*. IEEE. 2006, 8-pp.

[LKK10] Yujin LIM, Hak-Man KIM et Sanggil KANG. "A design of wireless sensor networks for a power quality monitoring system". In : *Sensors* 10.11 (2010), p. 9712-9725.

[Lon+17] Stefano LONGO, Tingli SU, Guido HERRMANN et Phil BARBER. *Optimal and robust scheduling for networked control systems*. CRC press, 2017.

[LWS04] Ruizhong LIN, Zhi WANG et Youxian SUN. "Wireless sensor networks solutions for real time monitoring of nuclear power plant". In : *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*. T. 4. IEEE. 2004, p. 3663-3667.

[NK06] Mikael M NORDMAN et Taneli KORHONEN. "Design of a concept and a wireless ASIC sensor for locating earth faults in unearthed electrical distribution networks". In : *IEEE transactions on power delivery* 21.3 (2006), p. 1074-1082.

[OP99] Bob O'HARA et Al PETRICK. *The IEEE 802.11 Handbook : A Designer's Companion*. Standards Information Network IEEE Press, 1999.

[PD08] K.S.J. PISTER et L. DOHERTY. "TSMP : Time Synchronized Mesh Protocol". In : *Parallel and Distributed Computing and Systems*. 2008.

[PTT] Georgios Z PAPADOPOULOS, Fabrice THEOLEYRE et Pascal THUBERT. "Operations, Administration and Maintenance (OAM) features for Reliable and Available Wireless (RAW) Networks". In : *Internet Technology Letters* ().

[RFG19] Saleem RAZA, Muhammad FAHEEM et Mesut GUENES. "Industrial wireless sensor and actuator networks in industry 4.0 : Exploring requirements, protocols, and challenges—A MAC survey". In : *International Journal of Communication Systems* 32.15 (2019), e4074.

[Sef+20] Amina SEFERAGIĆ, Jeroen FAMAEY, Eli DE POORTER et Jeroen HOEBEKE. "Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things". In : *Sensors* 20.2 (2020), p. 488.

[Sis+18] Emiliano SISINNI, Abusayeed SAIFULLAH, Song HAN, Ulf JENNEHAG et Mikael GIDLUND. "Industrial internet of things : Challenges, opportunities, and directions". In : *IEEE Transactions on Industrial Informatics* 14.11 (2018), p. 4724-4734.

[Spe08] WirelessHART SPECIFICATION. "75 : TDMA Data-Link Layer". In : *HART Communication Foundation Std., Rev* 1 (2008).

[SSW06] Kewei SHA, Weisong SHI et Orlando WATKINS. “Using wireless sensor networks for fire rescue applications : Requirements and challenges”. In : *2006 IEEE International Conference on Electro/Information Technology*. IEEE. 2006, p. 239-244.

[Sur+12] NK SURYADEVARA, Anuroop GADDAM, RK RAYUDU et SC MUKHOPADHYAY. “Wireless sensors network based safe home to care elderly people : Behaviour detection”. In : *Sensors and Actuators A : Physical* 186 (2012), p. 277-283.

[Tho05] J-P THOMESSE. “Fieldbus technology in industrial automation”. In : *Proceedings of the IEEE* 93.6 (2005), p. 1073-1101.

[TZC06] Vanessa WS TANG, Yuan ZHENG et Jiannong CAO. “An intelligent car park management system based on wireless sensor networks”. In : *2006 First International Symposium on Pervasive Computing and Applications*. IEEE. 2006, p. 65-70.

[WA+06] Geoffrey WERNER-ALLEN, Konrad LORINCZ, Mario RUIZ, Omar MARCILLO, Jeff JOHNSON, Jonathan LEES et Matt WELSH. “Deploying a wireless sensor network on an active volcano”. In : *IEEE internet computing* 10.2 (2006), p. 18-25.

[WADHW08] G. WERNER-ALLEN, S. DAWSON-HAGGERTY et M. WELSH. “Lance : Optimizing High-Resolution Signal Collection in Wireless Sensor Networks”. In : *SenSys*. ACM, 2008.

[Wan+16] Jiafu WAN, Shenglong TANG, Zhaogang SHU, Di LI, Shiyong WANG, Muhammad IMRAN et Athanasios V VASILAKOS. “Software-defined industrial internet of things in the context of industry 4.0”. In : *IEEE Sensors Journal* 16.20 (2016), p. 7373-7380.

[Wei91] Mark WEISER. “The Computer for the 21 st Century”. In : *Scientific american* 265.3 (1991), p. 94-105.

[WJ16] Quan WANG et Jin JIANG. “Comparative examination on architecture and protocol of industrial wireless sensor network standards”. In : *IEEE Communications Surveys & Tutorials* 18.3 (2016), p. 2197-2219.

[WSJ17] Martin WOLLSCHLAEGER, Thilo SAUTER et Juergen JASPERNEITE. “The future of industrial communication : Automation networks in the era of the internet of things and industry 4.0”. In : *IEEE industrial electronics magazine* 11.1 (2017), p. 17-27.

[Xu+04] Ning XU, Sumit RANGWALA, Krishna Kant CHINTALAPUDI, Deepak GANESAN, Alan BROAD, Ramesh GOVINDAN et Deborah ESTRIN. “A wireless sensor network for structural monitoring”. In : *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004, p. 13-24.

[Yan+15] Jiachen YANG, Jianxiong ZHOU, Zhihan LV, Wei WEI et Houbing SONG. “A real-time monitoring system of industry carbon monoxide based on wireless sensor networks”. In : *Sensors* 15.11 (2015), p. 29535-29546.

[Zha+04] P. ZHANG, C. M. SADLER, S. A. LYON et M. MARTONOSI. “Hardware Design Experiences in ZebraNet”. In : *SenSys*. ACM, 2004.

Université de Strasbourg

Vasileios KOTSIOU

Reliable Communications for the Industrial Internet of Things

Résumé

L'Internet des Objets Industriel (IIoT) cible les applications critiques telles que les usines intelligentes. Cependant, les applications industrielles requièrent souvent une haute fiabilité. Afin de répondre à ces contraintes, IEEE 802.15.4 a proposé le mode Time-Slotted Channel Hopping (TSCH). TSCH ordonnance les transmissions afin d'éviter les collisions, et du saut de fréquences lent pour combattre les interférences externes. Nous proposons ici d'améliorer TSCH pour les applications industrielles critiques. Nous avons tout d'abord caractérisé spatialement et temporellement les liens radio, à travers une série d'expérimentations sur testbeds. Nous avons également proposé des techniques de type blacklisting permettant d'exclure les moins bons canaux de la séquence de saut de fréquences, permettant ainsi d'améliorer globalement la fiabilité. Finalement, nous avons proposé un algorithme d'ordonnancement des transmissions permettant de fournir haute fiabilité et faible délai.

Résumé en anglais

The Industrial Internet of Things (IIoT) re-uses the Internet of Things mechanisms to enable smart factories. However, industrial applications often require deterministic communications as well as end-to-end reliability close to 100%. To address these requirements, the Time-Slotted Channel Hopping (TSCH) mode of the IEEE 802.15.4 standard was proposed in 2015. TSCH schedules the transmissions to avoid collisions and exploits a slow channel hopping technique to combat external interference. TSCH can be further improved for the IIoT to be exploited in critical industrial applications, which is the main goal of this work. Towards this direction, we highlighted radio links' spatial and temporal characteristics by conducting experiments in indoor testbeds. We proposed blacklisting techniques that exclude from the channel hopping sequence the low-quality channels, thus enhancing the overall performance. Finally, we proposed a scheduling function that aims to meet the requirements of IIoT.