Université de Strasbourg

UNIVERSITÉ DE STRASBOURG

msii ÉCOLE DOCTORALE

ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR

Laboratoire ICube – UMR 7357

THÈSE présentée par :

[Rodrigo TELES HERMETO]

soutenue le : **28 novembre 2019**

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

Discipline/ Spécialité : Informatique

Standard Improvements and Predictable Performance for Industrial Internet of Things in Indoor Deployments

THÈSE dirigée par :	
M. THEOLEYRE Fabrice	Chargé de recherche, Université de Strasbourg
RAPPORTEURS :	
Mme. CARNEIRO VIANA Aline	Chargé de recherche, INRIA Saclay
M. TOURANCHEAU Bernard	Professeur, Université Grenoble Alpes
AUTRES MEMBRES DU JURY :	
M. GALLAIS Antoine	Professeur, Université Polytechnique Hauts-de-France
Mme. PALATTELLA Maria Rita	Chargé de recherche, Luxembourg Institute of Science and Tech.
M. VALOIS Fabrice	Professeur, INSA Lyon

Université de Strasbourg

msii ÉCOLE DOCTORALE

Rodrigo TELES HERMETO

Standard Improvements and Predictable Performance for Industrial Internet of Things in Indoor Deployments

Résumé

Les réseaux industriels sont utilisés pour surveiller les processus liés à la sécurité, où une fiabilité élevée et des délais prévisibles doivent être assurés. Pour cette raison, la norme IEEE 802.15.4-2015 a été publiée en 2016, en définissant le mode TSCH (Time-Slotted Channel Hopping). TSCH permet l'ordonnancement des transmissions où chaque nœud dispose de ressources dédiées pour communiquer en évitant les collisions. De plus, le mécanisme de saut de canal permet aux nœuds d'atténuer les effets des interférences externes. Toutefois, les pertes de paquets continuent à se produire en raison de variations de radio. L'objectif de ce travail est d'améliorer la fiabilité des réseaux sans fil dans les environnements intérieurs, où les obstacles et les interférences externes sont la règle. Nous nous concentrons principalement sur la recherche expérimentale pour identifier les limites et dans quelles circonstances ces réseaux ne parviennent pas à fournir une performance prévisible.

Résumé en anglais

Industrial networks are typically used to monitor safety-related processes, where high reliability and an upper-bounded delay must be ensured. To attend these requirements, IEEE 802.15.4-2015 standard was published in 2016, defining the Time-Slotted Channel Hopping (TSCH) mode. TSCH allows the scheduling of transmissions, such that each device has enough opportunities for communicating while avoiding collisions. In addition, slow-channel hopping mechanism allows the nodes to combat the effect of external interference. Although TSCH increases the reliability, packet losses keep on occurring due to variations on the radio conditions, very common in indoor environments. The goal of this work is to improve the reliability of low-power wireless networks in indoor scenarios, where obstacles and external interference are the rule. We focus mostly on experimental research to identify the limits and in which circumstances these networks fail at providing a predictable performance.

Université de Strasbourg

École doctorale MSII

Standard Improvements and Predictable Performance for Industrial Internet of Things in Indoor Deployments

THÈSE

pour obtenir le grade de

Doctorat de l'Université de Strasbourg

(mention informatique)

28 novembre 2019

présentée par

Rodrigo TELES HERMETO

Composition du jury

Directeur de thèse:	Dr. Fabrice THEOLEYRE, CNRS, Université de Strasbourg, France
Co-encadrant:	Prof. Antoine GALLAIS, Université Polytechnique Hauts-de-France, France
Rapporteurs:	Dr. Aline CARNEIRO VIANA, INRIA Saclay, France Prof. Bernard TOURANCHEAU, Université Grenoble Alpes, France
Examinateurs:	Dr. Maria Rita PALATTELLA, LIST, Luxembourg Prof. Fabrice VALOIS, INSA Lyon, France

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

Acknowledgments

Phew! It's been three long years that has gone really fast. Doing a PhD is not easy, it requires commitment and perseverance for getting more negative then positive results. Firstly, I have to thank my wife Ana Alice for deciding to come with me for this journey, leaving our country and start a new life in a new country is not easy either. Her support, incentive and comprehension during the absent periods (especially before deadlines) were remarkable.

I would also to thank my two supervisors Fabrice Theoleyre and Antoine Gallais for mentoring me during this PhD. This work wouldn't be possible without their support and guidance. From the first to the last day, they were there always available for discussion. I have no word to express how grateful and lucky I feel for having you two as mentors. Thank you for pushing me beyond my limits.

I would like to thank Dr. Aline Viana, Prof. Bernard Tourancheau, Dr. Maria Rita Palattella and Prof. Fabrice Valois for having accepted to be member of my PhD jury. Their constructive comments and remarks helped me to improve the quality of this manuscript.

I also would like to thank my colleagues from the Réseaux research group: Thomas, Jean-Jacques, Cristel, Quentin, Stéphane, Pierre, Pascal, Julien and Guillaume. Since the first day, I felt well received by them. Thank you for the advices on how to improve my presentation and teaching skills.

A big thanks to *les gars*, or the other PhD students: Julian, Amine, Andreas, Philippe, Odnan, Renato, Sebastian, Jean-Romain and Loic. We all have become good friends, and their friendship and companionship made my staying in Strasbourg much easier and more fun. Thank you all for your patience, I know I complain a lot, especially the coffee taste and the heat during the summer. I hope you all know how to prepare a good coffee now!

A special thanks to my parents Joao and Mariana, my sister Danyelle, my brother-in-law Kevin, my nephew Michael and my niece Juliana for encouraging me to follow my dreams.

Abstract

Industrial Internet of Things (IIoT) is currently an emerging trend, aiming to adapt the Internet of Things (IoT) concepts in industrial deployments. Since industrial networks are typically used to monitor safety-related processes, they impose strict performance requirements. In particular, high reliability (e.g. $> 99\%$) and an upper bounded delay must be ensured.

With recent low-power wireless standards, industrial networks rely more and more on wireless communications. To provide Quality of Service (QoS) for industrial-like wireless networks, IEEE 802.15.4-2015 standard was published in 2016, defining the Time-Slotted Channel Hopping (TSCH) mode. TSCH allows the scheduling of transmissions, such that each network device has enough opportunities for communicating while avoiding collisions. In addition, slow-channel hopping mechanism allows the nodes to combat the effect of external interference and multipath fading.

Although TSCH increases reliability, packet losses keep on occurring due to variations on the radio conditions caused by uncontrolled and unpredictable factors. In particular, environment characteristics, obstacles and external interference lead to the attenuation of the radio signals, degrading the network performance. Additionally, low-power wireless networks emit weaker signals, making their transmissions even more prone to background noise and distortion.

The goal of this thesis is to improve the reliability of low-power wireless networks in indoor scenarios, where obstacles and external interference are the rule. We focus mostly on experimental research to identify the limits and in which circumstances standards widely used in IIoT deployments fail in providing a predictable performance. In particular, we target the 6TiSCH stack, since it encompasses higher layer protocols to grant IPv6 connectivity while providing high reliability through the TSCH protocol.

In support of this thesis, we first have (i) proposed a passive method to estimate accurately the link quality between neighbor nodes, (ii) improved the network stability by solving schedule inconsistencies and reducing useless routing reconfigurations, (iii) analyzed the interest of using anycast transmissions at the link-layer and (iv) modeled the network attachment delay of mobile nodes taking into account control plane traffic of 6TiSCH networks. With the refinements proposed in this thesis, we provide means that allow the network to respect the reliability and delay constraints, even in the presence of abrupt variations on the radio conditions.

List of Publications

International Journals

1. R. Teles Hermeto, A. Gallais and F. Theoleyre. Experimental in-depth Study of the Dynamics of an Indoor Industrial Low Power Lossy Network, in Elsevier Ad Hoc Networks, Vol. 93, October 2019;

2. R. Teles Hermeto, A. Gallais and F. Theoleyre. Impact of the Initial Preferred Parent Choice, in IEEE COMSOC MMTC Communications - Frontiers. pp. 43-46, Vol.12, No.6, November 2017;

3. R. Teles Hermeto, A. Gallais and F. Theoleyre. Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey, in In Elsevier Computer Communications, vol. 114, pp. 84-105, December 2017.

International Conferences

1. R. Teles Hermeto, Q. Bramas, A. Gallais and F. Theoleyre. Analysis of the Network Attachment Delay of Mobile Devices in the Industrial Internet of Things, in the 18th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now), Luxembourg, Luxembourg, October 2019.

2. R. Teles Hermeto, A. Gallais and F. Theoleyre. Is Link-Layer Anycast Scheduling Relevant for IEEE802.15.4-TSCH Networks?, in IEEE Conference on Local Computer Networks (LCN) - Osnabrück, Germany, October 2019;

3. R. Teles Hermeto, A. Gallais and F. Theoleyre. On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment, in ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) - Montreal, Canada, October 2018;

4. R. Teles Hermeto, A. Gallais, K. Van Laerhoven and F. Theoleyre. Passive Link Quality Estimation for Accurate and Stable Parent Selection in Dense 6TiSCH Networks, in ACM International Conference on Embedded Wireless Systems and Networks (EWSN) - Madrid, Spain, February 2018;

Contents

Chapter 1

Introduction

For this thesis, we have decided to describe the research conducted since the beginning of this PhD in 2016. During the last 3 years, we have conducted mostly experimental research on Industrial Internet of Things (IIoT). We have proposed improvements for reliable protocols that are typically used in IIoT deployments, targeting specifically indoor environments with concurrent wireless technologies. We believe that scenarios where multiple wireless networks have to cohabit in the same environment causing mutual interference are inevitable, specially with the raising of new technologies for the Internet of Things (IoT) seen in the last years. Ensuring high reliability while relying on a crowded spectrum represents a key requirement for IIoT deployments in the near future.

1.1 Context

In the next sections, we present a brief overview of the main topics discussed in this thesis.

1.1.1 The Internet of Things

The Internet of Things(IoT) is a concept that dominates the modern wireless telecommunications. In the IoT context, any common object (e.g. cars, bikes, lamps) is connected to the Internet and thus are accessible to the "external world". Typically, these objects rely on embedded devices with processing, sensing and communications capacities, turning them into *smart objects*. Through the use of sensors and actuators, a smart object can percept the world and interact with it, while being accessible through the Internet. Anyone (or anything) on the Internet can potentially request information or send commands to them, creating a new kind of automation like never before. The types of applications are numerous, from home automation [1] to smart cities [2].

The wide adoption of low-power embedded devices has attracted much attention to the IoT. While Cisco has predicted 50 billions small IoT devices by 2020 [3], Huawei even expects the deployment of 100 billions by 2025 [4]. Since these objects rely on unique address schemes, the transition to IPv6 is inevitable. Besides the larger address pool, IPv6 enhances the support for secure networks, key aspect to IoT applications.

The IoT scope is large and it includes many different network technologies. For instance, the Radio Frequency Identification System (RFID) is referred to be the first technology used in the IoT context [5]. Near-Field Communication (NFC) has also been used in IoT applications [6]. However, Wireless Sensor Networks (WSN) represent one of the base technologies used in IoT, since it can be easily extended to build more complex applications. In particular, WSNs supports different networks topologies, has low power consumption and can be used in large scale deployments.

1.1.2 Wireless Sensor Network and low-power wireless standards

A Wireless Sensor Network (WSN) is a low-power network of autonomous embedded devices (aka nodes) that sense the environment and communicate cooperatively to transmit their data to one or more *sinks*. A sink can work as gateway, connecting the network to the Internet, or as a base station that process the received information locally. Typically, a node collects through its sensors environmental conditions, such as temperature, humidity, light, etc. The collected information can then flow from all nodes to the sink (convergecast) or from the sink to the nodes (point-to-multipoint). In the latter case, the nodes are not only periodic data sources, they are also addressable end-points.

WSNs have been used for a plethora of applications with different characteristics. For instance, Lloret *et al.* [7] deployed a fire detection system for monitoring wildfire in rural environments. Djurivsic *et al.* [8] reviewed a collection of works focusing on military applications. WSNs have also been used in healthcare applications to assist elderly people in their daily home activities [9]. A WSN has been even deployed underwater to monitor the pollution levels on the seabed [10]. Therefore, the increasing number of WSN applications in the last years can be explained by its associated low-cost and ease of deployment, making them an attractive option for IoT deployments.

For communicating, the nodes in a WSN employ low-power wireless standards. The IEEE 802.15.4 standard [11] defines the set of operations for the physical and Media Access Control (MAC) layers, offering the foundation for a Wireless Personal Area Network (WPAN). In a general way, the standard was designed to be energy-efficient, since the nodes are battery powered with limited computational power. For this reason, the communication range between nodes is restricted to a dozen of meters, the maximum transfer rate is 250 kbit/s (2.45 GHz) and the Maximum Transmission Unit (MTU) size is 127 bytes. To cope with the limited communication range, a WSN employs typically a multihop topology, where the data is transmitted hop by hop toward the destination. These low-power characteristics are in contrast to the Wireless Local Area Network (WLAN), such as the Wi-Fi, which offers higher connection speed and longer communication range at the cost of higher energy consumption.

In contrast with the maximum packet size in IEEE 802.15.4, IPv6 supports links with a MTU of 1280 bytes. To cope with this limitation, the Internet Engineering Task Force (IETF) 6LoWPAN Working Group was created to work on an adaption layer to provide IPv6 Internet connectivity to low-power wireless networks. [12].

The 6LoWPAN group has defined extensions (e.g. fragmentation and reassembly, header compression) that allows IPv6 packets to be transmitted on IEEE 802.15.4 links.

Besides, existing routing protocols are not optimized to the constraints (e.g. energy-efficiency, scalability) in low-power wireless networks [13]. Thus, the IETF ROLL WG has defined the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [14] targeting specifically constrained devices. RPL has become the main routing protocol used in IoT deployments [15].

Even representing a cost-effective way for deploying IoT projects, a WSN is also known to be lossy with no delivery guarantees [16]. Interference from higher power networks (e.g. Wi-Fi, Bluetooth) and temperature variations are some examples of factors that make the links to exhibit time-variable characteristics [17]. While the early adopters bought small best-effort solutions for leisure, modern applications are more and more demanding in terms of responsive communications with high reliability. It is the case of industrial networks.

1.1.3 Communication in industrial networks

An industrial network is composed by a large collection of sensors and actuators that are connected to a Programmable Logic Controllers (PLC) to take real-time decisions [18]. The sensors are typically used to measure continuous and discrete process variables, while the actuators convert energy into motion. This kind of network has been widely deployed for a myriad of utilizations. For instance, a collection of sensors and actuators can detect intrusions, or control the Heating, Ventilation and Air-Conditioning system in building automation [19].

Because industrial networks monitor critical processes, they have strict performance requirements: high reliability and upper bound latency are critical. To attend such requirements, industrial networks are typically deployed on top of costly and inflexible wired infrastructures [20]. For instance, the Highway Addressable Remote Transducer (HART) [21] and the FieldBus [22] protocols are widely used to collect information from process instrumentations. The HART protocol employs a request/reply communication, or a master-slave model. In this case, the controller is in charge of requesting information from the field devices. On the other hand, the FieldBus protocol employs the peer-to-peer concept, where field devices can talk to each other independently.

A wired infrastructure has the advantage of providing higher connection speed, resilience to interference and it connects systems across long distances. The drawback is the associated cost of deploying and maintaining a large wired infrastructure. In many cases, an industrial process has to be disrupted if the network needs to undergo repairs or extensions [23]. Naturally, this brings additional expenses to the industry economical activities.

In order to reduce deployment and maintenance costs, industrial networks have started to replace this legacy infrastructure with WSNs. Therefore, sensors and actuators have now to interact in real-time: a reliable delivery of the command packets before a given deadline is expected. A best-effort solution is not anymore acceptable, since it does not guarantee a predictable delivery. The Medium Access Control (MAC) represents a key element in the networking stack to achieve such objective.

The MAC layer is in charge of deciding when a node is authorized to transmit to avoid both deafness (the receiver is not awake) and collisions (two transmissions overlap). Therefore, it needs to share the radio bandwidth among the different contenders, while minimizing the number of collisions and limiting the transmission delay. Random Access is particularly efficient to deal with a variable traffic and number of transmitters. In particular, random techniques such as Slotted Aloha or the Carrier Sense Multiple Access (CSMA) provide a very low latency with negligible packet losses when the traffic is very low. However, fairness, reliability and reactivity decrease very significantly as soon as the load exceeds a threshold. Thus, pure random access methods seem inaccurate, since they do not guarantee the reliability required for most industrial deployments.

To cope with the lossy nature of low-power wireless networks, standards have been proposed to provide predictable performance on top of an unreliable link layer. The IEEE 802.15.4-TSCH standard has proposed the Time Slotted Channel Hopping (TSCH) mode for industrial wireless sensor networks [24]. It is mainly built upon the previous WirelessHART [25] and ISA100.11a [26] standards, and the Time Synchronized Mesh Protocol [27]. IEEE 802.15.4-TSCH employs a strict organization of the transmissions to achieve higher reliability. By relying on Time Division Multiple Access (TDMA) paired with a synchronization mechanism, IEEE 802.15.4-TSCH removes most of the collisions. Besides, slow channel hopping has been proved to combat efficiently narrow band noise, very common in industrial environments [28]. The channel hopping mechanism adds frequency diversity by enabling transmissions and retransmissions on different channels, following a pseudo-random sequence. While IEEE 802.15.4-TSCH focuses on high reliability and low latency, it imposes to schedule the transmissions. Interfering transmitters should not be allowed to transmit simultaneously, to avoid collisions.

1.1.4 Industrial Internet of Things

IIoT is currently an emerging approach, aiming to re-use the IoT concepts in the automation world. The main objective of the IIoT is to leverage on the Internet of Things to make the industrial chain more malleable. In particular, we expect the industry evolving to a distributed chain production, combining different manufacturing processes that are now completely independent of each other. For instance, geographically distant factories can collaborate, sharing information to enhance their production chains. Interoperability is then essential to achieve such objective.

Therefore, some vendors have pushed their own independent solutions to interconnect independent systems through the IPv6 protocol. Because of the lack of interoperability between solutions from different vendors, the 6TiSCH IETF working group [29] was formed to standardize the employment of IPv6 on low-power wireless networks. 6TiSCH grants IPv6 connectivity by providing a low-power protocol stack that "glues" together higher layer protocols (6LoWPAN, RPL, CoAP [30]) and TSCH through the 6P sublayer [31]. Therefore, 6TiSCH has become *the* standard for the IIoT. For that reason, we employ the 6TiSCH stack in our research.

We expect a very large adoption of the IIoT, in various key areas. Smart agriculture would exploit a radio infrastructure to monitor in real time a greenhouse, or

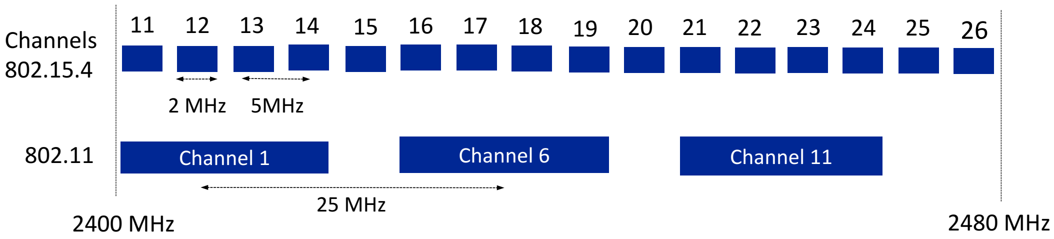


Figure 1.1: Overlapping channels between IEEE 802.11 and IEEE 802.15.4 standards.

a field [32]. The European Telecommunications Standards Institute (ETSI) has detailed the requirements (delay, reliability, volume of traffic) for different applications in Smart Cities [33]. To this purpose, real-time systems are needed, encompassing the operating system, the application and the communication protocols. End-to-end performance and delivery guarantees are the alpha and the omega.

1.2 Motivations

The ISM band represents a group of license free radio frequencies that are internationally reserved for operation of industrial, scientific and medical equipments. Many wireless network technologies operate within the ISM band (e.g. Wi-Fi, Bluetooth, RFID and IEEE 802.15.4).

Because of its low-power nature, a IEEE 802.15.4 network is particularly in disadvantage when collocated with others higher power wireless networks, such as IEEE 802.11 networks. The transmission power of IEEE 802.11 devices is 30 dBm, which is much higher than the 0 dBm employed by IEEE 802.15.4 devices. This asymmetry impacts directly the performance and the reliability of IEEE 802.15.4 networks [34]. Indeed, Musaloiu *et al.* [35] showed that a IEEE 802.11 network may increase the packet losses of an IEEE 802.15.4 network up to 58%.

While slow channel hopping employed by IEEE 802.15.4-TSCH and WirelessHART tends to increase the reliability, packet losses keep on occurring [36]. As shown in Figure 1.1, both IEEE 802.11 and IEEE 802.15.4 standards have overlapping channels. Only channels 15, 20, 25 and 26 are interference-free from IEEE 802.11 devices.

Blacklisting approaches have been proposed recently to deal with external interference [37]. Typically, these approaches identify congested channels and avoid them for data transmissions. However, by blacklisting “bad” channels, the network capacity decreases, since less channels are effectively used. In addition, when collocated networks similarly employ the slow-channel hopping mechanism (e.g. IEEE 802.15.1), blacklist becomes ineffective, since all channels are used uniformly.

Furthermore, we are witnessing the raising of new low-power wireless network technologies for IoT. For instance, Long Range (LoRa) [38] and IEEE 802.11ah [39] have received much attention during the last years for long-range deployments. Sigfox [40] and Narrowband Internet of Things (NB-IoT) [41] are other low-power wireless technologies competing in the IoT market. Thus, we expect in the coming years even more wireless network technologies (and new devices) cohabiting an already crowded communication band, causing mutual interference.

Besides, the environment characteristics are time-variable and some external sources of interference may stop or start randomly. For instance, more people using wireless devices or an obstacle (e.g. moving object/person) may temporarily impact the wireless links. Variations on the climate conditions have also been reported to affect the reliability of wireless links [42]. Thus, link quality metrics have a crucial role to distinguish between short and long term variations.

A route reconfiguration is particularly costly for reservation-based MAC layers: the bandwidth in the previous route has to be deallocated and then reallocated in the new one. Naturally, the bandwidth reallocation requires transmitting control packets, increasing the network energy consumption. Worse, during rerouting, the network reliability may be compromised while nodes try to find new reliable paths.

Therefore, we highlight the following research challenge in this thesis:

Scientific challenge

How to ensure high reliability with upper bound delay, while uncontrolled and unpredictable factors potentially causing variations on the wireless channels?

1.3 Goal and main contributions

The main goal of this thesis is to investigate the performance of communication protocols used in IIoT applications when they are deployed in indoor environments. In particular, we focus on the identification of factors that impact the network reliability when the wireless links present time-varying conditions. More importantly, we make the network more robust by proposing improvements that ensure high reliability even when the radio conditions changes unexpectedly. Therefore, we rely mostly on real deployments for evaluation purposes. Although simulations provide a fast way to test an hypothesis before a time consuming implementation, they heavily depend on the accuracy of models [43], and tend to under-estimate the problems which may arise in practical scenarios [44]. Therefore, simulation results may diverge from the experimental results obtained from real deployments. Thus, we use intensively physical devices deployed on open testbeds.

The first contribution is to provide a reliable and efficient passive link quality estimator for low-power wireless networks. We highlight the main limitations of using classical link estimators when employing reservation-based MAC algorithms. Additionally, we demonstrate how the routing layer can exploit our new link metric to construct effective and stable routes. Our experiments on a large-scale platform highlight that our approach improves the convergence delay, identifying the best routes to the sink during the bootstrapping (or reconverging) phase without adding any extra control packet.

The second contribution is to investigate experimentally the performance stability of the 6TiSCH stack in collocated deployments. In this context, we consider an open indoor testbeds where concurrent experiments are executed in parallel. For being an indoor environment, our experiments are also subject to external interference originated from others wireless devices, such as Wi-Fi. Thus, our scenarios provide a typical environment for IoT-related systems and applications experiments, where multiple networks with different performance requirements have to cohabit in the same environment. We focus on some key metrics to exhibit the intermittent losses of guarantees (e.g. delivery ratio) under yet static conditions. Our results highlight that in the presence of radio oscillations, the 6TiSCH stack introduces frequent network reconfigurations to combat interference and provide high reliability. We perform a multi-layer analysis of the 6TiSCH stack identifying the main sources of instability and proposing solutions to address each of them.

The third contribution is to assess the relevance of using anycast transmissions to improve the network reliability. Anycast is a link-layer technique to improve the reliability when using lossy links. When employing anycast, multiple receivers are associated to a single transmission. That way, a transmission is considered erroneous when none of the receivers was able to decode and acknowledge it. Appropriately exploited by the routing layer, we can also improve the fault-tolerance by adding path diversity. Although existing works consider packet drops as independent events, we rather consider the effect of external interference that may equally impact devices in a given region. Here, we use a large dataset obtained from an indoor testbed to assess the gain of using anycast in real conditions. We also propose a strategy to select the set of forwarding nodes: they must increase the reliability by maintaining packet losses as independent as possible.

The forth and last contribution of this thesis is to analyze the attachment delay of mobile devices. Industrial environments now integrate mobile industrial robots to enable IIoT. Thus, the challenge consists in handling a set of mobile devices inside a static wireless network infrastructure. A mobile robot has then to join the network before being able to communicate. Here, we model this attachment delay by employing a Markov Chain that comprises both the synchronization and bandwidth reservation. We point out that the slow channel hopping mechanism and collisions introduce a large attachment delay, complicating the employment of reservation-based MAC protocols in mobile scenarios.

1.4 Structure of the thesis

This thesis is structured in seven chapters. In Chapter two, we present a detailed description of the main protocols and standards for IIoT, giving elements for understanding the rest of this thesis. Additionally, it provides an updated state-of-art on low-power protocols for IIoT. Chapter three describes our passive link quality metric to allow a precise identification of reliable paths. In Chapter four, we present our findings on the network performance stability in collocated deployments. In Chapter five, we analyze the relevance of using anycast at the link layer level. Chapter six, we present our model of the attachment delay of mobile devices. Lastly but not less important, in Chapter 7 we present our conclusions after three years of intense research on IIoT. Additionally, we point out some perspectives of future research directions.

Chapter 2

Related Work

Contents

The so-called Industrial Internet of Things (IIoT) is expected to transform our world, and in depth modernize very different domains such as manufacturing, energy, agriculture, construction industry, and other industrial sectors. IIoT relies on wireless technologies that are able to provide a high Quality of Service (QoS) for a plethora of industrial applications with high requirements concerning the latency and the network reliability [45]. In addition, field instruments incorporate sensors and low-power radio transmitters with IP connectivity, making their reading fully accessible through the Internet.

While traditional low-power radio applications have been assumed as delay-tolerant, real-time requirements is of utmost importance in IIoT deployments. In these environments, some specific requirements arise, especially regarding real-time communications [46]. Indeed, sensor data may trigger chain reactions from the industrial facility wherein it has been generated. Thus, delayed sensing data would be either useless or detrimental to the decisions made upon this monitoring system.

In addition, we expect IIoT to comprise a large set of nodes in vast geographical areas. To deal with large densities, the frequency reuse must be optimized by enabling multi-hop technologies while ensuring a network wide connectivity [47] and upper-bounded delay. In contrast to single-hop, where all nodes send their data directly to sink, a multi-hop communication increases the network coverage by allowing nodes to use their neighbors as relay points that forward the data toward the sink (Figure 2.1). Even in such complex topologies, we have to guarantee scalability (e.g. number of nodes, number of communication flows) and viability (e.g. limited computational and storage resources).

Therefore, this chapter introduces the main functionalities of communication standards and design characteristics of IIoT deployments. We then perform a thorough literature review, presenting an overview of the major solutions proposed so far for IIoT.

2.1 IIoT Characteristics

Unfortunately, the environment presents characteristics (e.g. heat, humidity, dust) that make wireless communication challenging. Indeed, multiple studies have demonstrated that environment conditions affect directly the performance of wireless links [42]. In addition, wireless links are also impacted by surrounding electromagnetic interference from other wireless devices [48] or physical obstacles (heavy machines, walls, etc.). Thus, IIoT solutions have consequently to be designed under various scenarios (e.g. unreliable fading channels, rate adaptation, dynamic topologies). Such operations require to take various aspects of wireless real-time communications into account (e.g. per-packet delay bound, timely-throughput requirement of each flow, heterogeneity of wireless channels).

2.1.1 Radio Link Properties

Low power radio networks are known to be lossy and packet losses are frequent [49]. Expected Transmission Count (ETX) [50] was one of the first metrics to estimate the link quality, counting the expected number of packets to transmit before receiving

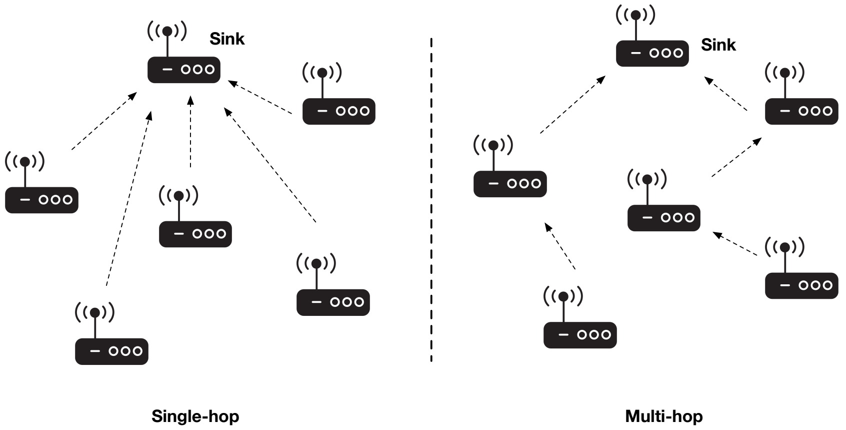


Figure 2.1: Single-hop and multi-hop wireless networks.

an acknowledgement. Unfortunately, the radio link quality is time-variant [51]. While very good quality links tend to be very stable, the medium quality links may exhibit significant long-term variations [42]. Constructing prediction models such as [52] may help to construct reliable solutions, tailored for the worst case scenario.

Radio links may also be asymmetrical, because of e.g. different transmission powers or noise floors. Asymmetrical links means that the delivery ratio in one direction is different from that in the other direction. Srinivasan *et al.* [53] consider a link as asymmetrical when the Packet Delivery Rate (PDR) in both directions differs by more than 40%. Therefore, considering link asymmetry when deploying IIoT solutions is crucial, since this asymmetry may impact directly the performance of higher layer protocols [54].

External interferences are very common in industrial environments. Electromagnetic interference caused by other higher power networks is reported as one of the main causes of packet corruption in industrial wireless networks [55]. To combat narrow-band noise, slow channel hopping has been proved to perform very well in industrial environments [28]. In addition, packet losses do not exhibit a perfectly random property. Some time periods exist during which all the packets are dropped in sequence: a link oscillates between the *good* and *bad* states. This so-called *link burstiness* is a consequence of obstacles or external interference [56], and has a strong impact on the reliability.

Wireless transmissions are also subject to *fading*, which is the variation of the attenuation of the radio signal. *Free Space Path-Loss* (FSPL) refers to the attenuation of signal over distance. In this case, the radio energy dissipates along the way, and a receiver is not able to decode correctly the packet due to weak signal. In addition, obstacles (e.g. walls, people) may attenuate the radio energy. This is known as *Shadow Fading.* Finally, the *Multipath Fading* refers to the case where the signal is reflected by nearby objects, making the receiver to hear multiple signals for a single transmissions, i.e. the transmission itself and the echoes.

2.1.2 Traffic Properties

The traffic intensity has naturally a strong impact on IIoT applications. Because the typical MTU is very small in the IIoT (e.g. 127 bytes), an application payload may be split (i.e. *fragmented*) into several Layer-2 frames. Thus, IIoT solutions has to provide guarantees by considering the packets in groups (of fragments). For instance, to achieve a given end-to-end delay, the *last* fragment must be received by the sink before the deadline. Similarly, respecting a 99% reliability for each fragment independently is not sufficient to achieve a 99% per flow reliability (losing one single fragment makes the reconstruction impossible, even if all the other ones have been received correctly).

We make distinction between two types of packets generation: *Constant Bit Rate (CBR)* and *Event-Based*. When employing CBR, each node generates its packets and maintains the inter-packet time constant. For instance, a network used in Structural Health Monitoring (SHM) applications collects data continuously at constant data rate [57]; On the other hand, with an event-based generation, a node generates a packet when it detects an event (e.g. a measure above a threshold, a physical stimulus). For instance, a fire alarm application may generate a bursty traffic when it detects smoke [58].

Besides, we make the following distinction among different types of traffic:

Uplink: a packet is generated by a node, and forwarded through a path to the sink. This type of traffic is typically used in data convergecast, where all generated data is addressed to the sink;

Downlink: the sink has to send a specific packet to one device;

Any2Any: a packet is generated for another device, and is forwarded along a path which may not pass through the sink. This type of communication involves typically a sensor which notifies an actuator;

Multicast: a packet is generated for a set of destinations, which all have to receive it;

Anycast: a packet has no specified destination and the receiver with highest priority acknowledges upon receiving it;

Flooding: a packet is forwarded from a node through all its outgoing links;

Broadcast: a packet has to be delivered to all logical neighbors. A broadcast is typically used to announce its presence via a `beacon`.

2.2 From best-effort to highly reliable MAC algorithms

The need for low power radio networks first led to low duty cycle approaches where nodes turn off their radio chipset most of the time to save energy. The Medium Access Control (MAC) has thus been largely investigated over the last years [59].

Classical contention access methods, such as the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), employ a random access for the medium, leading to two major drawbacks for real-time performance. The first one is that nodes need to keep their radio on more frequently in order to wait for possible incoming packets. To mitigate this overhead, some solutions rely on preamble sampling [60]. Typically, a transmitter announces via a *preamble*, its upcoming transmission. When a node detects a preamble, it must stay awake to receive the upcoming frame. However, these approaches suffer from the hidden terminal problem, being unable to provide strict guarantees for medium access. Secondly, in case of collisions, the transmitters have to wait for a random backoff value for the next transmissions attempt. Because of this *randomness*, there is no way of predicting when the next attempt will occur, eliminating the determinism. Thus, random access methods are ineffective when high reliability is a requirement.

Therefore, scheduling the transmissions represents the key element for achieving predictable performance. By carefully organizing all the transmissions, scheduling approaches remove most of collisions. In addition, it is also energy-efficient, since the nodes know precisely when they need to turn their radio on to transmit or receive data frames.

We detail the IEEE 802.15.4-TSCH mechanism next, since it includes most of the features of industrial wireless standards such as WirelessHart [25] and ISA 100.11a [26], and it supports both centralized and distributed scheduling algorithms.

2.2.1 Mechanisms of the IEEE 802.15.4 TimeSlotted Channel Hopping

IEEE 802.15.4-TSCH relies on a strict organization of the transmissions to make the protocol deterministic. TSCH relies on a slotframe, comprising a fixed number of timeslots (i.e. the slotframe length). This slotframe is cyclic and can be represented with a scheduling matrix (Fig. 2.2). In addition, the Absolute Sequence Number (ASN) denotes the number of timeslots since the beginning of the operation of the network. We have consequently a time reference, global to all the nodes.

A cell is defined by a pair of timeslot and channel offset denoting respectively its instant of transmission in the slotframe and the frequency it has to use for the transmission. A set of cells is allocated to each active link which has to forward a certain quantity of traffic. The slotframe is repetitive: the same cell is allocated in each slotframe to the same set of radio links.

Depending on the traffic forwarded through the link, two different types of cells are defined in TSCH:

dedicated cells should be assigned to a group of non-interfering radio links. These cells are represented as The transmitter does not implement in that case any contention resolution algorithm since it considers it has a *full* access. By allocating real-time traffic to dedicated cells, IEEE 802.15.4-TSCH provides a deterministic medium access. A dedicated cell can also be defined as *incoming* or *outgoing* cell. During an incoming cell, a node turns it radio in receiving mode, waiting for possible transmissions. Similarly, the node turns its radio in transmitting mode during an outgoing cell;

shared cells are assigned to a group of possibly interfering transmitters. Figure 2.3 demonstrates an example of the random access on shared cells. When a transmitter has a packet in its queue at the beginning of a shared cell, it

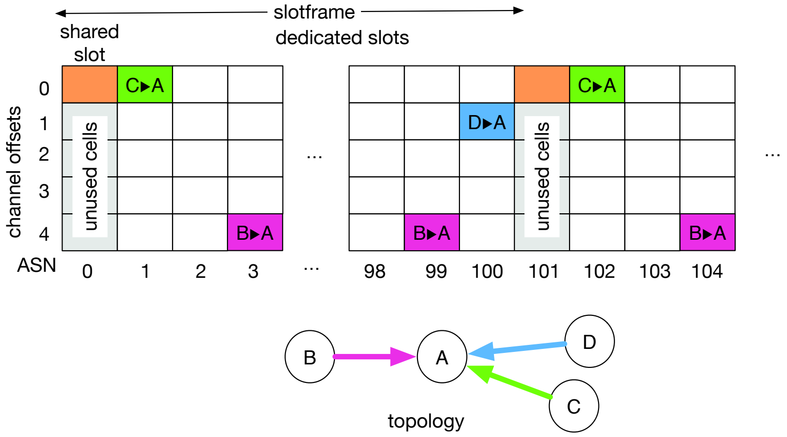


Figure 2.2: Structure of the slotframe in IEEE 802.15.4-TSCH.

transmits the packet immediately. If an ack is required but wasn't received, the transmitter considers that a collision occurred. In that case, it selects a random backoff value, and *skip* the corresponding number of shared cells. Because shared cells employ a slotted Aloha approach, they do not provide guarantee of packet delivery. Thus, these cells should be used only to transmit less sensitive packets, i.e. control packets.

During a shared cell, all nodes have to stay awake. In that case, the same cell should be allocated to all nodes. In dedicated cells, only the transmitter and receiver turn their radio on, while the other nodes keep their radio off to save energy. Therefore, a IEEE 802.15.4-TSCH network is able to present a very low duty-cycle, while providing high reliability.

Slow Channel Hopping for High-Reliability

The channel hopping mechanism adds channel diversity by using a different frequency for each transmission attempt, following a pseudo-random sequence. When a transmission on a specific channel fails, the next attempt will occur on a different one, probably with different conditions. Slow channel hopping is motivated by the fact that some channels tend to be more congested than others [61]. Thus, a slow channel hopping technique helps to improve the reliability in presence of external interference. and to combat fading in indoor environments [62].

As explained previously, a cell is defined by a pair of timeslot and channel offset. To support slow channel hopping, a node derives the actual frequency to use a the beginning of the timeslot with the following equation:

$$channel = F\left[(chOffset + ASN)\ \ mod\ NbChannels\right] \tag{2.1}$$

where $chOffset$ denotes the channel offset, ASN counts the number of timeslots since the network started, $NbChannels$ the number of physical channels (by default

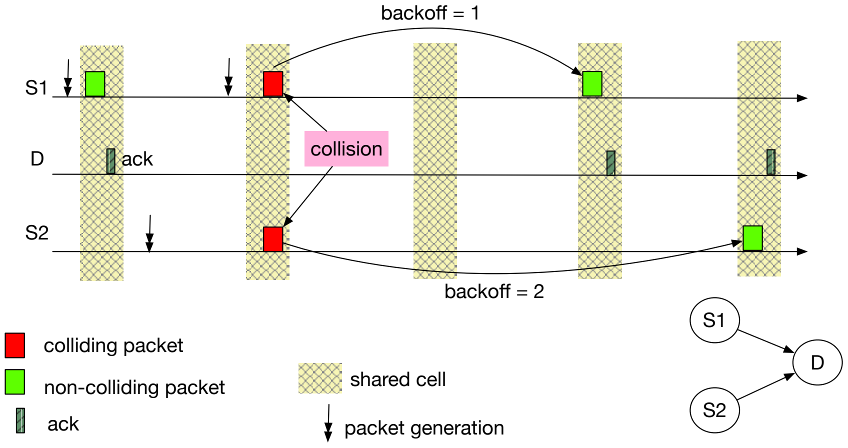


Figure 2.3: Random access on shared cells.

16 in TSCH), and *mod* the modulo operator. $F[]$ is a bijective function mapping an integer value into a physical channel.

If the slotframe length and the number of physical channels are mutually prime, this means that the same cell will be mapped to a different physical frequency in different slotframes. Using the whole diversity of the radio spectrum helps to improve the reliability.

Time synchronization

To exploit efficiently a schedule and to avoid collisions, the network has to be globally synchronized. Thus, each node selects a parent to create globally a synchronization tree, rooted at the sink. Then, a node readjusts continuously its clock when it exchanges packets with its parent.

More precisely, the transmitter sends its frames after a fixed offset from the beginning of the timeslot (Fig. 2.4). Thus, the receiver is able to compute the time difference between the expected and the actual time of reception. Since this difference corresponds to the clock difference, the receiver can readjust its clock after each reception. If no frame is received during a long time, keep-alive (KA) packets have to be transmitted to maintain the two nodes synchronized. The guardtime and the maximum clock drifts define the maximum period for these KA packets. Besides that, the offset for the beginning of the transmission is fixed even for shared cells: the receiver can readjust its clock after *any* reception.

Scheduling

All these approaches rely on a global scheduling matrix. In particular, a scheduler has to assign enough links for each transmitter, and its actual implementation was let unspecified. This authorizes to construct schedules with different objectives. For instance, a network may balance the load among the different nodes while another one would minimize the end-to-end latency.

Some solutions assume that a schedule may be pre-computed and installed in

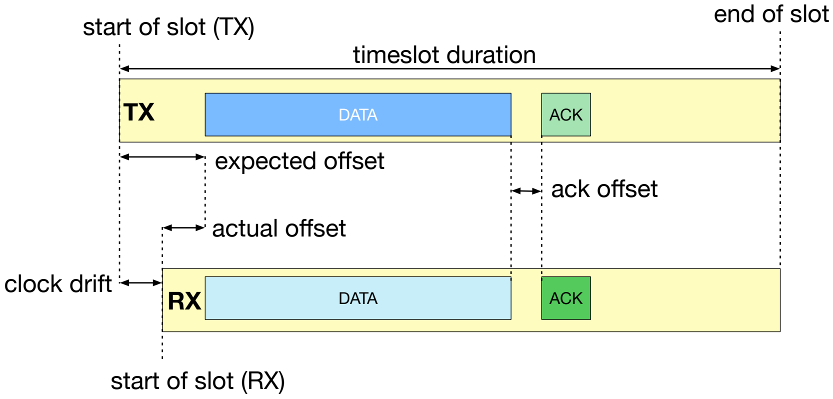


Figure 2.4: Resynchronization with contention-free timeslots – the receiver here realigns its clock with those of the transmitter.

each device. However, pre-configuring the devices may constitute a challenging task, specially in dense networks. Alternatively, the schedule may be changed dynamically, with an approach inspired from the Path Computation Element (PCE) [63]: new cells are reserved when a flow is inserted in the network.

Additionally, scheduling algorithms have to take into account the limited amount of available memory in embedded devices. In wireless multihop networks, each forwarding node has to listen to the medium during the incoming cells (rx mode), and can only transmit its packets during the outgoing cells (tx mode). Thus, a forwarding node has to buffer a packet between each incoming and outgoing cells. The buffer occupancy is also strongly correlated with the buffering delay: a longer delay increases the probability to create a buffer overflow in the forwarding nodes. If the buffer is too small, some of the packets have to be dropped: even if transmissions are perfectly reliable, the system as a whole is unreliable.

We present different scheduling algorithms in Sections 2.2.2 and 2.2.3.

Other standards

The IEEE 8021.5.4-2006 standard [11] mixes a Time Division Multiple Access (TDMA) and a random access to organize the transmissions. In the beacon-enabled mode, a coordinator maintains its neighbor synchronized by transmitting periodically a `beacon` (Fig. 2.5). After the reception of the beacon, any node can transmit a frame, adopting a random access approach until the end of the Contention Access Period (CAP). Only a few transmission opportunities may be reserved for a contention-free access – the Guaranteed TimeSlots. Finally, all the nodes can switch their radio off to save energy.

To cope efficiently with multihop networks, the beacons of the sleeping periods have to be carefully scheduled [64]. Some extensions propose to use a multichannel approach, where each coordinator receives a portion of time and a physical channel for its superframe [65]. However, the standard keeps on suffering from a burst of collisions at the beginning of the superframe [66], impacting very negatively the reliability.

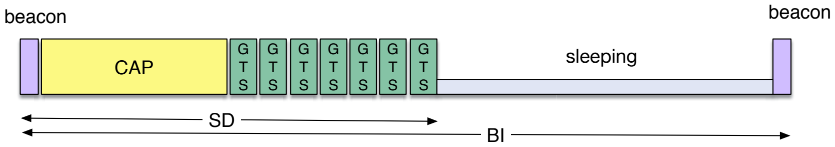


Figure 2.5: Superframe of IEEE 802.15.4-2006.

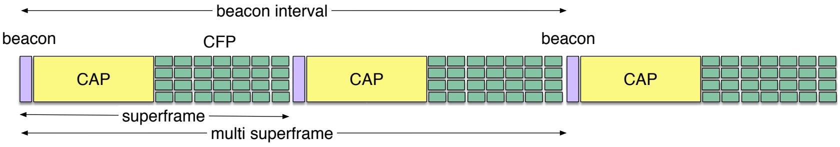


Figure 2.6: Superframe of IEEE 802.15.4-DSME.

The Low Latency Deterministic Network (LLDN) [67] mode is a mix between TSCH and IEEE 802.15.4-2006, and was tailored for time-constrained radio networks using a star topology. LLDN does not employ the slow-channel hopping procedure and it uses only a single frequency. In case of dense network, the standard suggests to create multiple networks operating in different frequencies. The superframe comprises 4 parts:

Beacon timeslot: the first timeslot in the slotframe and is reserved to the sink to transmit synchronization beacons;

Management timeslots: is optional, comprising 2 timeslots (downlink and uplink respectively) that are reserved only for exchanging control packets between the nodes and the sink;

Uplink timeslots: used by the nodes to send their data to the sink;

bidirectional timeslots: the bidirectional slots can be used by any node, sink and end-devices.

The Deterministic and Synchronous Multi-channel Extension (DSME) [24] mode extends IEEE 802.15.4-2006 which provided only seven reserved timeslots per superframe for real time traffic. DSME adopts a multisuperframe structure (cf. Figure 2.6): the superframe starts with a beacon so that all the receivers can re-adjust their clock. Then, the Contention Access Period (CAP) can be used by any transmitter, implementing a random access method, with a random backoff value before the transmission. Finally, the Contention Free Period (CFP) is located at the end of the superframe: a pair of nodes has a dedicated channel access and can transmit its packets. Beacons of all the coordinators have to be scheduled to avoid collisions [68].

WirelessHART extends the HART protocol [21] to allow wireless communication. The standard is based on IEEE 802.15.4-2006, but it defines its own medium access control (MAC). WirelessHART employs a supreframe structure, similar to the one illustrated in Fig. 2.2. With the slow channel hopping mechanism, WirelessHART presents resilience to external interference, providing additionally the possibility of blocking congested channels. The scheduling and routing are performed by a central entity, which constantly collects network statistics from the field devices.

Similarly, ISA 100.11a employs a Frequency-Time Division Multiple Access for medium access. While WirelessHART and ISA 100.11a share most of its principles in terms of scheduling, they present minor differences [69]. In particular, the schedule length in WirelessHART is fixed and contains 100 slots, of 10ms. In ISA 100.11a, there is no limit for the schedule length. Additionally, ISA 100.11a does not restrict the duration of each timeslot, the user has then to ensure that all devices employ the same duration. While WirelessHART allows only 15 different channels, i.e. the channel 26 is excluded, ISA 100.11.a allows all 16.

WirelessHART and ISA 100.11a target typically the same network architecture. In a simple way, the architecture mixes routing and non-routing devices, and a central manager that works like the "brain" of the network. The routing devices are placed in key locations to allow wide network connectivity. In addition, the central manager is in charge of constructing the schedule, and the routes. For this, all devices have to continuously transmit statistics to the manager for decision making. Both standards employ redundant routes to ensure that all transmissions are highly reliable. In this case, a node has always an alternative route when the primary fails.

2.2.2 Centralized Scheduling

We will describe here the scheduling algorithms that are built and maintained by a central controller. This kind of deployment is very common in industrial scenarios [70, 71]. As the controller has a global view of the network, its job consists in constructing an optimal schedule (concerning a criteria), while respecting a set of constraints (e.g. traffic, routes). Creating this schedule while considering all these aspects represents a NP-complete problem [72].

As pointed out by Tsitsiklis *et al.* [73], the centralized architecture is ideal in static networks where every property is known precisely. Indeed, a distributed approach can only use local information, and consequently may make sub-optimal decisions.

TASA [74] [75] is a multihop centralized scheduling algorithm for slow channel hopping. The algorithm gives a larger priority to nodes with more traffic to forward. This way, it tries to allocate bandwidth first to the most constrained nodes, using different channels for conflicting links. For this purpose, it uses a combination of matching and coloring algorithms to build an appropriate scheduling.

The authors of TASA implicitly assume perfect links conditions. Therefore, TASA_{rtx} [76] extends TASA functionalities to deal with non-perfect radio links. The algorithm calculates how many additional cells are needed in the schedule to satisfy the expected end-to-end PDR. TASA_{rtx} also considers the fragmentation of packets in order to transmit long messages over several slots. Similarly, Elsts *et al.* [77] propose an scheduling algorithm that deals with unreliable links by using shared cells for possible retransmissions. However, both algorithm assume that the link characteristics are time-invariant, which is not the case in most practical scenarios. Thus, they may fail to provide end-to-end reliability, when e.g. packets are dropped in bursts.

In order to allocate less slots and increase the network throughput, MOD-

ESA [78] considers a sink equipped with several radio interfaces. Having multiple interfaces helps to improve the performance and the scalability [79]. However, MODESA assumes that all nodes generate the same number of packets.

Some approaches consider scheduling the transmissions while blacklisting congested channels. MABO-TSCH [80] combines a centralized scheduling that excludes bad channels from the hopping sequence. The scheduler exploits a coloring problem where the nodes are sorted according to their degree in the graph. To be reactive, a pair of nodes decides locally the physical channels to blacklist. To estimate the channel quality, MABO-TSCH exploits the Stochastic Multi-Armed Bandits with 16 arms (one for each available channel). However, MABO-TSCH requires that nodes disseminate their local blacklist to neighbors periodically, complicating its applicability in higher density networks.

To handle mobility in industrial scenarios, Montero *et al.* [81] consider a centralized scheduler combining static and mobile nodes. In order to join (or rejoin) the network, the mobile node must be in receive mode to listen for advertisements sent by the static nodes. The advertisements include which timeslots the mobile node should use to send a joining request, forwarded by the static nodes to the sink. Finally, the sink replies with the list of allocated slots to the mobile device, valid until it changes its point of attachment. As highlighted in the experiments, mobility requires a short slotframe, and the handoff has still a strong impact on the performance, dropping packets until a new cell is allocated.

Finally, Munir *et al.* [82] calculate for every flow the number of slots to take into account the bursts of packet losses along the path to the sink. Its training phase consists in measuring the amount of consecutive failed transmissions ($\mathrm{B}_{\mathrm{max}}$) for each link. This information is crucial to determine the link burstiness level of each radio link. They trained their system over a 21 day period, which appeared as a minimum duration to obtain accurate statistics (e.g. longer training required for stationary nodes with high $\mathrm{B}_{\mathrm{max}}$). Then, the scheduler avoids the links that exhibit a link burstiness level $\mathrm{B}_{\mathrm{max}}$ larger than a given threshold. However, to the best of our knowledge, no method is able to predict systematically how long the training phase should last in order to collect statistically relevant measures to calculate the link burstiness level.

2.2.3 Distributed Scheduling

We focus now on algorithms that dynamically adjust the schedule to fulfill new communication requirements. The approaches are expected to deal efficiently with dynamic topologies and traffic. In most of the following approaches, each node is responsible to create a communication plan using only local information.

Using a distributed algorithms reduces communication overhead, as messages are transmitted only by the involved nodes. Guglielmo *et al.* [83] argue that distributed algorithms improve the scalability and the energy efficiency of the IIoT.

Accettura *et al.* [84], propose a decentralized version of TASA [74], DeTAS. At first, each node requests bandwidth from its parent. The parent node computes the amount of packets that it will receive from its children and its own traffic requirements and forwards this information recursively until it reaches the sink. The sink starts the allocation by scheduling the timeslots to receive the aggregated traffic from each child. To reduce the end-to-end delay, and the buffer overflows, DeTAS schedules alternatively the reception/transmission slots along the path to the sink. However, if a packet is lost due to a poor radio link quality, all the subsequent alternating slots scheduled on the parents nodes to forward this packet are wasted.

Similarly, the Completely Fair Distributed Scheduler (CFDS) [85] constructs the schedule alternating the transmission cycles for hierarchical networks. The leader of each cluster sorts its children according to their traffic demand. Each node reserves bandwidth using the Resource Reservation Protocol (RSVP) and Generalized Multi-Protocol Label Switching (GMPLS). CFDS works based on transmission cycles, where the most demanding node of each cluster is selected first by its parent. Each cycle begins after each leader schedules the transmission of the node selected in the previous cycle. The leader nodes also have their own transmission cycles, where they apply the same procedure with their own leaders. CFDS does not know a priori the traffic generated by each device. This information is collected in a distributed manner, aggregated along the tree rooted at the sink. However, considering a dynamic topology or traffic is not addressed in the original proposal. More specifically, mechanisms to notify the sink and to restart the schedule construction have yet to be proposed.

Distributed scheduling exploiting Reinforcement Learning has also been proposed to create traffic-adaptive solutions [86]. For every timeslot, each node executes a random action (transmit, receive or sleep) and receives a feedback. The feedback is positive when no collision occurs and when the node receives an acknowledgment. Upon positive feedback, the node will repeat this action during the slot in the next slotframe. Inversely, another action is selected randomly when the feedback is negative. Intuitively, the algorithm can accommodate new demands, since it keeps on executing the corresponding actions. However, the convergence is not guaranteed.

The Scheduling Function Zero (SF0) [87] was standardized by the 6TiSCH Working Group. The bandwidth estimation algorithm estimates the number of cells collecting the bandwidth requirement from each neighbor. It maintains the actual number of cells reserved with each neighbor equal to the amount of traffic to receive/transmit. When both quantities differ, SF0 engages a new reservation. SF0 also uses a hysteresis function to avoid oscillations in the allocation. In addition, SF0 keeps monitoring the used cells to detect when the packet delivery ratio drops below a given threshold, denoting a colliding cell. In such situation, the cells are reallocated *randomly* in the schedule table.

SF0 allocates the cells randomly, without knowing which cells are used in the neighborhood. Municio *et al.* [88] demonstrated that this strategy works well for star technologies, but the number of collisions increases quickly in multihop topologies. They propose the DeBraS approach, where each node has to piggyback its local schedule when broadcasting its Enhanced Beacons. This knowledge helps to reduce the amount of collisions when a node has to select a new cell for a flow.

Theoleyre and Papadopoulos [89] proposed a distributed scheduling for 6TiSCH networks. This approach allocates bandwidth while guaranteeing flow isolation: each application has its own bandwidth reserved along the whole path. The bandwidth estimation algorithm works similarly to SF0, but it directly uses the current queue length of the node, and the Expected Transmission Count (ETX) of each allocated cell to compute the number of cells to allocate/deallocate. The authors compared two scheduling policies: either the timeslot and channel offsets are selected randomly or it tries to reduce the end-to-end delay. In the latter case, the source tries to select the first available slots after the reception slots corresponding to the same flow.

6TiSCH minimal [90] relies on shared cells common to all the nodes, at the beginning of the slotframe. By default, 6TiSCH minimal recommends the usage of a single shared cell, this parameter being tunable in the Enhanced Beacons. The shared cells are inserted in the schedule as soon as the node receives an EB and joins the network.

Some algorithms rely on a pseudo-random sequence to define autonomously a *rendez-vous* for any pair of nodes. A rendez-vous consists in a pair of instant and channel, derived from a pseudo-random sequence, so that the transmitter knows when and through which channel it can send its packets. Orchestra [91] was an adaptation of this rendez-vous based approach for low power networks. The algorithm relies on TSCH, and defines three types of slots:

receiver oriented: the slots are dedicated to receive packets. Since several neighbors may exist, they correspond to shared slots;

transmitter oriented: the slots are dedicated for its transmissions. The node considers itself as the unique transmitter, and identifies the slot as dedicated;

broadcast: common shared slots are in charge of transmitting broadcast traffic.

Orchestra proposes to use a hash of the MAC address of the node to determine which cells should be dedicated for transmission/reception in each slot. Since any neighbor can apply the same hash function, the neighboring schedule can be directly computed without exchanging any additional control information. However, the number of transmission opportunities does not depend on the volume of traffic forwarded through a given radio link. Since the different nodes often do not have the same amount of packets to forward, this approach both wastes energy (some nodes do not use all their cells) and suffers from a poor network capacity (the most loaded nodes may not have enough cells).

More recently, the 6TiSCH Working Group has defined the Minimal Scheduling Function (MSF) [92]. Similarly as Orchestra, MSF employs the concept of *autonomous cells*, derived from the node MAC address. An autonomous cell is typically used for exchanging control packets, while advertisement packets are transmitted in regular shared cells. Thus, control traffic is isolated from advertisement traffic. Differently from Orchestra, MSF is traffic adaptive, i.e. the number of allocated cells depends on the number of packets exchanged between a sender and a receiver.

2.3 Link Quality Estimation

While the radio links exhibit very different link qualities in wireless networks, the routing protocol should select the shortest but most reliable paths [93]. Optimizing the service provided by the link layer to the network layer imposes to minimize the link layer transmissions required for a data packet to be correctly delivered (and acknowledged) by the sink.

2.3.1 Link metrics

Estimating accurately the link quality for low power lossy networks has been extensively studied in the literature [80]. The Received Signal Strength Indicator (RSSI) is one well used metric to evaluate the link quality [42]. The RSSI value is easy to obtain, since most recent radio chipsets already provide it. In the AT86RF231 [1] radio chipset, the RSSI is an 5-bit integer value, variating from -91 to 28 dBm. Unfortunately, RSSI and reliability are loosely correlated for medium link qualities [94], or close to the radio sensitivity.

Similarly, the Link Quality Indicator (LQI) measures the error in the incoming modulation of the received packet. LQI ranges between 0 and 127, where a low value indicates a good link quality. As RSSI, LQI was reported to have high variation for medium link qualities [95].

Differently from RSSI and LQI which are hardware dependent, the Packet Reception Ratio (PRR) estimates the link quality by computing the ratio between the number of acknowledged packets by the number of sent packets. However, its accuracy depends on the number of packets transmitted over a temporal window [96].

Because of the stochastic nature of radio transmissions, statistical estimators are required. A Window-Mean Exponentially Weighted Moving Average (WMEWMA) combined with the PRR is a popular estimator [97]. WMEWMA computes the average reception rate as:

$$WMEWMA(\alpha, w) = \alpha * WMEWMA + (1 - \alpha) * PRR \tag{2.2}$$

where α represents the importance of the historical data and w the time window. However, WMEWMA behaves poorly for medium link qualities [98].

The Expected Transmission Counter (ETX) [50] has been widely used in the literature. Let $PDR_{x \to y}$ denote the packet delivery ratio from x to y. ETX estimates the average number of transmissions before the emitter receives a link-layer acknowledgement:

$$ETX = \frac{1}{PDR_{s \to r} * PDR_{r \to s}} \tag{2.3}$$

However, ETX may create instabilities for long routes, because of its cumulative variations [99].

2.3.2 Active Monitoring

A node may estimate the link quality *actively* toward a neighbor, by transmitting probe packets. MoMoRo proposes to combine several metrics (ETX, RSSI and LQI) to estimate the link quality [100]. Bildea *et al.* propose to categorize the different links, discriminating good, bad and intermediate link qualities with a Gilbert-Elliot model [101].

The last version of Contiki (3.0) implements a probing method: each inactive neighbor is probed periodically to discover *better* next hops. Vallati *et al.* [102] verified experimentally that probes allow the nodes to select only the best links, and to reduce globally the packet losses. Unfortunately, probes are sources of energy

[1] http://ww1.microchip.com/downloads/en/DeviceDoc/doc8111.pdf

wastage and create collisions with other control packets. Besides, sending probes independently to each possible neighbor is too expensive for dense topologies.

Pradittasnee *et al.* proposed to use broadcast probes to reduce the overhead [103]: each node knows the amount of received probes to infer the packet delivery ratio. Even though broadcast probes induce lower costs (e.g. energy, processing) than unicasted ones, they still increase the probabilities of collisions with other control packets and resulting expensive retransmissions.

2.3.3 Passive Monitoring

Alternatively, passive approaches only use the existing traffic to infer the link quality. The monitoring process does not disturb the network (i.e. collisions and energy consumption).

Gomes *et al.* propose to introduce LQE nodes dedicated to traffic monitoring and link quality estimation [80]. Typically, LQE nodes are all the nodes which have to forward traffic. They use the RSSI and LQI values to infer the Packet Delivery Ratio (PDR) of the given links. Unfortunately, RSSI and PRR have been proved to be only loosely correlated in many situations [94].

LPL approaches use a preamble before each data transmission so that the receiver stays awake to receive the packet. Overhearing may be used to affine the link quality estimation, even if the node does not exchange packets with one of its neighbors [104]. In that case, a node must capture all the packets even if it is not the link layer destination. Then, it uses the RSSI of the received frame after having properly inferred the identifier of the transmitter to estimate the ratio of corrupted frames.

However, data packets are exchanged only with **active** neighbors. Thus, the link quality toward an inactive neighbor can only be estimated if the node overhears the traffic. Unfortunately, overhearing is expensive, and may even be impossible in multichannel (this neighbor may use a different channel offset during the same timeslot).

2.3.4 Hybrid Monitoring

Hybrid approaches try to combine both active (active neighbor) and passive (inactive neighbor) methods. Thus, probing packets are used only when no data traffic is available.

For instance, 4-Bit Link estimation mixes active (probe packets) and passive (ETX for data packets) criteria [105] so that probes are only used when no data packet is exchanged with a given neighbor. Probes are broadcast to mutualize the overhead: one single probe helps to refine the link quality toward *all* the neighbors.

2.4 Routing in the Industrial Internet of Things

IIoT networks present some harsh conditions for radio embedded devices. Physical obstacles and interferences from concurrent transmissions negatively impact signal propagation and the network reliability. The link quality therefore becomes unpredictable and may endanger the stability of upper layers. Because of this uncertainty and dynamics, routing becomes very complex in industrial deployments.

To construct an accurate schedule, the network needs to select the best route(s) for each flow and to allocate enough transmission opportunities to each node along the path to the sink. Estimating the link quality is of prime interest: selecting a suboptimal parent implies that many packets have to be retransmitted to be correctly received by the next hop. The routing topology is in other words inefficient and the incriminated nodes may quickly run out of energy.

Existing routing approaches, such as Open Shortest Path First (OSPF), were not designed for embedded devices with energy constraints. Thus, new routing protocols had to be designed considering the constraints of low-power devices. For instance, the IETF ROLL WG has defined the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [14] taking into account the characteristics and the constraints of low-power wireless networks. RPL has received much attention over the last years, and it is considered as the main routing protocol for the IoT [15].

Next, we explain the main mechanisms behind the RPL protocol.

2.4.1 Mechanisms of the Routing Protocol for Low-Power and Lossy Networks (RPL)

RPL is a IPv6 compliant routing protocol based on distance vectors and source routing for low-power and lossy networks (LLNs). The protocol was specified in the RFC 6550 [106] in 2012, but its design goals were first defined in 2010, considering industrial, commercial (building), home, and urban networks requirements [107]. Since them, RPL has attracted much attention from the academy and companies, turning it in as *the* standard for IoT [15]. Indeed, RPL has been used in numerous applications, from health-care [108] to precision agriculture [109].

RPL has been designed based on data collection protocols, such as the Collection Tree Protocol (CTP) [110]. The protocol targets specially convergecast traffic pattern, constructing a Destination Oriented Directed Acyclic Graph (DODAG) routed at the sink. A RPL instance may contain multiple DODAGs (Fig. 2.7), each one providing a different level of QoS. A node can join a single DODAG per instance, transmitting different types of traffic simultaneously. Multiple RPL instances is particularly relevant when different applications with different requirements have to cohabit the same infrastructure. For instance, Nassar *et al.* [111] consider multiple RPL instances to attend heterogeneous requirements in Smart Grid applications.

The RPL protocol is based on the concept of *ranks*. A rank represents the relative distance from the node itself to the DODAG root. In a general way, nodes with lower rank are closer to the root than nodes with higher ranks. The root is the device with the lowest rank in the DODAG. In Fig. 2.7, node S is the DODAG-root with rank x ($x < y < z$).

In addition, in a RPL DODAG, a node can have multiple neighbors but only one *preferred parent*. Therefore, a node send its packets towards the sink through its preferred parent. For instance, Node A is the preferred parent of the node C, and the sink S is the preferred parent of the node A.

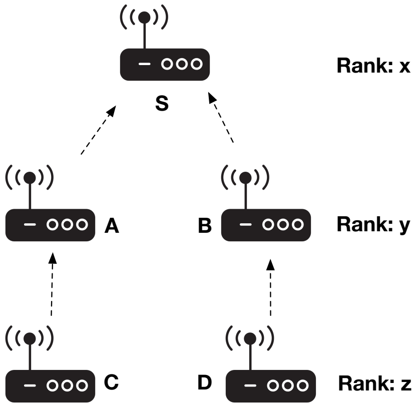


Figure 2.7: RPL DODAG routed at the node S.

Downward routing

Additionally, RPL supports downwards routing, making it possible to dispatch requests to all devices in the network. For downward routing, RPL defines two modes of operations: *storing* and *non-storing* modes (Fig. 2.8). In storing mode, nodes rely on routing tables containing the routing states of all nodes below them in the sub-DODAG. When receiving a packet, a node verifies in its routing table the path to the destination. On the other hand, non-storing mode relies on source routing: the sink is the only device aware of the routing topology. Thus, all packets go upwards to the sink before going to the destination.

Employ storing mode requires a lot of memory to keep a large amount of route entries. Thus, in dense networks it may be problematic to construct routing tables taking into account all relevant neighbors. Indeed, nodes have limited amount of memory, typically a few kB of RAM.

To improve the scalability, Eriksson *et al.* [112] propose rather better policies for adding neighbors in the routing table. RPL does not specify rules for adding new neighbors in the routing table. In fact, in current RPL implementations (e.g. OpenWSN, Contiki), a node adds new neighbors as soon as they are discovered, without proper evaluation. The limitation of this approach is that the routing table can keep *bad* neighbors, while excluding *good* ones, making it harder to scale the network up. Therefore, they propose to reserve dedicated spaces for: (i) nodes preferred neighbors, (ii) few other neighbors that are good candidates for upward traffic and (iii) a set of children nodes for downward. The criteria to add nodes in the routing table are based on the remaining space dedicated for each category and their distance to the sink. In addition, they rely on probing the neighbors periodically to keep fresh link qualities estimates.

Control packets

RPL employs four types of control packets to build and maintain a routing topology:

DODAG Information Object (DIO): aperiodic beacons that contain information to instruct nodes to join a DODAG. It defines the DODAG Identifier, the

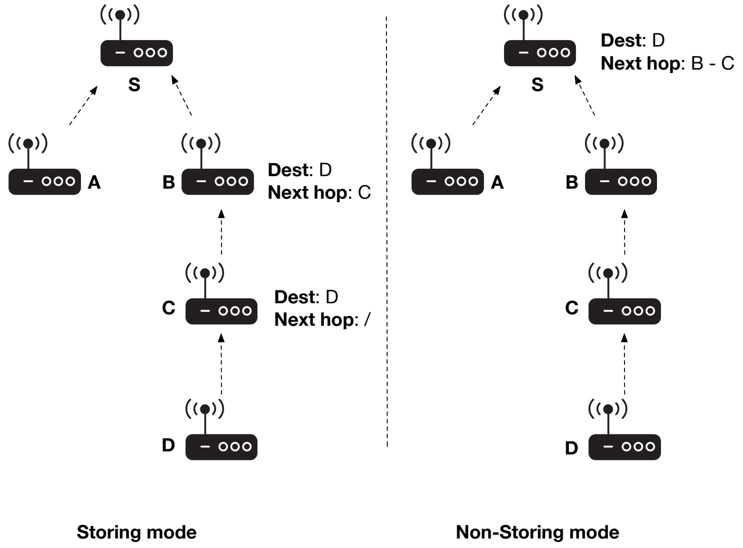


Figure 2.8: Storing mode and non-storing mode for downward routing.

Mode of Operation (MOP), instance identifier, etc. After receiving a DIO, a node joins the DODAG, keeping a list of neighbors in which it selects a preferred one, i.e. the *preferred parent*. Typically, the preferred parent is used for upwards routing. The frequency of DIO transmissions is given the Trickle Algorithm [113]. This algorithms ensures higher DIO transmissions when the network is unstable, i.e. frequent routing reconfigurations, and lower frequency otherwise.

DODAG Information Solicitation (DIS): instead of waiting passively for a DIO, a node may request a DIO by transmitting a DIS to one or more nodes. Typically, before joining the network, a node transmits the DIS in multicast, and in unicast when probing a particular neighbor;

Destination Advertisement Object (DAO): nodes build and maintain their routing tables by transmitting DAO upwards along the DODAG. In storing mode, the DAO is transmitted in unicast to the preferred parent. In non-storing mode the DAO is transmitted to the DODAG root;

Destination Advertisement Object Acknowledgment (DAO-ACK): this control packet is typically used for confirmation of route installation/removal after a DAO transmission. Both DAO and DAO-ACK are optional control packets.

Routing metrics

Routing metrics are cost values used by routing algorithms to determine the best path to a destination. For instance, Hop Count yields the number of hops between a sender and a destination. It is a simple routing metric, since is quite easy to obtain it once the routing topology is known. When minimizing the number of hops in the network, the routing topology tends to be more stable with a low rate of reconfigurations [99]. However, this metric may privilege long distant links, potentially

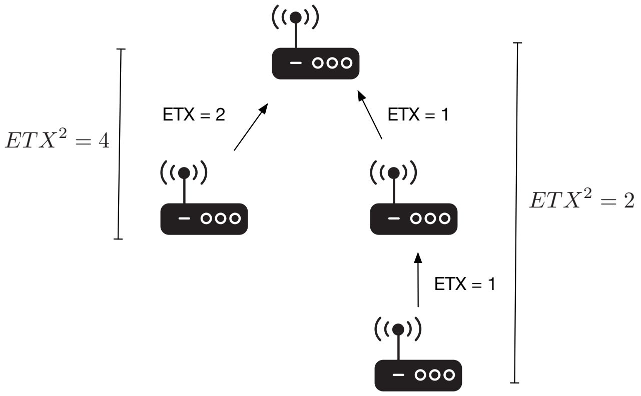


Figure 2.9: Path cost computing using ETX^2.

presenting poor performance [114].

ETX can also be employed as routing metric to indicate path quality. In this case, the path ETX is defined as the sum of the ETX for each individual link in a given path, or:

$$ETX(path) = \sum_{i \in path} ETX_i \tag{2.4}$$

This metric outperforms hop count, since it takes into account the accumulated delivery ratio of all links in the path.

ETX^n was recently proposed by Duquennoy *et al.* [115]. While ETX minimizes the aggregated number of transmissions, ETX^n tries to privilege reliable links by computing the n_{th} power of the ETX value. Figure 2.9 depicts an example ($n = 2$) where two radio links with a perfect reliability (ETX $=$ 1) constitute a better path than a direct link with a PDR of 50% (ETX $=$ 2), although both paths have the same accumulated ETX value. However, ETX^n implicitly penalizes more intermediate links than ETX. For instance, a link for which the PDR changes only by 0.1 (80% → 70%), exhibits a ETX^2 variation of 0.5. Because of this penalization, ETX^n may increase the routing instability.

Objective functions

An *Objective Function* (OF) specifies how the rank is computed. In particular, it defines how different routing metrics should be used to compute the node's rank.

There are two objective functions standardized for RPL: (i) the *Objective Function 0* (OF0) [116] and (ii) the *Minimum Rank with Hysteresis Objective Function* (MRHOF) [117]. When using OF0, a node computes its rank from the rank of its preferred parent. More precisely, the rank of a node is computed by adding a scalar value to the rank of its parent. The standard defines that this value can vary with a ratio from 1 (excellent) to 9 (worst acceptable) to represent link properties. OF0 does not restrict any metric to compute the rank: the metric is left to the implementation.

On the other hand, MRHOF selects routes with the smallest cost, while using hysteresis to reduce churn for small metric changes. The hysteresis avoids un-

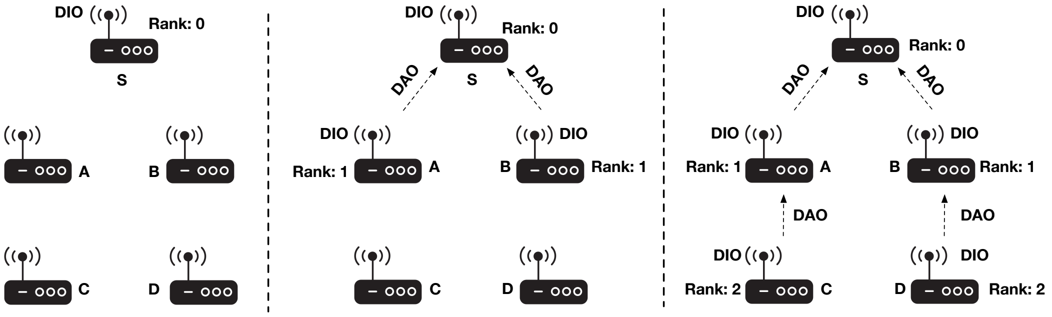


Figure 2.10: The DODAG construction using OF0 and hop count as metric.

necessary routing reconfigurations, which comes with an increasing control traffic. MRHOF requires the use of addictive metrics, such as ETX or hop count.

DODAG construction

Figure 2.10 illustrates the DODAG construction employing OF0 and hop count as routing metric. During the bootstrap, the DODAG root S is the only device that broadcasts DIO to announce the network. Thus, the root S has rank 0. At this moment, the Trickle Algorithm ensures a higher transmission rate of DIO to allow fast topology discovery.

Eventually, nodes A and B receive the DIO transmitted by the root and they join the DODAG. Since they are 1 hop away from the root, they compute their rank as 1. Now, nodes A and B start broadcasting their own DIO, increasing the reach of the network by allowing a multi-hop topology. At the same, they start to transmit DAO in unicast to the root to construct a downward routing. Similarly, nodes C and D receive the DIOs transmitted by the nodes with rank 1: node C receives from A and node D from B. They join the DODAG with rank 2 and again they start transmitting their own DIO and DAO packets to their preferred parents.

Once the routing topology is stable, the Trickle Algorithm reduces the rate at which nodes transmit DIO, to save energy. In particular, the period is doubled when no inconsistency is detected. Otherwise, the period is reseted and nodes start transmitting DIO at higher frequency to allow a fast update of the DODAG.

Routing stability

Network stability represents a key challenge in radio environments. In particular, a route has to be constructed, selecting the most stable nodes/links, so that their link quality fluctuations are minimized [118].

Clausen *et al.* highlighted the presence of routing instabilities when using RPL under real deployment [119]. Because of the oscillations in the link quality, a node tends to change its parent aggressively, impacting negatively the convergence.

Kermajani and Gomez [120] have conducted a sensitivity analysis of RPL. The parameters have been proved to have a strong impact on the convergence delay. They propose to use DODAG Information Solicitation (DIS) to accelerate the convergence: each node already attached to the DODAG has to reply with a DIO. However, these control packets would also use shared cells in TSCH, increasing the number of collisions.

Other works have focused on load balancing approaches to achieve higher end-to-end delivery rate and stabler topologies. Kim *et al.* [121] avoid overloading a forwarding node by introducing the queue size to RPL objective function. The approach minimizes the occurrence of packet losses and reconfigurations due to buffer overflow. Additionally, the network stability can also be increased by adding an adaptive mechanism that adjusts the transmission power to minimize the occurrence of collisions [122] .

Iova *et al.* [99] evaluate the impact of different link quality metrics on the routing stability. They highlighted the existence of a tradeoff between stability and performance. When using the hop count metric, RPL operates steadily with low frequency of reconfigurations but performs badly. On the other hand, with ETX and LQI, the network presents higher frequency of reconfigurations and higher end-to-end reliability.

Alvi *et al.* propose rather to change the objective function, combining the ETX and min hop metrics [123]. However, this method is quite conservative: a node tends to be stuck with the same parent, even if a better alternative choice exists.

2.4.2 Multipath routing

Multipath routing consists in a loop-free topology between the nodes and the sink, where all nodes have at least two neighbors able to forward its packets toward the sink. A node gives higher priority to transmit on the *primary path*, which is the path with least cost from the node itself to the sink. The other paths, or the *backup paths*, are used after a transmission failure on the primary path. The backup routes provide a certain diversity and the network is able to recover after a temporary or permanent fault.

Distributed

Some works aim at extending RPL to allow the construction of a routing topology with redundant paths for IEEE 802.15.4-TSCH networks.

Shi *et al.* [124] focus also on distributed routing by proposing the Distributed Graph Routing and Scheduling (DiGS). They employ the RPL protocol to construct the routes, and each node selects two preferred parents based on their ranks and on the RSSI level. Next, they employ an autonomous scheduler, similarly to Orchestra [91], to reserve bandwidth without control packets. However, RSSI and the delivery ratio are not correlated for intermediate link qualities, which can lead to inaccurate decisions.

Similarly, Papadopoulos *et al.* [125] rely on the RPL protocol to parallelize transmissions over two paths. The approach provides a packet elimination mechanism to drop duplicated data based on a sequence number. However, the approach increases exponentially the number of allocated cells, since it reserves unicast links to each possible path. Consequently, the energy wasted by idle listening may be significantly higher, impacting the network lifetime.

Minet *et al.* [126] compare 3 redundant schemes, namely Disjoint, Triangular and Braided patterns (Fig. 2.11). These 3 patterns increases the reliability by creating

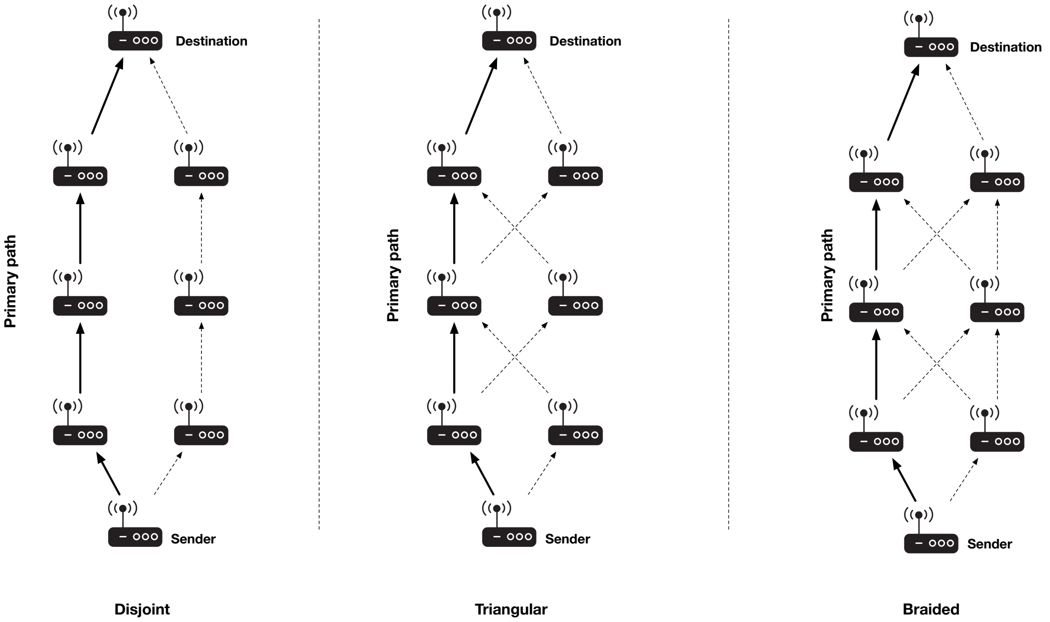


Figure 2.11: The three redundant patterns evaluated by Minet *et al.* [126].

an alternative route in addition to the primary one built by RPL. The number of eligible nodes that can forward the packets depend on the scheme used. Their experiments demonstrate that the Braided pattern ensures higher path diversity, providing higher reliability at the cost of higher energy consumption.

Additionally, Jenschke *et al.* [127] propose multiple algorithms to select a backup parent in RPL. The *Second Best ETX* simply selects the best candidate with lowest rank among all neighbors, excluding the preferred parent. The *Common Ancestor* selects a neighbor as alternative parent if this node and the preferred parent share a common ancestor, similar to the Triangular pattern illustrated in Fig. 2.11. Finally, the *Non-Common Ancestor* selects a neighbor that does not share any ancestor with the preferred parent, creating disjoint paths. The algorithms prioritize neighbors according to their ranks in the selection. However, their solutions rely on overhearing and probing, making them less energy-efficient.

Graph routing

Differently from the works described previously, graph routing solutions require the use of a central controller to constructs the routes. To do that, the controller continuously collects network statistics from all devices, builds the routes and then distribute to all devices. Most graph routing solutions have been proposed for the WirelessHART standard.

Dang *et al.* [128] allocate additional timeslots based on graph routing for retransmissions. When the communication with a neighbor fails, the node tries to retransmit on a different channel. If the transmission still fails, the node tries to transmit to the other parent. Unfortunately, the authors did only present a theoretical analysis, without evaluating the algorithm with any simulation or experiment.

A similar approach is adopted by Zhang *et al.* [129]. The authors aim to minimize the delay by allocating retransmissions slots only to the nodes on the optimal route. The optimal route is given by the algorithm ORMGR [130], a routing algorithm for WirelessHART networks that constructs a connected graph from the source to the destination, by using statistics of packet losses. As in [128], the authors evaluated the algorithm only analytically.

Lee *et al.* [131] schedule dedicated links for a redundant communication path. They exploit the max-min algorithm to optimize the number of dedicated slots necessary to establish an alternative route once the link fails. The algorithm properly allocates the slots in order to meet the delay bounds after the route switching.

Similarly, Wu and Lee [132] reserve sections of the superframe to allocate the transmissions for an alternative path when the link between two nodes fails. Each section is used by a different set of nodes that are equidistant to the sink. The length of these sections (number of reserved slots) depends of the traffic rate and the number of nodes that may use them.

Modekurthy *et al.* [133] base their solution on the Bellman-Ford algorithm to construct the routing graphs. The nodes transmit periodically their routing table, containing the cost of reaching other nodes in the network. To cope with the limited size of the packets, they consider a network clustering, so nodes only broadcast the routing information considering only other nodes in its own cluster and cluster head from other clusters. However, transmitting periodically the routing tables increases both the energy consumption and the probability of collision with other traffic. Additionally, since the packet length in IEEE 802.15.4 networks is restricted (127 bytes), this solution does not scale well for dense networks.

2.4.3 Opportunistic forwarding

Opportunistic forwarding approaches exploit the broadcast nature of wireless communication to increase the number of potential forwarders for a given transmission. The key objective of opportunistic forwarding is to increase the probability of packets making progress in the routing tree by allowing nodes to overhear outgoing transmissions. Because of opportunistic forwarding increases the spatial diversity, it improves the network throughput and the reliability when compared to traditional unicast routing techniques [134].

Opportunistic forwarding relies on a consensus mechanism, where the receivers decide among themselves *the* node that will forward the packet. Otherwise, a collision occur when multiple receivers forward the packet at the same time. ExOR [134], for instance, piggybacks a list of candidates prioritizing them according to their distance to the destination. A receiver with lower priority only forwards the received packet if candidates with higher preference do not transmit first.

In duty cycle networks, nodes keep their radio off most of the time to save energy. Thus, duty cycling limits the number of potential forwarders that may overhear an outgoing transmission, assuming that the nodes are not synchronized.

Asynchronous MAC

Asynchronous MAC solutions do not require global time synchronization, each device follows its own schedule to turn its radio on or off. Figure 2.12 depicts a packet forwarding employing an asynchronous MAC solution. Node *A* keeps transmitting

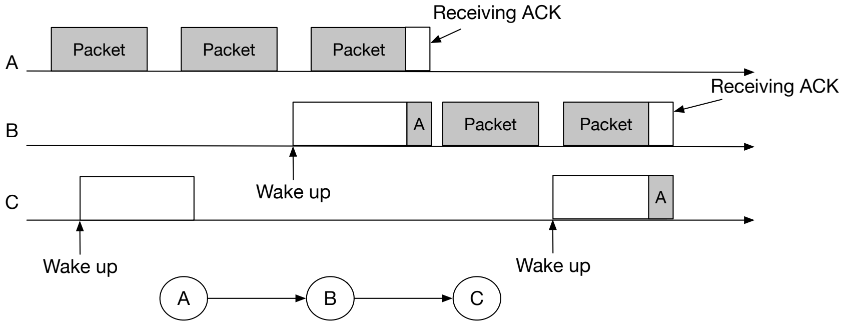


Figure 2.12: A forwarding scheme with asynchronous MAC.

until B turns its radio on for receiving the incoming packet. The same goes for the transmission from B to C. Therefore, it is necessary that the schedules of neighbor nodes have overlapping periods so the transmission can occur.

Even with the lower spatial diversity, opportunistic forwarding represents a viable solution for asynchronous MAC protocols. Landsiedel *et al.* [135] propose the Opportunistic Routing in Wireless sensor networks (ORW) to improve delay in low-duty cycles networks. ORW considers all neighbors as potential forwarders, even if they present unreliable links. Thus, instead of waiting for a particular node to wake-up, ORW allows a node to send its data to the first awoken parent. The solution also counts with a coordination algorithm in case of multiple receivers. In this case, the affected nodes “flip a coin” to decide which neighbor will forward the data.

Similarly, Duquennoy *et al.* [136] employ downward and upward opportunistic routing to RPL. They employ ORW for constructing opportunistically the upward routes, along a gradient. For downward, they define the concept of *routing set*, i.e. all children nodes in a sub-DODAG. Since the nodes elect themselves opportunistically for data forwarding, there is no need of using traditional routing tables. Thus, a node forwards downward a packet if it wakes up at the right moment to receive it and by checking if the destination is in the routing set.

Mohammad *et al.* [137] propose the Oppcast, an opportunistic routing scheme that exploits spatial and channel diversity at the same time. Oppcast differs from the previous solutions as it is received based. More precisely, potential receivers broadcast periodically probe messages announcing their availability for forwarding. At the same time, a transmitter waits for a probe packet before attempting to transmit. In addition, Oppcast employs the Fast Channel Hop (FCH) scheme to mitigate the impact of external interference. The FCH scheme ensures that both transmitter and receiver are on the same channel to avoid deafness.

Synchronous MAC

Opportunistic forwarding has also been proposed for synchronous MAC protocols. ISA100.11a [26] implements the concept of *duocast*, where two receivers are assigned to the same transmission. The receiver with the highest priority, first acknowledges the packet upon correct decoding. When the primary receiver fails, the secondary node has the opportunity to acknowledge the packet, and to place it in its forwarding queue. Thus, we have immediately a fallback solution to reduce the transmission delay, and the required number of retransmissions.

Huynh *et al.* [138] investigate the interest of opportunistic forwarding in IEEE 802.15.4-TSCH networks. They demonstrated that opportunistic scheduling helps to increase the network capacity, even exploiting a Gilbert-Elliot channel model. To schedule accurately the transmissions, their Approximate Dynamic Programming (ADP) approach relies on the knowledge of the channel state, complicated to obtain in practice.

More recently, Hosni *et al.* [139] employ a greedy policy where a node selects up to k possible forwarders based on the routes constructed by RPL. They demonstrated that picking the best parents (higher PDR) represents an optimal strategy when neglecting the overhead for notifying the transmitter of successful decoding.

However, all these approaches work ideally only when packet losses are uncorrelated. Unfortunately, the literature has proved so far that packet losses exhibit a high spatial correlation [42] In particular, the κ metric tries to estimate the correlation of packet losses among different links [140]. This metric is used to decide which path to choose, and how to implement network coding to improve the reliability.

2.5 Standardizing IPv6 over reservation-based MAC layers

The fast evolution of IoT has caused the deployment of more and more applications. This massive growth can be explained by the easy acquisition of inexpensive embedded devices and advancements in low-power wireless networks, making IoT deployments cost-effective. At the same time, vendors have pushed their own specific solutions for IoT employing proprietary protocols. However, this diversity of protocols comes with a drawback: the lack of compatibility between them. Thus, standardization is necessary to ensure that all communication softwares from different vendors can interoperate.

Although there are a wide range of standards for IoT [141], until 2013, no standard focused specifically on Industrial Internet of Things. In particular, a standard focusing on an high deterministic performance *and* IPv6 connectivity targeting low-power wireless networks still needed to be proposed. On the one hand, IEEE 802.15.4-TSCH brings high reliability to low-power wireless networks by scheduling the transmissions. On the other hand, 6LoWPAN [12] defines mechanisms to make it possible transmitting IPv6 packets on low-power wireless links. There was still a “gap” to be filled connecting both IEEE 802.15.4-TSCH and the network layer. For that reason, the 6TiSCH Working Group was formed in 2013 to propose a standardized stack taking into account the requirements of IIoT deployments.

Similarly, the Deterministic Networking (DetNet) [142] working group was created in 2014, focusing on providing deterministic performance at network layer level. DetNet is currently defining a Software Defined Network (SDN) architecture, providing a network model that allows scheduling operations fully orchestrated by a central controller [143]. For this, the working group is defining a set a signaling elements that will allow the construction of reliable paths between a node and the controller. This includes data models for collecting network topology statistics by the controller and a packet replication scheme to ensure high reliability. The DetNet working group collaborates closely with other IETF groups, such as the 6TiSCH.

2.5.1 6TiSCH Working Group

The 6TiSCH IETF working group aims to define protocols to bind IPv6 (i.e. 6LoWPAN, RPL) to a reservation based MAC layer (i.e. TSCH). 6TiSCH makes a distinction between the protocol which defines how to negotiate the cells, i.e. the 6P protocol [31], and the algorithm deciding how many cells to allocate in the schedule (a.k.a the Scheduling Function). The solution is very flexible since any scheduling algorithm may be practically implemented: a new Scheduling Function has just to be defined and interfaced with 6P. Thus, 6P may work either in a centralized manner (e.g. a node asks a Path Computation Element for new cells to use) or in a distributed manner (e.g. SF0 [87] decides how many cells to allocate based on the local traffic demands).

In addition, 6TiSCH introduces the concept of *track*, representing a set of cells along a route from the source to the destination [144]. Thus, a scheduler is able to allocate a given bandwidth (amount of cells) for a set of end-to-end flows. More importantly, by selecting different cells for different flows, 6TiSCH enforces traffic isolation [89]. Traffic isolation is particularly relevant when multiple applications have to share the same wireless infrastructure. Furthermore, 6TiSCH allows nodes to reserve regions (chunks) of the scheduling matrix for their own purpose [145]. When a node needs to allocate additional bandwidth, it schedules cells that are in its chunk. However, because a chunk is allocated to a node (and not a link), we have to over-estimate the interfering area: a timeslot can be used with any neighbor. A localized mechanism to detect collisions and to re-allocate another cell of the chunk dynamically still need to be proposed.

6TiSCH also defines a minimal configuration, i.e. *6TiSCH-Minimal*, [90] to achieve basic interoperability. Any 6TiSCH-compliant device should implement this mode of operation. In this mode, all devices wake up synchronously during *shared* cells. These cells are typically used by default to transmit control packets, i.e. Enhanced Beacons (EB), DODAG Information Objects (DIO) or 6P packets. To cope with collisions, 6TiSCH-Minimal employs the Slotted-Aloha protocol [146]. Because 6TiSCH-Minimal uses only best-effort cells, it tends to present low reliability [147].

2.5.2 6P protocol

The 6top Protocol (6P) [31] defines a set of procedures in which two neighbors negotiate dedicated cells. Typically, 6P specifies a set of high level operations that are used by the nodes during the negotiation. For instance, the ADD command is used for bandwidth request, while $DELETE$ for remotion. 6P defines also commands in which a node can request information about the schedule of a given neighbor. The $COUNT$ command retrieves the number of cells allocated, while the $LIST$ command return the list of cells in used in this neighbors. Finally, the $CLEAR$ command is used to remove all allocated cells of a particular node in the neighbor schedule. The CLEAR command is typically used when a node changes its preferred parent and it deallocates its old cells in the previous parent.

When two neighbors negotiate for the first time, they may rely on common shared cells for transmitting 6P packets, since they have no other common cells to

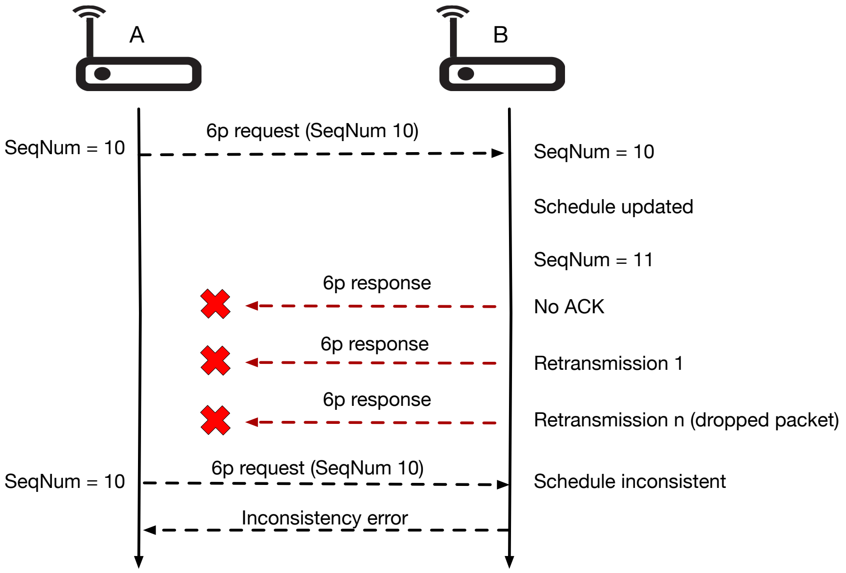


Figure 2.13: An example where two neighbors have inconsistent schedules.

bootstrap a negotiation. Naturally, 6P packets can collide with other control traffic, delaying the negotiation. To cope with this delay, some solutions consider employing autonomous scheduling [92, 91] to accelerate the bandwidth allocation/deallocation.

In a general way, 6P is based by default on a two-way handshake, where the inquirer sends a request (*6P request*) to a receiver (e.g. preferred parent) node and waits for its confirmation (*6P response*). A three-way mechanism has been considered by Duy *et al.* [148]. In this case, the authors define a new *6P confirmation* message, where the inquirer confirms the receiving of the response.

Additionally, 6P defines the concept of *transaction*, which it starts when a node requests to add/delete/reallocate one or more cells with one of its neighbors. A transaction ends when the cell(s) are effectively added/deleted/reallocated in the schedule of both nodes, or when the transactions fails after a specified timeout.

Because of the links lossy nature, the 6P protocol relies on mechanisms to ensure the schedule consistency between a node and its parent. Typically, a 6P sequence number is maintained per link, incremented after every modification triggered by any of the two corresponding nodes. In particular, their schedules are considered inconsistent if the two nodes have a different 6P sequence number.

Figure 2.13 shows a typical scenario to illustrate this inconsistency creation. When the node B receives the request from node A, B will update its schedule and will return a response back to A. Thus, B increments its 6P sequence number. Because of lossy links, the reply may be dropped when the link layer triggers too many retransmissions. In this case, A never receives the reply from B and has to use the previous sequence number: its next request has a lower sequence number, which is interpreted as schedule inconsistency by the 6P protocol. However, the 6P protocol does not define how to proceed after the detection of inconsistency, the scheduling algorithm is in charge of handling it.

2.6 Experimental Research for IIoT

Given the critical nature of industrial applications, the validation of new protocols and mechanisms becomes a crucial step for any IIoT deployment. Indeed, we must ensure that the performance requirements will be met before deploying the network in real-world conditions. An inaccurate evaluation can lead to unexpected results, potentially impacting the industry economic activities, or even putting people in unsafe conditions. Typically, IIoT protocol designers evaluate their solutions by employing theoretical analysis, simulations or physical deployments on open testbeds.

2.6.1 Theoretical Analysis

Theoretical analysis consists in obtaining a mathematical derivation of a given problem to obtain a mathematical solution for it. It may involve approximations and simplifications of the problem for acquiring mathematical closed form expressions.

In our perception, Markov Chain [149] has been a well accepted mathematical system for theoretical analysis in IIoT. It is based on probability rules that make the modeled system to transit from one state to another. The memoryless property is particularly relevant, as the current state of the system depends only on the previous one. Therefore, the time in which a system remains in a given state is exponentially distributed, making it possible to estimate the active state at a given time.

Although theoretical analysis may demonstrate that the performance guarantees are met, it is usually complicated to obtain mathematical expressions for most of the problems [150].

2.6.2 Simulations

Simulations provide a fast way to test an hypothesis before a time consuming implementation. Indeed, many aspects related to hardware functionalities (e.g. drivers compatibility, malfunctioning components) are abstracted. Additionally, it is relatively easier to reproduce experiments through simulations, using the same parameters and models is enough to achieve the same results. The availability of different models for radio propagation and interference also allows the channel conditions to be approximated to those expected in real deployments. Therefore, simulations have been extensively used for performing experiments in low-power wireless networks [151]. In our perception, open-source projects, such as Cooja [152] and OpenSim [153], are among the favorites simulators used in IIoT studies.

Although simulations provide a simple and controlled way for performing experiments, it presents some limitations. In particular, simulations heavily depend on the accuracy of propagation models [43], and tend to under-estimate the problems which may arise in practical scenarios [154]. Indeed, low-power wireless links are time-variant, and their reliability are highly affected by complex environment variables (Sec. 2.1.1). Thus, taking into account different sources of wireless links degradation would increase considerably the complexity of communication models, requiring more computation time.

Additionally, simulators employ a simulated clock, which advances equally for all devices in the experiment. Usually, embedded devices employ quartz-based crystals as timing source due their low-cost. However, these crystals are very susceptible to frequency drift, causing clocks to lose accuracy over the time. Therefore, nodes have to resynchronize frequently, since an efficient duty cycling requires clock synchronization [155]. Unfortunately, clock resynchronization usually requires nodes to exchange packets, impacting on the network energy consumption.

Therefore, we believe that simulations can not entirely capture the complexity of real networks deployments. For that reason, we have decided to conduct an experimental research during this thesis.

2.6.3 Testbeds

Open testbeds give access to a pre-deployed infrastructure, simplifying the deployment of experiments. Indeed, most testbeds provide full remote access to the devices, allowing researchers to read their inputs/outputs, measure energy-consumption and network-related metrics. The major advantage of using testbed is to allow researchers to test their solutions with real hardware under real-world conditions. For instance, an indoor testbed deployed on a workplace with several people using wireless devices may provide a representative environment to measure the impact of external interference on the performance of IIoT solutions.

Many open testbeds are available to the research community with different radio characteristics [150]. In indoor testbeds, the infrastructure is typically deployed in universities/scientific institutions buildings, usually sharing the same environment with concurrent wireless technologies. In addition, in indoor deployments is not rare to have walls and people acting as obstacles, attenuating the signal strength. On the other hand, outdoor testbeds are deployed in open areas, subject to sudden climate variations [156].

In a general way, each testbed have inherent radio characteristics that may differ from the others. In particular, indoors testbeds can use different types of wall materials with different thickness, or the infrastructure may be placed close to offices being more subject to external interference. Even the same testbed can present different radio conditions depending on the period of experimentation. For instance, conducting an experiment during a regular working day may have different results when deploying it during the weekends. Thus, before performing experiments on testbeds, its is important to study in deep the associated characteristics of the radio conditions for the considered facility.

Next, we introduce some testbeds that are frequently used for performing low-power wireless networks experiments.

FIT IoT-LAB

FIT IoT-LAB is a large-scale open indoor platform that provides an infrastructure for testing wireless sensors devices and heterogeneous communicating objects. It is also a shared platform with potentially multiple concurrent experiments. The facility offers quick experiments deployment, along with easy evaluation, results collection and analysis. In addition, the platform provides a variety of microprocessor architectures (Cortex-A8, STM32, MSP430) and different wireless chips (CC2420, AT86RF231).

The testbed is spread at six different sites across France with over 2700 available devices, making it the largest open low-power wireless remote testbed today

[157]. In details, Inria Grenoble (640 devices), Inria Lille (293 devices), Inria Saclay (264 devices), ICube Strasbourg (400 devices), Institut Mines-Télécom Paris (160 devices) and CITI Lab Lyon (29 devices). Additionally, the platform provides different topologies and environments. For instance, the testbed in Strasbourg isolates all devices in a single room [158], while in Lille the nodes are deployed over there floors across the Inria building [159]. A global networking backbone provides power and IPv6 connectivity to all nodes.

Typically, all deployments in FIT IoT-LAB are subject to interference from higher power networks (e.g. Wi-Fi), interference from concurrent experiments and moving obstacles (people). Therefore, FIT IoT-LAB provides a realistic environment for IoT-related systems and applications experiments. For that reason, we decided to perform our experimental research on it.

Indriya

Indriya is a large-scale wireless sensor network testbed deployed at the National University of Singapore. Its objective is to provide a public, permanent framework for development and testing of sensor network protocols and applications [160]. In addition, the testbed inherits many of its components from the popular MoteLab, deployed at Harvard. Using a web interface, users can have full control of their experiments, including upload and monitor theirs jobs remotely and in real-time [161]. The testbed has been available for researchers around the world since 2009.

The testbed counts with 127 TelosB motes, equipped with different sensors modules. All nodes are deployed over three floors, including laboratories, tutorial rooms, seminar rooms, study areas, and walkways. Due to the difficulties to make system upgrades and to provide maintenance, a more recent version has been considered. Indriya2 has been designed to address the limitation of the previous deployment, including the capability to support multiple platforms (SensorTag CC2650 and SensorTag CC1350) and wireless technologies (Bluetooth Low Energy and IEEE 802.15.4g) [162].

FlockLab

This platform corresponds to a low-power wireless testbed, managed by the Computer Engineering and Networks Laboratory at the Swiss Federal Institute of Technology Zurich [163].

FlockLab employs a three-tiers architecture that increases the flexibility for managing the experiment. The first tier consists of a wireless sensor network, or the system under test. Each individual node is attached to a more powerful device (a.k.a. observer node), consisting in the second tier. The observer provides means for extracting data and statistics from the low-power node itself. Finally, all observers are connected to a dedicated server, which manages them all.

The testbeds support different sensors platforms (Tmote Sky, OpenMote) and wireless technologies (IEEE 802.15.4 and LoRa). Actually, the testbed consists of 27 observers distributed across one level of the ETZ-building, including offices, hallways, and storerooms.

CorteXlab

The testbed described previously are deployed inside buildings, offering little or no external interference control. Therefore, is very challenging to reproduce experiments in such uncontrolled environment. In this context, the CorteXlab [164] testbed employs nodes in a shielded room, isolated from outside interference. This isolated deployment allows researchers to have a full control of the radio channel characteristics.

In addition, CorteXlab comprises heterogeneous platform that deals with cognitive radios. Cognitive radio is an adaptive and intelligent technology that can automatically change its transmitter parameters based on interaction with its environment [165]. CorteXlab has been defined to re-use the network infrastructure employed in IoT-LAB to integrate Software Defined Radio (SDR), offering a remotely accessible development platform for Cognitive radio.

2.7 Summary

In this chapter we provided guidelines to contextualize the research conducted in this thesis. First, we exposed the main IIoT characteristics, highlighting its predictable performance requirement and the challenge of attending such requirement in lossy wireless networks. Next, we performed a literature review of scheduling algorithms for highly reliable low-power wireless networks. We classified them according to the key characteristics of the scenarios they are targeting (e.g. dynamic vs. static traffic, lossy versus ideal links, mobile vs. static topologies). Third, we described the main procedures for routing in IIoT. We gave a special attention to RPL, since it has become the main routing standard for IoT deployments. In addition, we discussed a number of works that focus on graph and opportunistic routing, and why they can improve the network reliability. Fourth, we discussed the main monitoring approaches for link quality estimation in low-power wireless networks. Fifth, we exposed the 6TiSCH standard and its specifications to bind IPv6 on top of IEEE 802.15.4-TSCH. Finally, we discussed the main techniques for performing IIoT experiments. In particular, we compared both simulations and real deployments, justifying our decision of conducting an experimental research during this thesis.

In the next chapter, we start presenting our main contributions of this thesis.

Chapter 3

Passive Link Quality Estimation for 6TiSCH Networks

Contents

Estimating the link quality is crucial for building efficient routing topologies. In particular, radio links with high Packet Error Rate should not be exploited since they are less energy-efficient (more retransmissions are required) and they negatively impact the reliability. Indeed, Righetti *et al.* highlighted the need of an accurate link estimation before constructing a schedule: A mis-estimation negatively impacts the RPL convergence [166].

Unfortunately, estimating the link quality is not trivial. Many approaches rely on active probing to evaluate the link quality toward all the neighbors. While the probing period can be adapted dynamically to reduce the overhead [167], control packets are still required to quickly react to changes.

Worse, active probing is very sensitive in 6TiSCH. Using shared cells for the probes leads to many collisions, even with L2 and L3 control packets. Thus, some propositions rely on reserving a dedicated cell for *each* neighbor to send the probes (e.g. [166]). Such strategy tends to be expensive and suboptimal, though: could we reach the same objective without relying on additional control packets?

Here, we propose a purely passive approach for 6TiSCH, where a node identifies the best possible parents without testing individually every unicast link. To select the most accurate preferred parent, a node *ranks* its neighbors by the amount of advertisement packets received from each of them and chooses one of the top ranked ones to forward its packets. To the best of our knowledge, this work is the first to argue that a ranking of the candidates is sufficient: a node does not need to estimate *'exactly'* the link quality through a neighbor. We verify experimentally that this metric may constitute a good estimator even if the broadcast packets are subject to collisions.

Contribution

1. We propose a passive approach where a node can identify good neighbors without unicast probes. More precisely, we *rank* different possible parents using only the broadcast rate instead of estimating expensively the Packet Delivery Ratio (PDR). This metric can be used for both bootstrapping and the DODAG reconfiguration;

2. We analyze statistically the size of the observation window in order to minimize the reactivity time while identifying accurately the best candidates parents;

3. We integrate our passive ranking method in the 6TiSCH stack, to achieve an higher stability by avoiding *'blindly'* preferred parent changes;

4. We perform an experimental evaluation on the FIT IoT-LAB to confirm our hypothesis regarding the correlation of broadcast and acknowledged unicast receptions rate. We highlight that a node is able to safely identify one of its best parents, and to construct close to optimal routes.

3.1 Problem Statement

We need to identify good links to construct efficient routes. In particular, a node N has to *estimate* the link quality toward:

- its **active neighbors**, with which it exchanges data packets. Typically, an active neighbor is a node for which it forwards the packets, or to which it sends its own traffic, a next top toward the sink. A passive method is easy to implement: it is sufficient to measure the Packet Delivery Ratio;

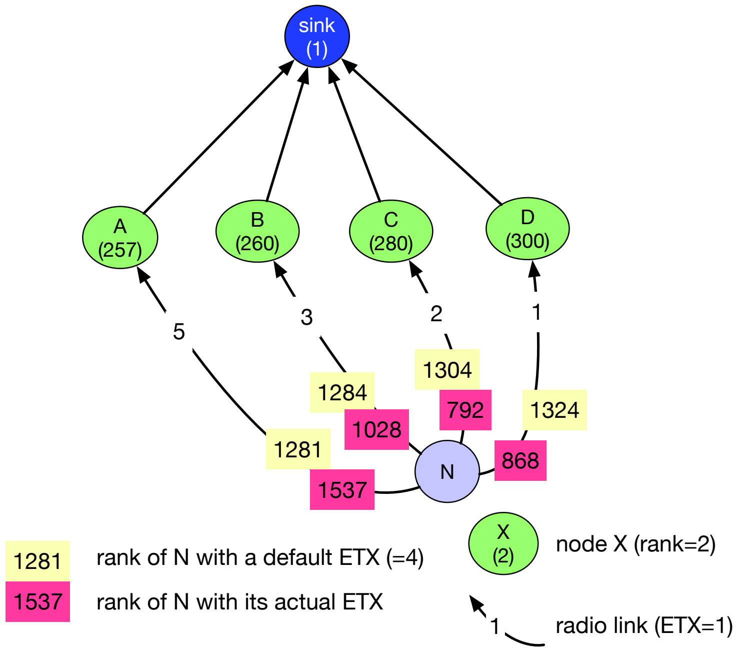


Figure 3.1: Convergence of RPL when using an initial default link metric.

- its **inactive neighbors**, with which no unicast packet is exchanged. In 6TiSCH, no dedicated cell is reserved to these neighbors, and no data packet is available for a passive measurement. Nevertheless, an inactive neighbor may be selected as preferred parent when the radio link qualities change or if the primary parent crashes.

3.1.1 Motivation

We assume that RPL is used for routing, and we use the objective function OF0 to compute the rank, based on the ETX of the links[2]. 6TiSCH minimal [90] does not recommend any default ETX value for inactive neighbors. The OpenWSN implementation[3] uses a default link cost of 4, i.e. the ETX toward an inactive neighbor is assumed to be 4.

A node N computes its own rank from those of its preferred parent P:

$$rank(N) = rank(P) + MinHopRankIncrease * ETX_{N \to P} \tag{3.1}$$

with $rank(S)$ denoting the rank of the node S, `MinHopRankIncrease` is a constant (by default equal to 256), and $ETX_{N \to P}$ denotes here the ETX value from N to P.

Let us consider the topology depicted in Figure 3.1. Since all the neighbors are considered initially as *inactive*, all the ETX values are assumed to be equal. Thus, N picks as preferred parent its neighbor with the lowest rank. After its association, N sends some packets to A and is able to refine its ETX estimation to reflect the actual link quality. Thus, its rank is updated from 1281 ($= rank(A) + 256 * 4$)

[2] We use here the default parameters for OF0 [116]

[3] `http://www.openwsn.org/`

to 1537 ($= rank(A) + 256 * 5$). Next, N detects that another neighbor (B) would provide a lower rank: N will change its preferred parent to B. If the actual ETX value is superior than the default ETX, N will probe iteratively each neighbor.

Inversely, using a too large default ETX is also suboptimal. In Figure 3.1, N will not select C or D as preferred parents although they provide a better path to the border router. Indeed, their rank with the default value would be 1304 and 1324 respectively. Since their default ETX is too large, these nodes will never be selected, without any opportunity to re-estimate more accurately the link quality.

The problem becomes even trickier to handle with temporal variations, very common for this kind of scenario.For inactive neighbors, the link metric was evaluated a long time ago and does not reflect the current quality. De facto, these neighbors will never be considered again to serve as preferred parent, except if the current one crashes or its link quality becomes very bad (i.e. ETX > 4).

For higher network density, a node may limit the number of neighbors to be included in its neighbors table. In such scenario, the nodes exclude periodically from their neighborhood table their worst neighbors. Later, these neighbors might be added back with the default link cost until they are probed again. With this inclusion/removal, a node may consider again a bad neighbor when a parent change is required.

3.1.2 Challenges

A naive approach would consist in probing each radio link individually to measure the Packet Error Rate. However, such an approach is practically inapplicable:

- each neighbor must be probed independently, generating a large volume of control packets. Such an estimation phase would waste too much energy as it should be executed continuously to detect radio link quality changes;

- if the probes use the shared cells, they may collide with other sensitive packets such as EB and routing packets. Possibly, such method would disturb the routing and synchronization protocols, thus preventing the network from converging properly.

We have to tackle the following challenges:

1. **Inactivity**: the link toward all the nodes must be estimated. In a 6TiSCH network, a node only communicates with its preferred parent and its RPL children [168], while its other neighbors are inactive;

2. **Passiveness**: the link quality should be estimated passively to reduce the energy consumption;

3. **Reactivity**: the estimation must handle link quality variations, and should be able to recover when the link quality toward a neighbor significantly decreases;

4. **Stability**: the estimator should be tolerant to short-terms variations to avoid excessive parents switching;

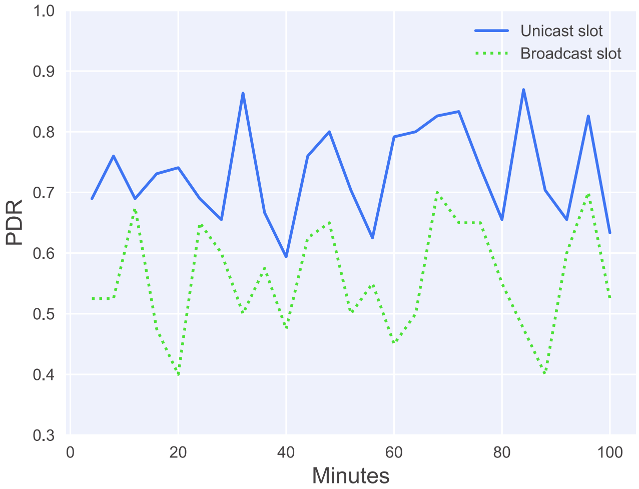


Figure 3.2: Difference between the PDR of unicast and broadcast slots between two nodes.

5. **Multichannel**: we have to accurately estimate the link quality even when multiple channels are used. In particular, the different channels may exhibit a very different PDR [169], thus distorting the estimation when only a few packets are used to compute the PDR;

6. **Broadcast accuracy**: broadcast packets use shared cells (with collisions) while data packets use dedicated (contention free) cells. Thus, the PDR of both types of cells is very different and transforming the broadcast PDR into an unicast PDR cannot be made easily.

   For instance, Figure 3.2 illustrates the evolution of such different PDR, as obtained in shared and dedicated cells with a star network of 11 nodes (experimental setup described in Section 3.2.1).

3.1.3 Scenario & Assumptions

We focus here on a convergecast multihop network, since bidirectional traffic is still not efficiently supported by 6TiSCH and RPL [170]. Figure 3.3 illustrates a typical topology: the plain lines represent the uplinks selected by the routing protocol (RPL in our performance evaluation). Each node selects a preferred parent, to which it sends all its unicast traffic.

To be received correctly, the EB and the DIO have to be transmitted during the shared cells, when all the nodes are awake. Thus, a node continuously receives broadcast packets from its neighbors. Since the broadcast packets are never acknowledged, no backoff is used before the transmission, and the control packets may collide. Figure 3.3 depicts the outgoing transmissions of node E. The broadcast packets are received by B,D, C and F, while unicast packets are received by B only.

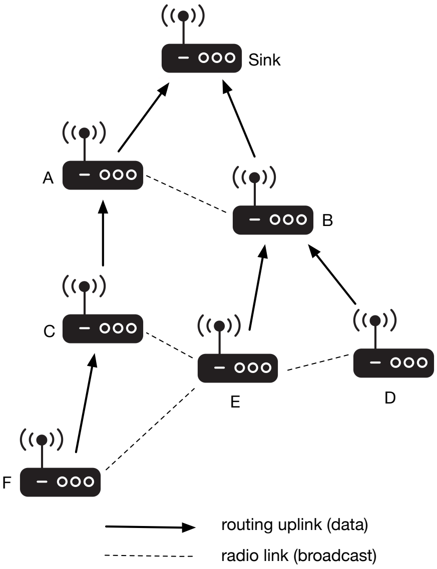


Figure 3.3: A multi-hop network with its uplinks and downlinks.

Our hypothesis is that there is a correlation between the reception rate of advertisement packets from a given neighbor and the link quality it would provide as a preferred parent. In other words, if the broadcast rate of $C \rightarrow F$ is larger than those of $E \rightarrow F$, the unicast PDR is assumed to be higher for the radio link $F \rightarrow C$ than for the link $F \rightarrow E$.

As mentioned in Chapter 2, a TSCH slotframe is composed of shared and dedicated slots. They are used in our scenario in the following way:

1. shared cells are used to transmit EB or DIO packets (broadcast);

2. dedicated cells are only used to forward the data and DAO packets (unicast toward the border router).

3.2 Estimation of unicast link quality from broadcast rate

A broadcast packet may not be received correctly because of:

Collisions: broadcast packets have to be transmitted during the shared cells, i.e. with contention, so that all the neighbors are awake to receive them;

Signal-to-Noise Ratio (SNR): a bad link quality implies some packets are not received by some of the neighbors (statistically with the worst SNR).

However, unicast packets are protected in dedicated cells and do suffer only from the physical Packet Error Rate (PER). We investigate here how the broadcast rate can allow to identify good links to use for unicast transmissions.

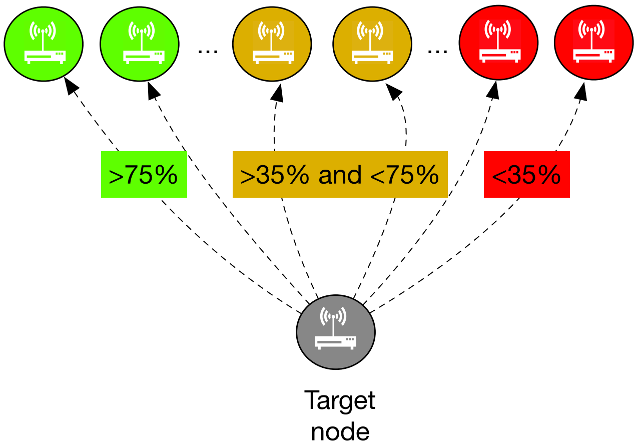


Figure 3.4: Topology with different links qualities.

3.2.1 FIT Iot-LAB & Hardware

We present now our experimental campaign over the FIT IoT-LAB platform. We employ the A8-M3 motes, based on ARM3505 (ARM Cortex A8) combined with a STM32 micro-controller and a AT86RF231 radio chipset, providing an IEEE 802.15.4 compliant PHY layer. We also execute OpenWSN that implements the full 6TiSCH stack (i.e. IEEE 802.15.4-TSCH, 6P, SF0, 6LoWPAN and RPL).

We implemented our proposal in OpenWSN, where nodes register the number of broadcast packets sent by their neighbors. To control more finely the experiments, and to mimic radio links with different qualities, we pre-installed a schedule at the compilation time (without collisions for dedicated cells). Our modifications are freely available on GitHub [4].

3.2.2 Measurement

We use here the multi-path topology depicted in Figure 3.4. The radio link quality depends coarsely on the distance to the target node. We focus on a specific node (hereafter designated as *target node*) which sends its packets periodically to each neighbor at a time.

We compute in a centralized manner a schedule so that:

1. the shared cells are used to transmit the enhanced beacons (EB) and the RPL's DIOs. The target node tracks all the broadcast packets received from each of its neighbors;

2. a dedicated cell is reserved with each neighbor. This cell is used to transmit unicast (data) packets to compute the unicast Packet Delivery Ratio (PDR). The target node would expect to obtain this PDR if it chooses the corresponding neighbor as next hop toward the sink.

Reserving a dedicated cell with each neighbor is required here to measure practically the unicast PDR. In a real deployment, the target node would reserve dedicated

[4] https://github.com/rodrigoth/openwsn-fw

Table 3.1: Default values used in the correlation and ranking analysis.

Experiment	Duration	24 hours
	Topology	Multipath
	# of nodes	11
	Testbed side	Grenoble
TSCH	Slotframe length	31
	# of shared cells	1
	Timeslot duration	15 ms
	EB period	15 s
	Schedule policy	fixed
CoAP	CBR	1 pkt/sec
RPL	DAO period	60 s
	DIO period	10 s
Radio	802.15.4 channels	11-26
	Transmission power	-4 dBm

cells with its preferred parent only. We use the default values for the different parameters, as depicted in the Table 3.1. Each experiment lasts 24 hours, logging approximately 132,000 broadcast and 140,000 unicast packets [5].

3.2.3 Correlation Factor Analysis

First, we quantify the correlation for each neighbor (n) between:

1. **unicast PDR**: the ratio of unicast packets transmitted to n and for which the transmitter receives an acknowledgement;
2. the **broadcast rate** (BR): the number of broadcast packets received from n during a certain time window.

We employ here the Pearson metric to compute the correlation between the unicast PDR and the broadcast rate. All the symbols used in this work are depicted in Table 3.2. The Pearson coefficient (Equation 3.2) is a linear correlation factor commonly used in statistics to measure the linear correlation between two input variables. It is defined formally by:

$$r_{x,y} = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)} \tag{3.2}$$

with $\sigma(X)$ denoting the standard deviation for the stochastic variable X, and $cov(X,Y)$ the covariance of the variables X and Y. This coefficient ranges from -1 to 1, where -1 indicates a perfect negative correlation and 1 indicates a perfect positive correlation. The correlation of 0 means the absence of any correlation between X and Y.

[5] Our dataset is available at `https://github.com/rodrigoth/ewsn2018` and can be freely exploited by the community.

Table 3.2: All the symbols used in this work.

Symbol	**Description**
n	n^{th} neighbor node
w	length of the observation window (time during which the different metrics are measured and averaged)
$EB(n)$	Nb. of Enhanced Beacons (EBs) received from the neighbor n
$DIO(n)$	nb. of DIOs received from the neighbor n
$ack(b \rightarrow a)$	Number of acks received from node b by node a
$packets(a \rightarrow b)$	Number of unicast packets sent to node b from node a
r	Pearson correlation coefficient
ρ	Spearman ranking correlation coefficient
f	Fisher coefficient
R_u	Unicast rank (i.e. ranking obtained when the neighbors are ordered by their unicast PDR)
R_b	Broadcast rank (i.e. ranking obtained when the neighbors are ordered by their broadcast reception rate)
$PDR(R_u, v)/PDR(R_b, v)$	PDR of the node in the v^{th} position in the unicast/broadcast ranking

We aim to investigate how much time is required to identify a correlation. We adjust consequently from 1 to 10 minutes the observation window w during which we measure the unicast PDR and the broadcast rate. Because the PDR and the broadcast rate are stochastic variables, we need a sufficient observation window to have an accurate estimation. A small w means that the network can quickly react to changes, by identifying a significant change in the value of the stochastic variables.

We focus now on the approach used to calculate the correlation coefficient between the number of broadcast packets received and the dedicated cell reception rate. We divide the dataset in portions of w minutes. During a given portion, we compute the number of broadcast packets sent by a neighbor n (eq. 3.3) and received by the target node, likewise the PDR from the target node to this neighbor (eq. 3.4).

$$broadcast\ packets(n) = \sum [EB(n) + DIO(n)] \tag{3.3}$$

$$PDR(a, n) = \frac{\sum [ack(n \to a)]}{\sum [packets(a \to n)]} \tag{3.4}$$

With the results obtained from Equation 3.3 and 3.4, we calculate the Pearson coefficient r for every portion of time contained in the dataset (eq. 3.2). We obtain finally a set composed of all Pearson coefficients:

$$C = \{r_a, r_b, ..., r_{z-1}, r_z\} \tag{3.5}$$

Before calculating the average correlation, we apply the Fisher transformation to all the elements in C (eq. 3.5), which gives us the set F (eq. 3.6). This transformation reduces the bias of the average when working with repeated measurements [171].

$$F = \{f_a, f_b, ..., f_{z-1}, f_z\} \tag{3.6}$$

where f_i is calculated using the Equation 3.7.

$$f_i = 0.5 * ln\left(\frac{1 + r_i}{1 - r_i}\right) \tag{3.7}$$

To calculate the average of all correlation coefficients, we apply Equations (3.8) and (3.9), which consist in taking the average of all normalized coefficients and then converting them back to r scale.

$$\bar{F} = \left(\frac{f_a + f_b, ..., +f_{z-1} + f_z}{length\ of\ F}\right) \tag{3.8}$$

$$\bar{r} = \frac{e^{2\bar{F}} - 1}{e^{2\bar{F}} + 1} \tag{3.9}$$

Figure 3.5a shows the average correlation for different window sizes (w). We observe that when the size of w is small (less than 2 minutes), the correlation between the two variables is weak. Indeed, the collisions represent a stochastic variable. Since broadcast transmissions are not so frequent, the broadcast PDR is mis-estimated, highlighting a loose correlation. This phenomenon is even exacerbated by the slow channel hopping approach: the PDR may depend heavily on the physical channel. With a small number of packets, the broadcast transmissions are in this case not uniformly distributed among all the channels, creating a statistical bias.

To calculate the ideal size of w, we define the value $\bar{r}$ above which the correlation is assumed to be strong enough. Evans [172] considers a threshold value of 0.8 is an accurate choice. Indeed, for $r = 0.8$, 64% of the variation between the two variables can be explained by their relationship. The other 36% are due to external factors or sampling error.

In the evaluated topology, we note a strong correlation in a multichannel network from 3 minutes onwards. This result indicates a high positive correlation between the PDR and the number of received broadcast packets.

3.2.4 Neighbors Ranking Correlation

We aim here to evaluate the ability of the broadcast rates to identify a correct *ranking* of neighbors. Intuitively, the neighbor with the highest broadcast rate should also provide the best unicast quality.

We use the Spearman coefficient to evaluate the quality of such ranking. This coefficient (Equation 3.10) is a nonparametric measure between two ranked variables: their values are sorted based on a specific criteria. We use the PDR of unicast packets for the first ranking, and the broadcast rate of DIOs and EBs for the second ranking. The Spearman coefficient has the same range as the Pearson coefficient and exhibits the highest value when both variables have identical ranking and the lowest when they are fully opposed.

$$\rho_{x,y} = 1 - \frac{6 \sum \left[rank(x_i) - rank(y_i)\right]^2}{l(l^2-1)} \tag{3.10}$$

where l is the sample size.

Figure 3.5b shows the average rank correlation for different values of w. We adjust the observation window from 1 to 10 minutes. We adopt the same approach (Fisher transformation) as in Section 3.2.3. We also consider that any value of $\bar{\rho}$ greater than 0.8 is a strong correlation factor (Section 3.2.3). As seen in the Pearson correlation result (Figure 3.5a), for lower values of w the ranking correlation ρ is also weak. In this case, the collisions bias the estimation as well. Around 3 minutes, the two rankings exhibit a strong correlation.

3.2.5 Neighbors Ranking Efficiency

We consider that a node does not need to estimate initially the exact PDR of unicast packets to a neighbor. A node has rather to identify one good neighbor to join the network and to start reserving cells and sending packets. Thus, neighbors with almost the same link quality should be considered equivalent: a refined estimation would be too expensive for the corresponding added value. Inversely, a node must absolutely avoid selecting a bad parent: the joining procedure would be very expensive.

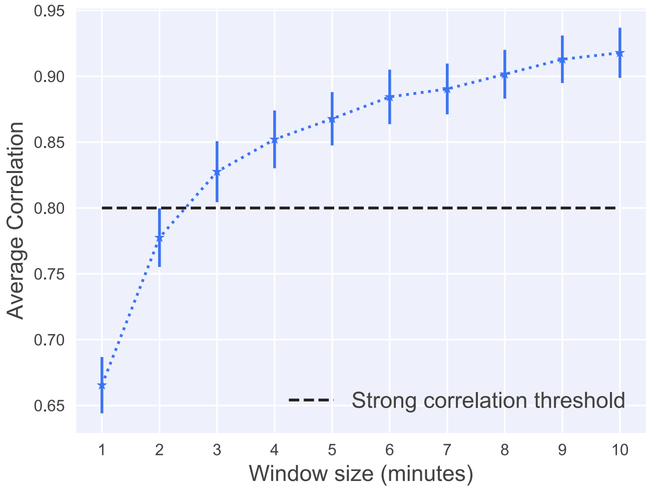
We make the following distinctions between different rankings (as in Figure 3.6):

1. **Perfect ranking**: both rankings are identical. While the PDR are different in broadcast and unicast, all the neighbors are ranked identically;

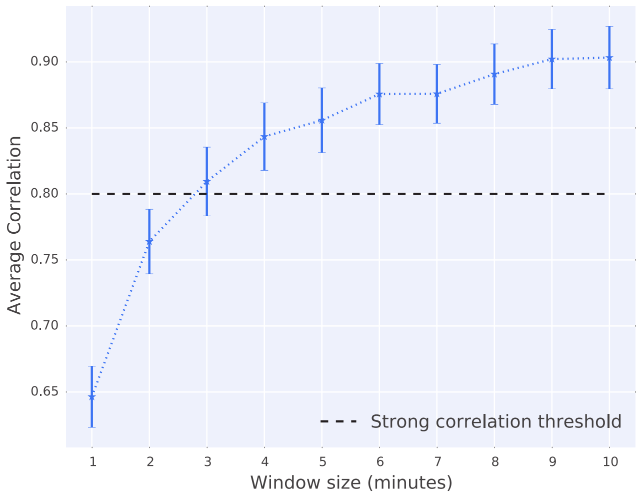
2. **Acceptable ranking**: the rankings are slightly different, but the top ranked nodes are still neighbors with the best qualities. For instance, the node B (Fig. 3.6) is identified as a good neighbor when considering the unicast PDR;

3. **Bad ranking**: both ranking are very different, and we cannot use the broadcast PDR to identify good neighbors. In Figure 3.6, we would select the node D since it provides the highest broadcast rate although it provides a very poor unicast PDR.

We aim here to quantify the difference among the broadcast and unicast rankings. For instance, if we select the neighbor with the highest broadcast rate, what



(a) Pearson correlation of broadcast vs. unicast PDR.



(b) Spearman correlation of the rank's neighbors.

Figure 3.5: Average Pearson and Spearman correlation coefficients, considering a confidence level of 95%.

would be the decrease in PDR compared with a solution measuring directly the unicast PDR (which reserves a dedicated cell with each neighbor and generates probe packets)? We define the following metric to compute the normalized difference between the PDR which may be obtained using the two corresponding rankings (unicast vs. broadcast):

$$\Delta(R_u, R_b, t) = \frac{\sum\limits_{v=1}^{t} PDR(R_u, v) - PDR(R_b, v)}{\sum\limits_{v=1}^{t} PDR(R_u, v)} \tag{3.11}$$

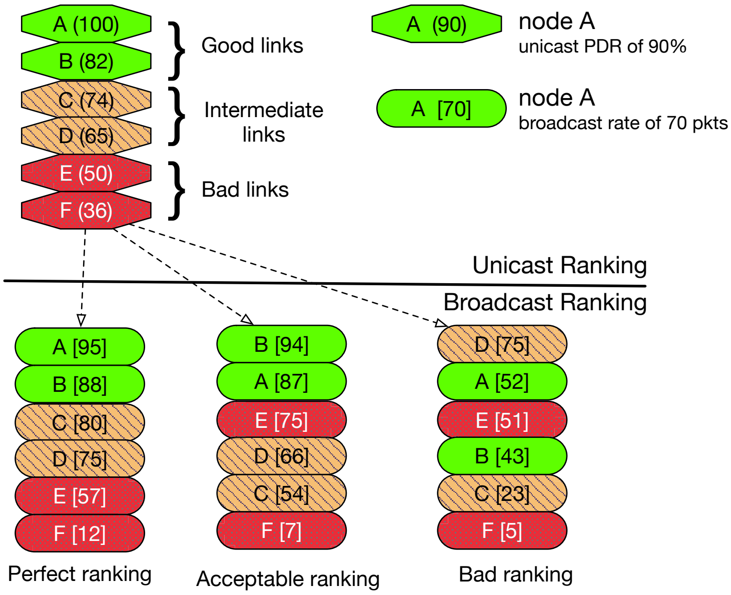


Figure 3.6: Different types of ranking agreement.

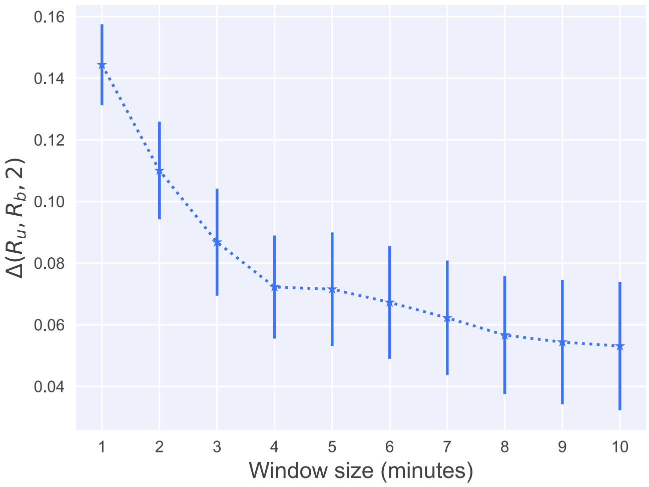
We consider that a node must be able to identify two good preferred parents. Indeed, several routing solutions rely on a primary and a backup route to provide high-reliability, i.e. multipath routing (Chap. 2). For instance, 6TiSCH advocates the selection of two parents, with a replication scheme between the two routes [29]. Thus, we aim here to evaluate the ability of the broadcast rate metric to select the two best ranked neighbors ($t = 2$ in eq. 3.11).

We can note that, in Figure 3.7a, the $\Delta(R_u, R_b, 2)$ tends to decrease for higher values of w. Around $w = 3$ minutes, its average yields less than 0.1, corresponding to a very low error rate: the broadcast rate can be used successfully to infer the unicast quality.

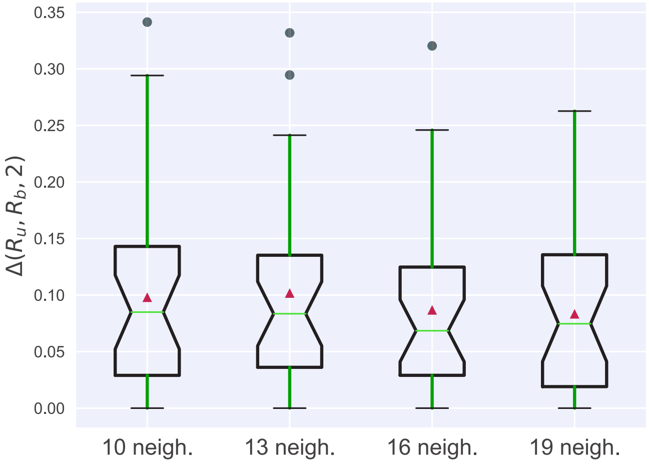
Next, we measure the impact of the density, i.e. from 10 to 19 neighbors. When the density is high, the collision rate in the shared slots increases. Thus, the broadcast rate is more loosely dependent on the link quality: the packets are dropped mostly because of collisions.

To limit the collision probability for this quite high density, we use three shared cells, placed uniformly in the slotframe. Otherwise, the number of neighbors is too high and one shared cell would not be sufficient to transmit all the control packets [89].

Figure 3.7b shows the impact of the node density on our ranking approach for $w = 4$ minutes. We observe that the ranking performs similarly regardless of the number of neighbors. For such scenarios, the quantity of shared cells and their new positioning in the slotframe decreases the impact of collisions on the ranking and we achieve a very low error rate. To deal with even higher density, we would have to increase the number of shared cells so the nodes have more opportunities to send their broadcast packets at different times. A collision of the broadcast packet would impact the convergence and the stability of the network [168].



(a) Ranking accuracy



(b) Ranking accuracy for different densities

Figure 3.7: Average of the difference between the top ranked nodes in both rankings for different observation windows and the impact of the number of neighbors on the ranking (95% of confidence).

3.3 Using the Broadcast Rate for the RPL's rank

6TiSCH minimal advocates the use of the Objective Function OF0 [116] for rank computation using the ETX metric to estimate the link quality. Straightforwardly, we can use the rate of broadcast packets received instead of the ETX in the rank computation:

$$rank(i,j) = rank(j) + 3 * \left[\frac{expected\ adv.\ packets\ by\ i}{DIO(j) + EB(j)}\right] * 256 \tag{3.12}$$

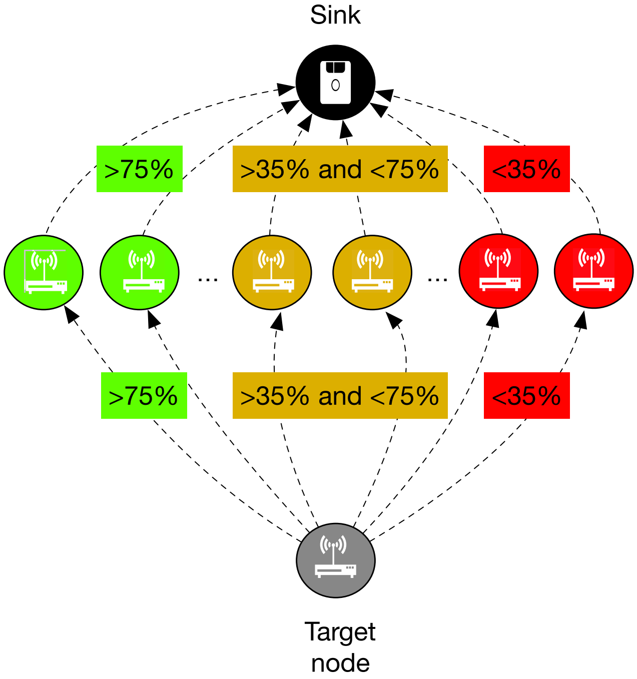


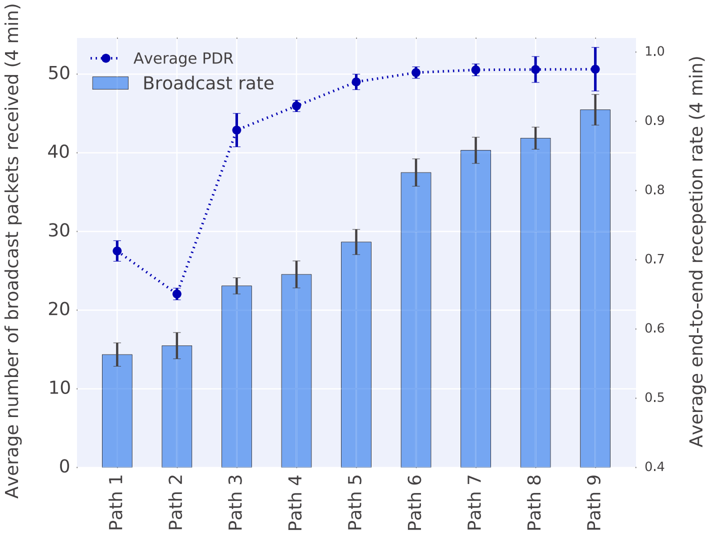
Figure 3.8: Multipath topology to measure the end-to-end path quality.

The rank of a node is in this case the rank of the preferred parent, plus the inverse of the broadcast rate. To normalize the broadcast rate, a node assumes its broadcast period is the same as those of its neighbors.

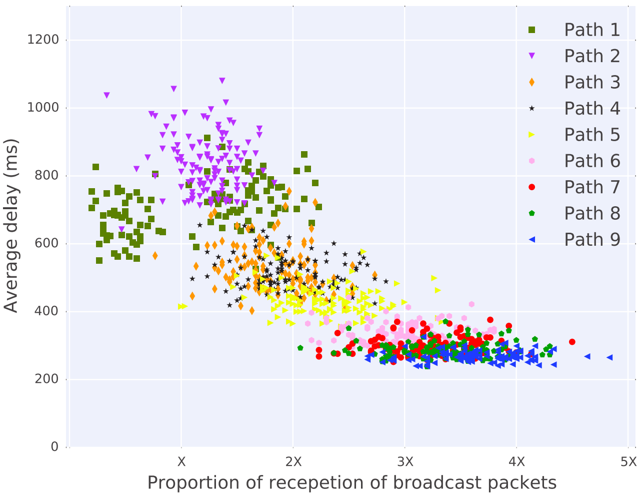
To measure the accuracy of this rank to capture the end-to-end path quality, we focus now on a multipath topology (Figure 3.8). Each link has a different link quality, which is roughly the same for each hop of the path. A good path must provide a high end-to-end reliability, with a small delay. We argue that the broadcast rate allows each node to select the best parents, and to set up globally efficient routes.

The broadcast rate is simply obtained by monitoring the number of broadcast packets received from each neighbor along each path. Then, we measure independently the end-to-end PDR of each possible path: dedicated cells are reserved for each path, and the target node generates one data packet per second to the sink. We selected an observation window of 4 minutes since it exhibits a high linear correlation (Figure 3.5a) and a high rank correlation (Figure 3.5b.

Figure 3.9a illustrates the broadcast rate of each neighbor and the PDR of each path, ordered by their path quality. Both paths 1 and 2 exhibit the lowest broadcast rate and the poorest end-to-end PDR – $\approx 65\%$. We have then the medium quality paths (3 and 4) which provide a medium end-to-end PDR and they are classified medium for the broadcast rate metric. Finally, all the other paths provide an high PDR, with significantly different broadcast rates (37 for the path 9 vs. 23 for the path 5). While we are not able to derive the PDR from the broadcast rate, it remains a very good discriminator of path quality. Practically, a node has several good possible parents and should select one of them, while limiting the overhead (probes, number of 6P reservations, parent changes, etc.). In conclusion, the broadcast rate



(a) End-to-end PDR



(b) End-to-end delay

Figure 3.9: Unicast PDR and Broadcast rate for each path (confidence level of 95%) and the impact of the number of broadcast rate on the end-to-end delay.

is definitely an accurate metric to detect fast and passively one of the best parents.

We finally measure the end-to-end delay achieved through the different paths (Figure 3.9b). We clearly identify a strong correlation between the PDR and the delay: a low reliability implies that the node has to retransmit several times the packet before receiving an acknowledgement. Thus, as anticipated, paths 1 and 2 present the highest delay and the poorest reliability.

3.4 Integration in 6TiSCH and SF0

We integrated our passive approach in the latest version of OpenWSN stack[6]. Each node registers every reception of a broadcast packet sent by each of its neighbors for a period of time (equal to the observation window w). Every w minutes, the node ranks its neighbors based on the broadcast rate, and selects the best one as its preferred parent. In addition, we activate the Scheduling Function SF0 [173] to allocate the dedicated cells in a distributed manner.

We compare the two following approaches:

- **blind selection**: the default behavior of openWSN, using a default ETX metric of 2 (50% of error rate) for an inactive neighbor;

- **broadcast-rate aware selection**: a node selects a neighbor as preferred parent taking into account the broadcast reception rate.

To study the convergence, we compare the number of parent changes for both versions. Every parent change starts a new negotiation process between a node and its new preferred parent, where they exchange control packets to reserve bandwidth. The objective here is to show that our method is tolerant to short variation on the link quality.

We use a peer-to-peer network composed of 13 nodes deployed in the same corridor in the FIT/IoT-LAB in Grenoble. Unlike the others experiments, we did not force any topology here: the nodes decide autonomously the final topology and the dedicated cells to use. The slotframe length was 101 slots, with 5 shared cells. In our implementation, we positioned uniformly the shared slots in the slotframe and we restrict them to broadcast packets only, as we did in the previous experiments. Each experiment lasts for 1 hour and the observation window was 4 minutes.

Figure 3.10 shows the Complementary Cumulative Distribution Function (CCDF) of the number of parent changes for both implementations. We can note that the blind selection has a higher rate of parent switching. Because a node does not effectively compute the link quality with an inactive neighbor, it uses the default value cost to estimate the link quality (see Section 3.1.1). This implicitly makes a node to "blindly" select another parent when the link between this node and its preferred parent is refined and classified as bad. On the other hand, our passive approach divides approximately by half the number of parent switchings.

3.5 Lessons Learned

Sending probe packets in a 6TiSCH network is unrealistic: it may consume a huge volume of resource to test each node individually and to select the most accurate ones. From the experimental results we collected, our approach has showed that estimating the link quality using the broadcast reception rate is a viable solution for 6TiSCH networks. With our passive method, a node can identify the *best* neighbors, i.e. those which would provide a high PDR and a low delay.
We also draw the following conclusions:

[6] http://openwsn.org/

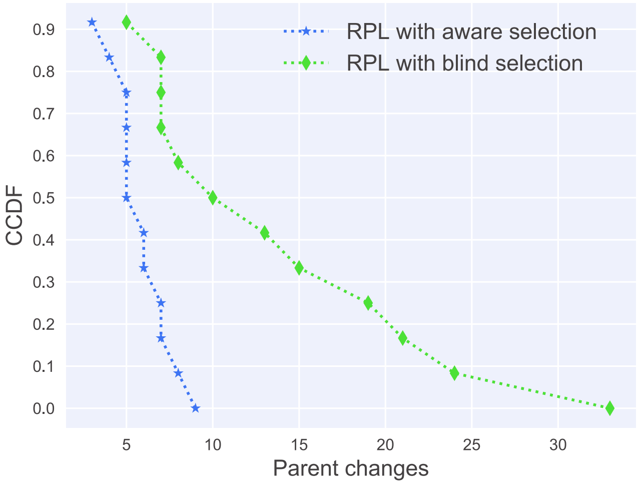


Figure 3.10: The difference in number of preferred parent changes between the two implementations.

Observation window size: straightforwardly, the accuracy of our method is closely linked to the length of the observation window. Since packet losses represent a stochastic variable, a too short observation window implies a larger bias and the node would misestimate the link quality. This problem is even exacerbated by the *multichannel effect*: each channel must be used a sufficient and equal number of times to have a correct estimation since some channels may be subject to external interference [28]. However, a longer observation window reduces the reactivity time and a node may not react fast enough when a sudden fault happens. In our experiments, 4 minutes represented a good tradeoff.

Rapid convergence: our method minimizes the number of parent changes during both the bootstrap and the operating phases. A passive estimation is required to take fast decisions, without generating a large overhead. As a result, our approach has shown to be energy efficient and a node will change its preferred parent only when necessary.

Collisions minimization: because our method exploits the shared cells, we need to minimize the occurrence of collisions in these slots. One option is to restrict the shared cells to broadcast packets only and to reduce the slotframe length. With a shorter slotframe, the shared cell will repeat more often giving the nodes more chances to transmit their advertisement packets at different times. A second one is to uniformly distribute the shared cells along the slotframe, instead of placing them together at the beginning. By applying the second option, our method showed robustness to be used in a network with high density (up to 19 neighbors).

Scheduling integration: our method may be plugged in both centralized and distributed scheduling approaches. In a centralized approach, each node may compute a ranking of its neighbors, pushed to the Path Computation Element (PCE) which would be in charge of selecting the best routes and to construct the final schedule. This monitoring information may be continuously transmitted to the central component by a combination of piggybacking and dedicated control packets [174]. Different broadcast rates and rankings may be a good indicator for the PCE to detect asymmetrical links, and to allocate accordingly enough bandwidth along the best routes.

3.6 Summary

We proposed here to monitor the broadcast rate to estimate the unicast link quality of the different neighbors. Rather than estimating precisely the unicast PDR, we aim to *classify* the different neighbors, to only select as preferred parent one of the best neighbors. Indeed, the unicast PDR and the broadcast rate exhibit a very strong correlation, and we highlighted experimentally that we could safely rank the link qualities using this metric. We integrated this mechanism in the 6TiSCH stack, and we demonstrated experimentally the relevance of this passive method. We reduced the number of parent changes because a node selects immediately one good preferred parent, instead of testing iteratively each of its neighbors.

We only considered a fix rate of control traffic, since we wanted to keep all nodes with a constant rate of broadcast transmission during the experiments. This allowed us to have a more predictable collision probability, simplifying our analysis. However, different nodes may use a different RPL's DIO period, because of the trickle algorithm: the period is doubled when no routing inconsistency is detected.

A variate DIO traffic rate may, for instance, increase the time window to build an accurate ranking when using our passive link estimator presented in Chap. 3. However, we conjecture that the trickle timer should not impact the convergence during the bootstrapping period: all the nodes will reset their DIO period until the routing layer has converged. We recommend additionally to evaluate the precision of our estimator when using only Enhanced Beacons for ranking. To find an optimal frequency for transmitting EB, considering the probability of collision and the energy cost still need to be proposed.

In the next chapter, we employ our method to reduce the network instability caused by the blind parent selection used by 6TiSCH. In particular, we apply a filtering approach that reduces the probability of a bad parent being chosen, even if this parent presents a significantly lower RPL rank.

Chapter 4

The Dynamics of an Indoor Low Power Lossy Network

Contents

Although IEEE 802.15.4-TSCH provides strong delivery guarantees, the radio environment still exhibits time-variable characteristics [42]. Thus, the network has to provision sufficient resource (bandwidth) to cope with short-term variations: additional resources allow the network to operate in the worst situation. However, over-provisioning decreases the network capacity, since more cells are reserved for retransmissions.

RPL tries to deal with channel instabilities by adjusting the topology dynamically and by ensuring that each device has one appropriate neighbor to communicate with. Typically, a trickle timer aims to decrease the volume of control traffic when the network topology is stable. Unfortunately, RPL has been proved to overreact to link quality changes [168]. Thus, dynamic routes may be a source of instability.

In this chapter, we investigate the performance stability of a 6TiSCH network in indoor deployments. In particular, we perform a multi-layer analysis identifying inefficient operations of the 6TiSCH stack that hinder the routing topology and the schedule to converge. We focus particularly on long-term deployments, where the network should reach the steady-state with all nodes synchronized and able to communicate. The radio environment is time-variant, because of e.g. multipath fading, external interference, obstacles. However, we argue that the network should keep on providing a deterministic behavior.

Contribution

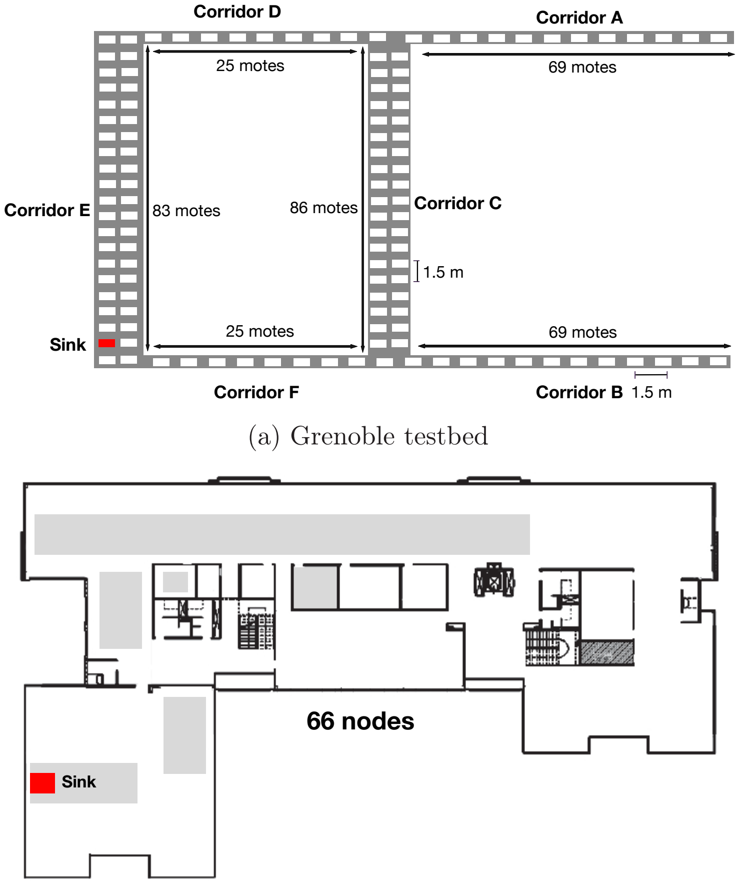
1. We present a preliminary study to highlight the presence of instability: the schedule and the routes keep on changing continuously, even for long experiments;
2. We identify several causes of instability in the 6TiSCH stack (collisions for control packets, link quality mis-estimation, negotiation of cells through unreliable links, routing oscillations), and quantify them in an indoor environment;
3. We compare the performance achieved through two different testbeds to highlight the persistence of this instability problem;
4. We propose per-layer solutions to solve this instability, and to make the network performance stable, without continuous reconfigurations.

4.1 Problem statement

In typical industrial deployments, it is not rare to find different wireless technologies cohabiting in the same environment [80]. Because of its low power nature, a 6TiSCH network is particularly in disadvantage when sharing the same unlicensed band with other higher power networks such as Wi-Fi. Additionally, obstacles like concrete walls and heavy metal machines make the wireless channel non-stationary in long-term [175].

Iova *et al.* [99] already demonstrated that RPL changes routes frequently even with a static topology. Indeed, RPL tries to identify efficient routes by exploiting a link quality metric (e.g reliability). Unfortunately, this quality is usually a stochastic variable and can only be estimated with a certain inaccuracy.

A topology reconfiguration does not come for free and it is usually followed by a burst of control packets transmissions. In particular, 6TiSCH is reservation-based and cells have to be negotiated all along the path toward the sink. Worse, the number of dropped packets tends to increase since reservations take time, generating buffer overflows [89]. Therefore, we need efficient protocols that ensure reconfigurations only when really needed, e.g. sudden fault, definitive link quality changes.



(a) Grenoble testbed

(b) Lille testbed (2nd floor)

Figure 4.1: The testbeds used in this study. Only the sinks were used in all experiments.

4.1.1 Definition

Industrial networks are deterministic to provide strict guarantees. It is particularly hard to respect a set of guarantees if the protocols keep on oscillating. Making a diagnostic when everything is dynamic is also particularly challenging. More precisely, we adopt here the following definition:

> A network infrastructure is stable if the state of its protocols remains unchanged for small-term variations (e.g. static topology with sporadic external interference).

In other words, this means that a network has to be reconfigured when a long-term variation is detected. For instance, the routing protocol must change the route when the next hop runs out of energy, but has to keep the same next hop for non significant link quality variations. The optimality has a cost, since the protocols have to exchange many control packets before re-converging eventually to a steady state. Even worse, the network may keep on oscillating, never converging. We argue that this reconfiguration cost exceeds the additional resources to cope with short-term variations.

4.1.2 Impact of network instability

Because we expect a wide adoption of the Internet of Things, all the applications will have to share the same unlicensed band, creating a high volume of cross technologies interference, possibly with different transmission power. Thus, as highlighted previously, the networks will rely on dynamic algorithms, with continuous reconfigurations even under unchanged network conditions (same nodes location, same volume of traffic), in order to combat narrow-band noise and to provide strict guarantees. The multichannel feature of IEEE 802.15.4-TSCH should rather allow the network to limit the number of reconfigurations to provide stable performances.

In centralized scheduling algorithms, the sink has a global view of the network and can adjust the schedule in case of instability. However, we need efficient mechanisms to detect the level of interference dynamically and to avoid the sink to schedule interfering links simultaneously. To our minds, this constitutes a very challenging problem. Indeed, centralized algorithms usually assume that nodes more than k hops apart do not interfere, which is a very imprecise assumption.

For distributed schedule functions, the nodes decide autonomously the cells to use, before detecting and solving collisions. Thus, changes in the schedule also increase the probability of collisions, thus impacting even flows which have not been rescheduled. Additionally, before allocating the bandwidth, a node relies only on shared cells to send its reservation packets to the new parent, where collisions are frequent. As a result, the re-convergence may be deferred while the nodes try to communicate in a best-effort way.

4.2 Preliminary Study

To illustrate the problem discussed previously, we investigate the performance stability of 6TiSCH networks when deployed in indoor environments.

4.2.1 Scenarios

We use two different testbeds for this study, FIT IoT-LAB in Grenoble and in Lille (Figure 4.1). In Grenoble the nodes are placed in long corridors without fixed physical obstacles among them. On the other hand, in Lille the nodes are distributed across different corridors separated by walls. These two testbeds correspond to different scenarios with different channel characteristics.

We select 31 nodes, and we perform 30 repetitions of 90 minutes in both testbeds. The sinks are the same in all experiments and we place them at the extremity of the corridors. The other nodes are selected randomly before launching each experiment, providing a different topology for each repetition and increasing the representativity of our experiments. Additionally, we adjust the transmission power in order to have multi-hop networks with several hops. We target here industrial applications where nodes report their data to the sink with a Constant Bit Rate (CBR) traffic (10 seconds). Thus, everything is here static (i.e. topology, traffic), and we expect a convergence to a steady-state. We employ M3 nodes in our study and the OpenWSN implementation. We set a slotframe composed of 199 slots, where 8 were shared slots restricted to broadcast and 6P control packets.

Figure 4.2 illustrates (i) the number of parent changes, (ii) the number of 6P requests, (iii) the packet delivery ratio and (iv) the end-to-end delay. Surprisingly, we keep on identifying continuously routing reconfigurations (i.e. parent changes) during all experiments in both testbeds, highlighting that the instability problem is not restricted to a single testbed. Unfortunately, routing reconfigurations introduce a burst of 6P request packets while nodes reserve dedicated slots. In multihop networks, this problem becomes even more severe since the cells have to be re-negotiated all along the path (i.e. exactly until a common ancestor is reached if it did not yet release the corresponding resources). Worse, these reconfigurations deeply impact the reliability: the packet delivery ratio drops significantly when a burst of RPL reconfigurations occurs. Indeed, the re-convergence takes time as reserving new cells is not instantaneous. Packets are thus dropped because of buffer overflows. Additionally, the frequent routing reconfigurations impact directly the delay, since the routing protocol may build not optimized paths while the network re-converges.

4.2.2 Causes of network instability

We describe now the causes that led the networks to transit from a stable to an highly unstable state. We have identified two events that can be the origin in the flow of events leading to instability:

Schedule inconsistency: it typically happens when a node adjusts its schedule because of topology changes in its subtree. For instance, when a new child joins the subtree, the node requests more dedicated slots to its preferred parent to accommodate the new requirements. In some cases, the schedules of the node and its parent become inconsistent;

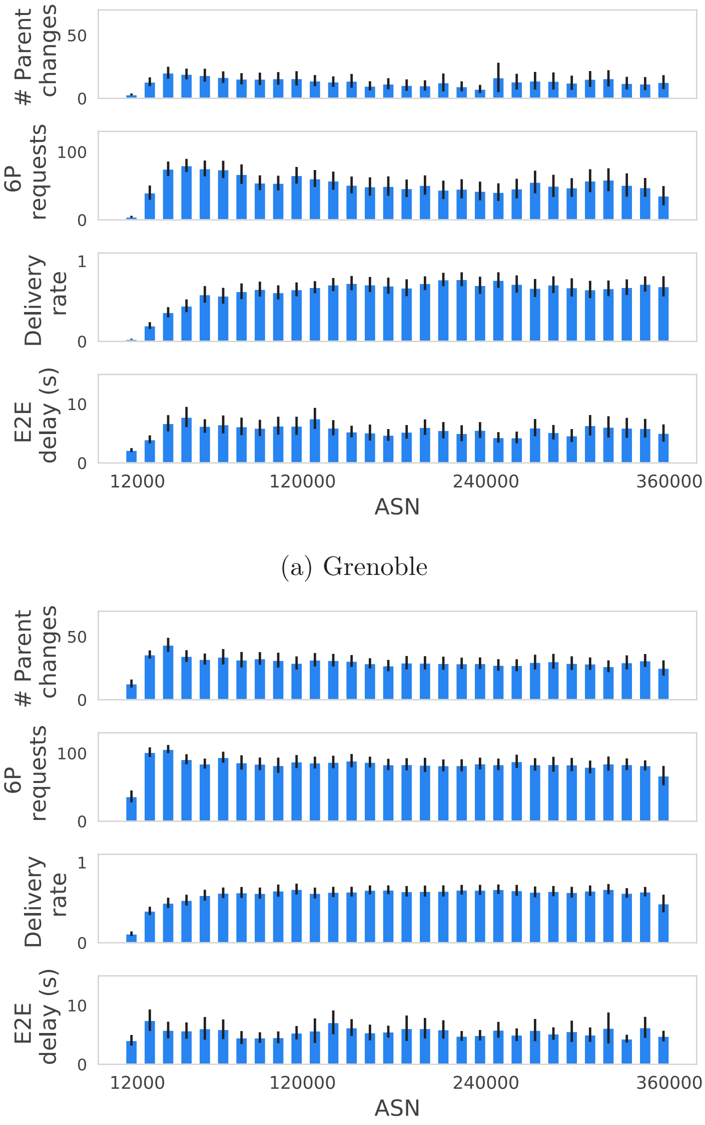
Routing oscillations: a node changes its preferred parent when another neighbor provides a significantly smaller rank. This occurs even if the link quality between the node and its immediate parent is high. We consider that a parent change may come either from a bad initial choice, or from an overreaction to a link quality change. To our mind, the two problems require specific solutions.

In both cases, nodes have to use shared cells for communicating. Thus, ensuring low collision probability in shared cell is crucial for fast reconvergence. In the next sections, we perform our multi-layer analysis proposing improvements to each layer to reduce the network instability.

4.3 Layer 2: collision mitigation in shared cells

During moments of high routing instability, several nodes may change their preferred parents, increasing even more the contention in the shared cells. In particular, during the bootstrap phase, nodes probe their neighbors iteratively until they find a suitable preferred parent.

Here, we discuss approaches to mitigate the collision problem in the shared cells. We demonstrate that a simple re-arrangement of shared cells in the slotframe is enough to reduce the overhead caused by high contention. We do not consider the



(a) Grenoble

(b) Lille

Figure 4.2: The network performance in both testbeds for 30 experiments repetitions (95% of confidence level).

discovery time for novel nodes, which have to scan all the channels before receiving an EB to get synchronized [176]. The discovery is triggered once, before a node gets synchronized, and is thus less prejudicial to the stability.

4.3.1 Repeated collisions

In a 6TiSCH network, all nodes broadcast EB and DIO packets periodically. An EB contains crucial information (e.g. slotframe length, hopping sequence) that allows other nodes to synchronize with the sink. However, the IEEE 802.15.4-TSCH standard does not specify the rate at which the EBs have to be transmitted.

Similarly, DIO are used to disseminate routing information (rank) over the network.

A node starts to send broadcast packets as soon as it joins the network, according to a time interval common to all nodes. Because initially nodes that joined the network at the same time have the same frequency of transmission, their broadcast packets collide repeatedly. Obviously, these collisions impact very negatively the network formation time: a node has to wait longer to receive a correct EB, so that it can be synchronized with the TSCH network.

Using a jitter before broadcasting control packets is an alternative to avoid repeated collisions. Typically, the jitter increases the time window in which a node enqueues broadcast packets. For instance, when the broadcast period is l and the jitter is γ, the time when the next broadcast packet will be enqueued will be randomly selected within the interval $[l - \gamma, l + \gamma]$. Therefore, collisions are less repetitive, since the nodes have now the possibility to transmit broadcast packet at different moments.

4.3.2 Distributed shared cells

Even when enqueuing broadcast packets randomly, the probability of collision remains high when the shared cells are placed at the beginning of the slotframe. To illustrate this problem, let us consider a slotframe that has only one shared cell placed at the first position of the slotframe. For simplicity, let us consider that a shared cell is only used for transmitting EBs. Additionally, all nodes send EBs periodically every ε seconds. To avoid repeated collisions, we consider a small jitter γ in a way that a node selects its next transmission time by randomly choosing a value between $[\varepsilon - \gamma, \varepsilon + \gamma]$. In this scenario, we denote as Δ_{shared} the time between two consecutive shared cells. The total number of Δ_{shared} repetitions within the interval $[\varepsilon - \gamma, \varepsilon + \gamma]$ is given by the following equation:

$$K = \left\lfloor \frac{2\gamma}{\Delta_{shared}} \right\rfloor \tag{4.1}$$

The probability of repetitive collisions can be calculated as the probability of at least two nodes re-selecting the same Δ_{shared} repetition to enqueue their control packets. In such occurrence, the transmitters will try to send their packets using the next shared cell. The probability of collision is then given by:

$$p[collision] = 1 - \frac{K!}{K^n * (K - n)!} \tag{4.2}$$

where n is the number of neighboring nodes.

According to Equation 4.2, the probability of collision for a network sending control packets in a range of 10 seconds, having 101 timeslots, each lasting 10 ms, and having 6 neighbors is approximately 85%. Adding more shared cells to the beginning of the slotframe does not reduce the probability of collision. The reason is that nodes can still enqueue their control packets during the same Δ_{shared} repetition before the next slotframe repetition.

To reduce the probability of collision, we need to consider a larger jitter or to shrink Δ_{shared}. However, in dense networks we would need a large γ to have enough Δ_{shared} repetitions to accommodate a high number of neighbors with low

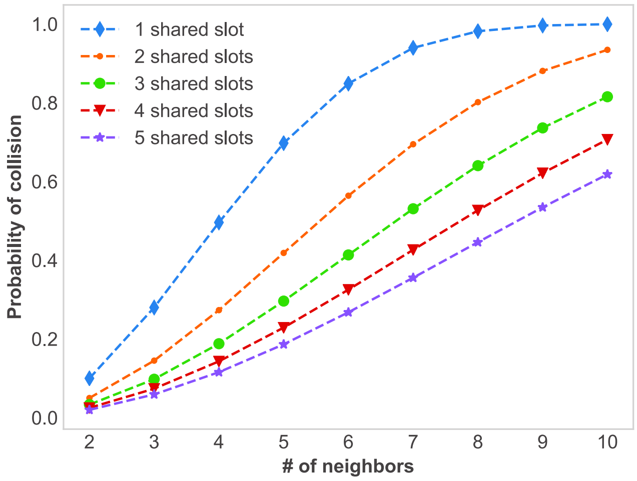


Figure 4.3: Probability of collision for different number of neighbors and shared cells.

probability of collision. Consequently, the network would be less reactive, as the rate of control packets being transmitted would be lower. On the other hand, by distributing shared cells uniformly over a slotframe, we shrink Δ_{shared}, forcing the nodes to pass through a shared cell more often. In that case, we can modify slightly Equation 4.1 to:

$$K' = \left(\frac{2\gamma}{\Delta_{shared}}\right) * Total_{shared} \tag{4.3}$$

where $Total_{shared}$ is the number of shared cells placed distributively over the slotframe.

Figure 4.3 shows the probability of collision considering different number of shared cells. For a single shared cell case, the probability of collision yields 50% when the number of neighbors is only 4. When we increase the number of distributed shared cells, the probability of collision goes down, as the nodes use different Δ_{shared} repetitions to queue EB. For 5 shared slots, the probability of collision yields approximately 60% considering all 10 nodes. These results highlight that the probability of collision in the shared cells depends on the number and the position of shared cells.

4.4 Layer 2.5: schedule inconsistency management

Because radio environments are known to be lossy, 6P relies on mechanisms to preserve schedule consistency between a node and its parent. Typically, a 6P sequence number is maintained per link, incremented after every modification triggered by any of the two corresponding nodes. In particular, their schedules are considered inconsistent if the two nodes have a different 6P sequence number. Unfortunately,

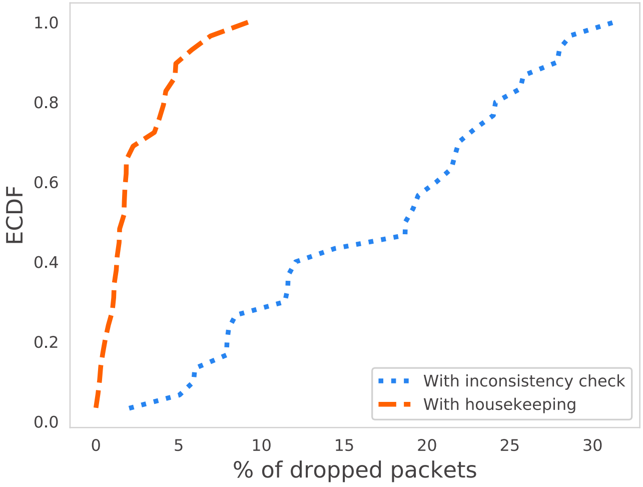


Figure 4.4: Frequency of dropped packets by experiment caused by buffer overflow after a 6P-clear command in Grenoble.

this may arise when a 6P transaction exceeds the maximum number of retransmissions. The 6P packet is dropped, and both sides of the link have then a different sequence number.

SFx advocates that when two neighbors have inconsistent schedules, they have to flush all their cells with these neighbors after a `6P-clear` command [173]. Then, the two nodes have to restart the reservations from scratch. This corresponds exactly to the same situation as a parent change, when a node redirects the traffic from its children to the new parent, re-allocating all the dedicated cells. Therefore, the shared cells have to be used since the two nodes do not have a common cell anymore. Since collisions are frequent during these cells, the reservation may take a long time.

In our preliminary study (Section 4.2), we also measured the number of dropped packets caused by buffer overflow. Figure 4.4 reflects all the 30 experiments performed in the Grenoble testbed. We count the number of packets dropped after a 6P-clear command. Typically, this command is triggered either after a parent change to deallocate the cells in the previous parent or an inconsistency detection by 6P. We can note that a large number of packets are dropped after a 6P-clear command (30% of the packet for one experiment).

Therefore, we consider not using the sequence number: a housekeeping feature [177] is enough for detecting schedule inconsistencies. A housekeeping is an autonomous feature that monitors system parameters periodically looking for inconsistencies, i.e. cells that are allocated in the transmitter but not in the receiver or vice-versa. Inconsistent cells have two different effects:

tx-cell: the receiver is not awake, and the transmitter never receives an acknowledgement. Typically, this inconsistency occurs when the transmitter tries to deallocated dedicated cell because of new communication requirements, e.g. some children may leave its subtree. Quickly, the incriminated cell exhibits a poor reliability compared with the other valid cells toward the same neighbor. Thus, the Scheduling Function will trigger a relocation, exactly like when a collision occurs, and will remove the corresponding inconsistency;

rx-cell: the receiver has just to stay awake but does not receive anything. After a timeout, the corresponding cell will be released. This inconsistency just impacts the energy consumption (idle listening) but not the reliability.

Figure 4.4 illustrates the number of packets dropped after a 6P-clear command when we use a housekeeping function, and we deactivate the 6P inconsistency check. We can clearly see that we reduce largely the number of packets dropped because of 6P. Inconsistencies should be handled individually, and a clear-command appears too aggressive.

4.5 Layer 3: routing oscillations

The routing protocol aims to construct efficient routes in multihop topologies. By efficiency, low power lossy networks often designate the reliability and energy efficiency. Thus, we have to select the most reliable links while globally limiting the energy consumption.

The link quality indicators provided by the radio chipset are easy to obtain and they provide a quick way of assessing the channel state. However, these indicators have been proved to present good estimates only under specific circumstances. For instance, the accuracy of the Radio Signal Strengthen Indicator (RSSI) decreases when it falls in the gray zone or under the effect of external interference [178].

As shown in Chapter 3, estimating directly the reliability of the links in a 6TiSCH network is much more challenging. Indeed, a node may continuously count the number of packets correctly acknowledged by it children and parent nodes. However, it cannot estimate passively the quality for inactive neighbors: no packet is exchanged, by definition. Usually, most solutions rely on probing: some dedicated control packets are transmitted to re-evaluate continuously the link quality [167].

In addition, we also discussed the main limitations of employing a default link metric for unknown neighbors. The challenge consists in not being too passive or too aggressive when defining the default value. In the former case, nodes will persist communicating using unreliable paths, decreasing the network reliability. On the other hand, a very low value can cause the network to react excessively, leading to recurrent routes reconfigurations.

4.5.1 Circular parent changes

To quantify the number of useless parent changes, we count the number of circular changes. By circular change, we mean a list of parent changes so that the node reselects back the initial parent of the list. For instance, in Figure 4.5 the neighbor 1 is selected at the instant t_0 and reselected at t_2. Similarly, the neighbor 2 is selected at t_3 and t_5. This chain of changes is certainly useless and it wastes energy: overprovisioning additional cells would have been much more efficient than changing the preferred parent, and renegotiating several cells (de/re-allocation).

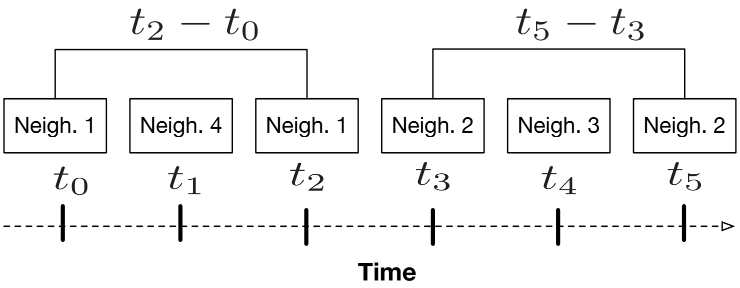


Figure 4.5: List of circular changes ordered by time.

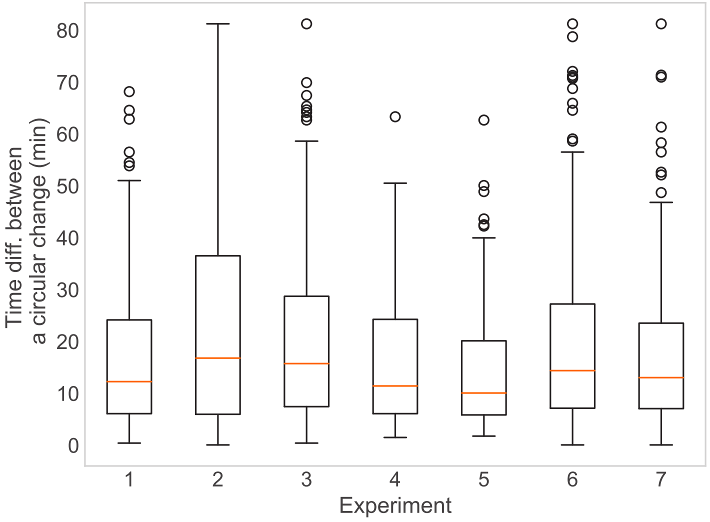


Figure 4.6: Distribution of the time difference between circular changes for 7 experiments selected randomly.

Figure 4.6 illustrates the distribution of the time difference between two changes to the same parent. We select 7 experiments randomly, from our preliminary study in Lille (Section 4.2), to quantify the number of circular changes. We observe that the selected experiments follow typically the same distribution. In these experiments, approximately 50% of all circular changes occurred with less than 15 minutes. In IEEE 802.15.4-TSCH networks, we must account the control packets exchanged during the negotiation and the required time to reallocate the active cells. In long-term deployments (e.g. weeks, months), circular changes should be present only when the gain exceeds the reconfiguration cost. The reconfiguration has also to be correctly handled to configure properly the novel paths *before* the flows are redirected, to respect the reliability and delay constraints.

4.5.2 Inaccurate parent selection

In our preliminary evaluation (Section 4.2), we identified several parent changes due to what we call the existence of a *misleading zone*. A *misleading zone* is composed

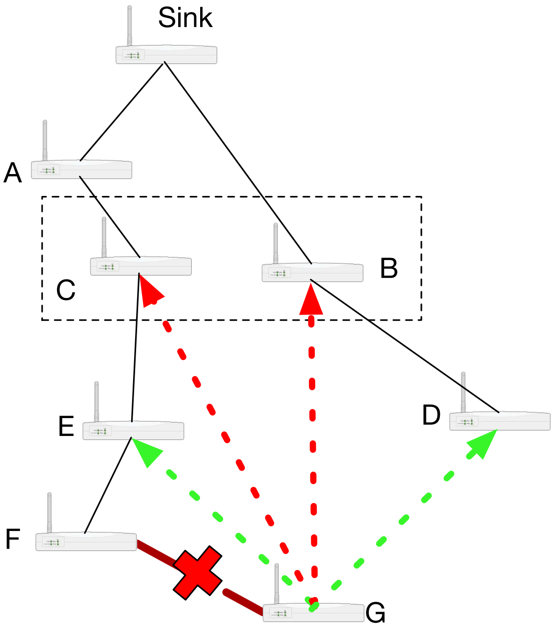


Figure 4.7: Misleading zone of the node G.

of nodes with low ranks, *virtually close* to the sink. Long links have been proved to exhibit often a very poor reliability [179]. Thus, a node may receive `beacons` from neighbors in the misleading zone time to time, infrequently.

Let us consider the topology illustrated in Figure 4.7. The node G has a preferred parent (F) for which the link quality degrades, and has to search for another parent. If it picks the neighbor with the lowest rank, it will select probably the nodes B or C, which provide a very low PDR. The node G should rather select E or D, with a much better tradeoff between reliability, and routing progress (i.e. closer to the sink).

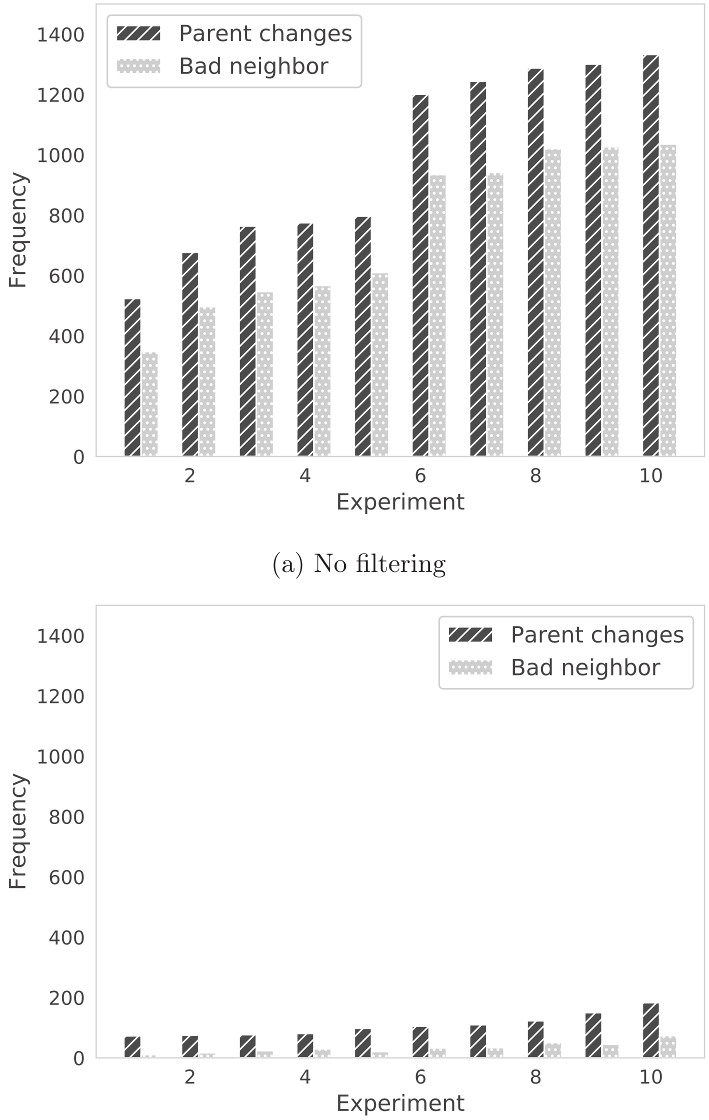
To avoid selecting a bad parent, we propose to exploit a passive link quality estimation by monitoring the broadcast packets transmitted periodically by all the nodes. We demonstrated in Chapter 3 the existence of a correlation between this unicast and broadcast Packet Delivery Ratios.

We propose to apply a filtering approach:

1. a node counts the number of EB packet received from each of its neighbors. Then, a Exponential Weighted Moving Average (WMEWMA) is applied to smooth this metric;

2. the link quality is then estimated proportionally to this broadcast rate metric. This way, the best neighbors are scanned preferentially.

Figure 4.8a illustrates the number of parent changes and the number of times that a bad preferred parent was selected. A new parent is considered a *bad neighbor* when the negotiation between a node and its new parent fails or when the new parent has a lower PDR than the previous one. We selected 10 experiments randomly before and after introducing the filtering approach from our preliminary study in

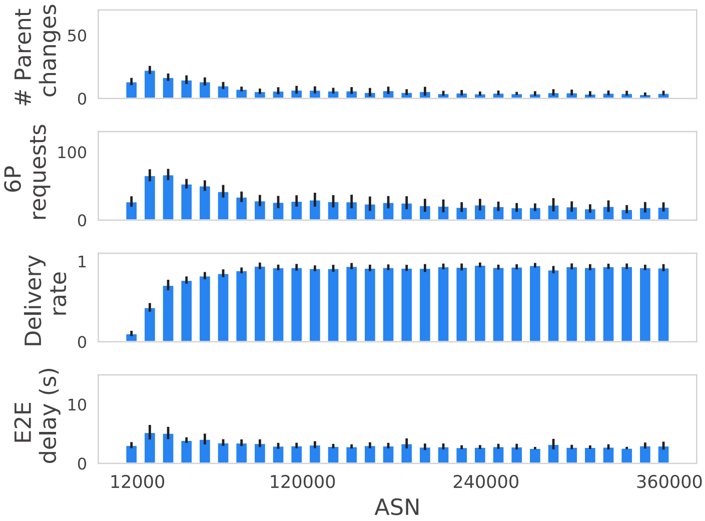


(a) No filtering

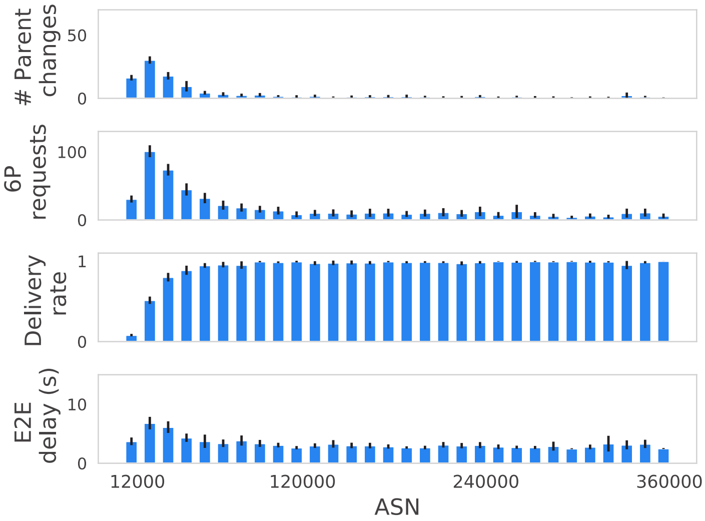
(b) Filtering

Figure 4.8: Frequency of parent changes and bad parent selection.

Lille. Without pre-filtering the neighbors, the number of parent changes is quite high, between 500 and 1300 depending on external interference. On the contrary, using the broadcast rate (Figure 4.8b) helps to identify the best neighbors, without using an erroneous default link metric. Consequently, we select a good parent at the first try, and we have much less parent changes. Alternatively, we may here execute an active (and expensive) measurement method for the second step, to discriminate the best neighbors. However, our experimental evaluation demonstrates this passive measurement is sufficient to construct a stable and efficient topology, without requiring any additional control packet.



(a) Grenoble



(b) Lille

Figure 4.9: The performance stability in both testbeds after adding the housekeeping and filtering approaches (95 % of confidence level).

4.6 Experimental Evaluation

We focus now on measuring the network performance stability after integrating our modifications to the 6TiSCH stack (schedule housekeeping, misleading zone identification, filtering). We conduct our experiments in the same deployments (Grenoble and Lille), using the same platform (M3). Our proposals were implemented in OpenWSN and are freely available on GitHub[7]. We use the same settings as in Section 4.2 (e.g. slotframe size, shared cells, traffic load, sink position). In these experiments,

[7] `https://github.com/rodrigoth/openwsn-fw/tree/convergence`

all nodes execute our housekeeping approach (Section 4.4). Every 10 minutes, each node looks for schedule inconsistencies, i.e. cells with no rx or tx success. When a node has an allocated rx-cell that was never used, the housekeeping algorithm removes it. Additionally, a tx-cell that has never received an ACK is also removed.

We use a shared platform, deployed in office buildings, where Wi-Fi networks are also used. We do not have any control on the level of external interference. Thus, we run 30 different experiments for each value, picking randomly the nodes in the testbed (except the sink which remains the same). Each experiment runs for 90 minutes independently on each of the two testbeds. We plot confidence intervals to verify that the results are significant, i.e. exhibit small variations. The dataset generated in our experiments is also freely available online[8]to be analyzed by anyone. We expect to conduct experiments with controlled external interference in a controlled environment in a future work to analyze finely the impact of external interference.

Moreover, all nodes register the broadcast packets (EB+DIO) that were sent every 30 seconds in average by their neighbors. As explained in Section 4.5, each node uses this broadcast rate to rank its neighbors. It selects its neighbor with the highest broadcast rate as preferred parent. A WMEWMA estimator helps to smooth the variations. In particular, we decided to *not* implement any probing mechanism, particularly expensive in IEEE 802.15.4-TSCH environments. Thus, the best candidate neighbors are those who are ranked in the first positions and a node prioritizes them when it needs to change its preferred parent.

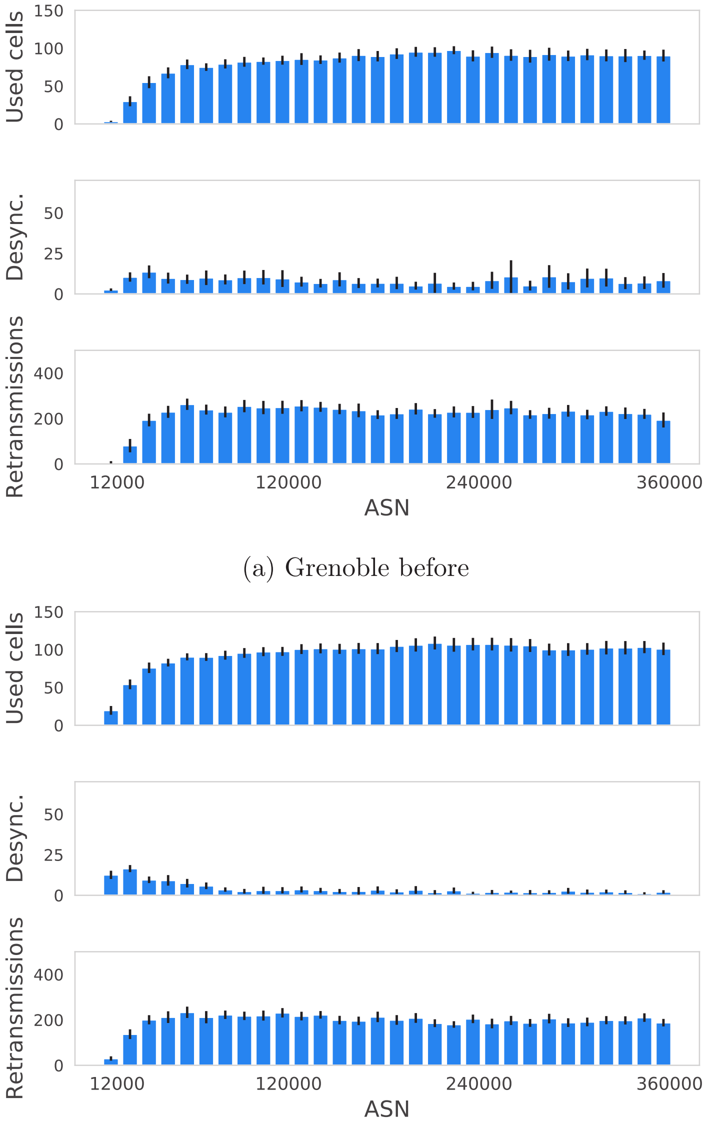
Figure 4.9 highlights the network performance when using our approaches. Since our filtering approach relies on counting broadcast packets, the convergence time depends on the intensity of collisions in the shared cells. Indeed, the nodes send initially a burst of 6P requests and the probability of collisions in a shared cell is very high. However, as soon as the nodes reserve dedicated slots, the shared slots are less used and our solution becomes more precise.

Because the nodes have a precise view of their environment, they select a good candidate neighbor in the first attempts, thus largely reducing parent changes. This is reflected in a shorter convergence time and a smoother routing instability period during the operating phase. Routing reconfigurations are very infrequent, compared with the 6TiSCH stack without any optimization (Fig. 4.2). We reduce by 88% the number of parent changes during the 90 minutes of the experiment.

Similarly, we also reduce the number of 6P requests. Indeed, a parent change means also that the cells have to be re-negotiated with the novel parent. Moreover, a 6P request has a limited length, and contains only a small number of cells to insert. Thus, several 6P handshakes may be required before having a sufficient number of cells to forward all the packets in the queue. 6P requests are still required to adjust the number of cells when the link quality changes significantly. However, these changes just allow the node to empty its queue more easily, and do not impact negatively the reliability.

Additionally, our modified version of the 6TiSCH stack provides a end-to-end reliability of 99% for both testbeds. These results differ from the unoptimized 6TiSCH stack (Fig. 4.2), where the network performance was very unstable in all experiments.

[8] `http://www.rodrigoteleshermeto.com/#dataset`



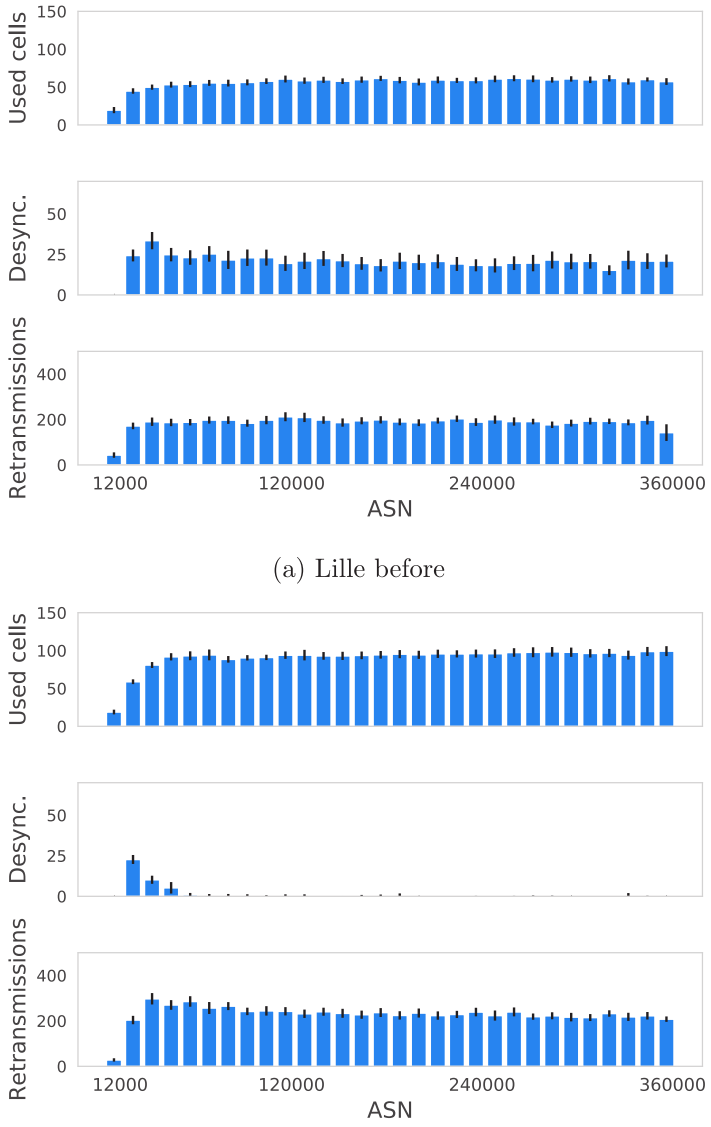
(a) Grenoble before

(b) Grenoble after

Figure 4.10: Network efficiency before and after our modifications in Grenoble (95 % of confidence level).

Differently from the results in our preliminary study (Section 4.2), we observe now a near constant end-to-end delay across all experiments in both testbeds. The routing stability achieved after adding our modifications allows the construction of reliable paths, with an upper bounded delay.

Finally, we compare the impact of our modifications on the schedule occupancy (Figures 4.10 and 4.11). We can note that we reserve approximatively the same number of cells in Grenoble, but the number of cells is higher for Lille after our modifications. Indeed, we remind that the delivery rate before our modifications was very low, and many packets were dropped (Fig. 4.2). Mechanically, these non-transmitted packets do not need cells to be scheduled.



(a) Lille before

(b) Lille after

Figure 4.11: Network efficiency before and after our modifications in Lille (95 % of confidence level).

In timeslotted networks, the number of retransmissions is tightly related to the energy consumption. Indeed, the nodes can quickly turn off their radio during idle slots, and their energy consumption is much smaller compared with busy slots [180]. In Figures 4.10 and 4.11, we can verify that enforcing stability is not detrimental to the network reliability. In Grenoble, we keep on using the most reliable links, with the same number of retransmissions while providing an higher packet delivery rate. In Lille, we succeed to deliver the packets through long and unreliable routes, which increases both the delivery rate and the number of retransmissions.

Finally, we also measure the number of desynchronizations. A node becomes desynchronized typically after a schedule's inconsistency, when it has to flush all its dedicated cells. We can see that we reduced largely the number of desynchronizations in Lille: all the nodes keep synchronized, able to forward the packets.

Therefore, our approaches allow the deployment of stable, reliable and predictable multihop networks even in presence of external interference (e.g. Wi-Fi, other concurrent experiments on the testbed).

4.7 Summary

We investigated here the performance stability of 6TiSCH networks in two indoor testbeds with different channel conditions. We showed that the network does not succeed to converge to a steady-state through two different testbeds, even in static situations. For both testbeds, we can identify moments of performance instability due to oscillations in the radio conditions caused by external interference and obstacles.

We identified the causes of instabilities, and proposed solution for each of the layers in the 6TiSCH stack. First, we demonstrated that a rearrangement of shared cells in the slotframe reduces the probability of collisions for control packets, paving the way to a faster negotiation during topology reconfigurations. Next, we simplified the schedule consistency management between two nodes to reduce the network instability caused by renegotiating from scratch all the cells when they detect a schedule inconsistency. We also exploited the existing correlation between the broadcast packet reception rate and the unicast link quality to create a two-step parent selection, avoiding bad choices leading to instabilities. We finally obtain a network that converges faster and that reacts accurately during moments of instabilities.

As perceptive, we recommend a further convergence analysis to study in-depth the network re-convergence in the presence of long-term modifications. The network must be robust enough to keep on guaranteeing a minimum reliability when e.g. a node has crashed. When reconfiguring one part of the network, we must be sure to be able to respect the reliability and delay constraints after the reconfiguration. Thus, the network has to be reconfigured before the flow is redirected. Multipath, over-provisioning, and worst-case analysis should help to solve such challenge. In addition, it is still necessary to investigate the stability of a 6TiSCH stack when considering bursty traffic, e.g. alarms with strict delay constraints. The scheduling functions must be designed to not over-react to changes. Tuning the network resources for the worst case then appears particularly challenging.

In the next chapter, we investigate the relevance of replacing unicast to *anycast* transmissions. We believe that anycast can increase the chance of a packet making progress in the network, since multiple receivers are associated to a single transmission. Therefore, we just need one of them to decode correctly the packet to consider a transmission as successful. However, we must ensure that the packet losses between multiple receivers are low correlated, otherwise a transmission would be equally lost by all of them.

Chapter 5

Link-layer Anycast Scheduling for IEEE 802.15.4-TSCH Networks

Contents

Anycast is a link-layer technique to improve the reliability when using lossy links. This technique has been proposed to authorize several receivers to be active at the same time, so that a transmission is considered erroneous when none of the receivers was able to decode and acknowledge it. Since a transmission is lost only if all receivers fail to receive a packet, the network reliability and the energy efficiency can be significantly improved [138]. Hosni *et al.* [139] investigated the impact on the reliability when choosing the best parents to forward the packets. However, they assume that packet loss probabilities are independent for all the links, which may not hold practically. For instance, external interference may impact all the receivers simultaneously [181].

While ideal radio propagation models provide very stable characteristics, experimental evaluations prove reality is much more complex. The link burstiness measures (see Chap. 2) the time-variant packet losses for a given link [140]. A receiver may succeed to decode a sequence of packets, and then stop receiving for a long time. In other words, packet losses are not independent, because of e.g. external interference. Identifying long-term stable links may help to avoid oscillations, but would also reduce the routing diversity [182].

Here, we first conduct a thorough experimental study to assess the relevance of an anycast technique at the link layer to improve the network reliability. We also investigate the impact of slow channel hopping on the packet losses independency. We use an indoor large-scale platform, mimicking a smart building application, where multipath propagation is very common. Besides, other colocated networks (Wi-Fi, or IEEE 802.15.4 compliant networks) may also generate external interference. We then propose a strategy to select the set of forwarding nodes, while investigating specific rules for next hop selection at the routing layer.

Contribution

1. We evaluate the correlation factor among the packet transmissions for a group of receivers. In other words, for each device, does there exist a set of neighbors with (almost) perfectly independent packet losses?
2. We explain a method to schedule several receivers for a packet, combined with minimal 6TiSCH [90] to increase the packet delivery ratio for each transmission;
3. We show that greedily selecting the best parents (i.e. providing the higher Packet Delivery Ratio) is insufficient since they may exhibit very correlated statistics. We thus propose an heuristic to select a proper set of parents;
4. We assess the performance of our heuristic, evaluating the reliability achieved in a multihop, realistic environment.

5.1 Anycast in a 6TiSCH stack

6TiSCH has been initially designed for unicast transmissions: in the schedule, a cell is assigned to a pair of receiver / transmitter. We describe here how to modify this stack to enable anycast transmissions. We consider only non cross-layer features, and focus on the anycast feature at the link layer.

5.1.1 Limits of Unicast Communications

Because radio links are lossy, over-provisioning must be implemented, i.e. additional cells to retransmit the packets [183]. Unfortunately, these additional cells impact negatively:

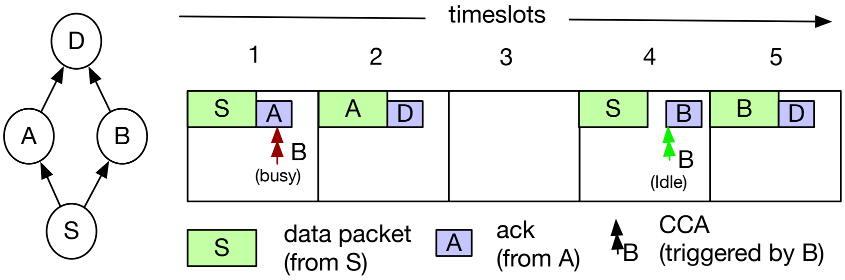


Figure 5.1: Anycast Transmissions.

the delay: the device has to wait for the next transmission opportunities. Since we implement a low duty cycle ratio, we may in the worst case wait for the next slotframe, which increases the delay and the jitter;

the network capacity: we expect a huge utilization of the same unlicensed band. Thus, we must use scarcely the radio resources: provisioning additional cells means also reducing the network capacity for the rest of the network, or the other co-located deployments. In particular, IEEE 802.15.4-TSCH has been proved to generate a large number of collisions in dense deployments [184].

Using a single path is also not fault-tolerant. A single faulty node is sufficient to *break* the route, and no packet is finally received by the controller. Some may argue that the routing protocol (i.e. RPL [106]) is in charge of finding another route after having detecting the fault. However, such reconfiguration is particularly expensive in synchronized networks: the bandwidth has to be re-allocated along the novel path. As shown in Chap. 4, this reservation requires a large number of control packets, the convergence may be quite long, e.g. a few minutes in some cases, with potential oscillations. Multiple *anycast paths* should make the routes much more robust and fault tolerant, while avoiding heavy reconfiguration costs (i.e. bandwidth, energy, time).

5.1.2 Implementing anycast

Anycast would reduce the number of transmissions by assigning several receivers for one single transmitter. We aim to provide the same level of reliability with less retransmissions. Let us consider the scenario illustrated in Figure 5.1. The node S has two parents A and B. When it transmits a data packet, A has the highest priority and sends its `ack` first. The second parent B triggers a CCA after a fixed interval, and detects in the first timeslot an `ack` is on-going: it drops the packet from S. Then, A forwards the packet to D. For the second packet of S (timeslot 4), the node A is unable to decode the packet, and does not detect anything when triggering its CCA: B sends its `ack` to S and finally relays the packet to D.

We can note that anycast is efficient for the nodes more than 2 hops away from the border router. However, longer routes tend to be less reliable, since more relay nodes have to forward the packets. Thus, anycast may improve the reliability and would in the worst case fallback to the unicast scenario.

Typically, anycast implies the following consequences:

Receivers ordering: because several receivers can decode the packet, the receivers have to be *prioritized*. A receiver will forward a packet only if all the other receivers with a larger priority failed to decode the packet;

ACK: the transmitter must be sure its packet has been received by at least one receiver. For this purpose, anycast often advocates the usage of contention resolution for `ack`. More precisely, a receiver waits for a backoff inversely proportional to its priority. If the CCA triggered before transmitting its `ack` is positive, the node estimates that another receiver with a higher priority is currently transmitting an `ack`; it stops the process;

False negative: with high external interference, the nodes may conclude erroneously with a CCA that a `ack` is already in transmission. They would stop the process, and the transmitter will not receive an `ack` for its transmission, even if one of the receivers was able to decode it.

However, the transmitter would in that case retransmit the packet. Thus, the reliability with anycast would be at least as good as the unicast case;

Duplicates: if the `acks` collide (false negative for the CCA), the transmitter will retransmit its packet, generating duplicates. Thus, the set of forwarding nodes has to be properly constructed to avoid hidden terminals. Typically, each node has to report the list of its neighbors in its `beacons`, so that a node can select parents with a sufficient link quality from one to the others.

5.1.3 Shared Cells Scheduling

Anycast requires to let several receivers to wake-up synchronously, spending more energy. We want to show experimentally that such mechanism is really efficient to improve the reliability. We aim to demonstrate that packet losses may be sufficiently independent to provide a significant gain in end-to-end Packet Delivery Ratio (PDR). Anycast is energetically relevant when the gain is higher than the energy spent for overhearing.

In addition, anycast scheduling imposes that a transmitter negotiates a common cell with all the receivers. Thus, several handshakes are required to pre-reserve a cell in each receiver and then start using it when all receivers confirm it was available. Practically, this increases both the number of 6P packets and the convergence time. Our objective is here to rather quantify such gain, before modifying the protocols to support anycast. Thus, we propose here to use the 6TiSCH-minimal schedule [90]. A collection of shared cells are reserved for all the nodes. Thus, a transmitter can safely use shared cells: all its neighbors will be awake, and will be able to decode the packet if they have to forward it.

By employing a low-traffic rate of data packets, we can reduce the probability that the same shared cell is used by different transmissions (i.e. collisions). Thus, we can focus uniquely on the anycast mechanism.

5.1.4 Preliminary Results

To assess the degree of correlated losses, we collect a large dataset of packet transmissions / receptions on a large-scale testbed. Our objective here is to characterize the wireless links to verify the feasibility of using anycast communication in an indoor environment. For this, we rely one more time on the FIT IoT-LAB testbed in Lille (Chap. 4).

We configure a static schedule where each node takes a turn to broadcast a burst of 20 packets, every 30 seconds, during 15 minutes. We select randomly 15 nodes on the second floor of the testbed. Each node records the success / failure of each packet of the burst.

We compute the Pearson (ϕ) correlation for every pair of links that received at least one packet from a transmitter in a given burst. This correlation factor is particularly relevant to measure the correlation among two stochastic variables which do not present the same average value. Indeed, we aim to compare the correlation between two links, whatever their average PDR.

We consider all links with correlation factor below 0.4 as low correlated [172]. Figure 5.2a depicts the Empirical Cumulative Distribution Function (ECDF) of the number of low correlated parents by transmitter. We observe that all nodes have at least two *independent* parents, i.e. their packet losses are very loosely correlated. Thus, selecting these two parents would increase the Packet Delivery Ratio for the link: when the first parent fails to decode the packet, the packet success for the secondary parent is probabilistically uncorrelated.

We also measure the impact of the distance on the inter-link correlation. Figure 5.2b presents a linear regression using the least squares method between the two variables considering 95 % of confidence level. As expected, farther receivers tend to present weaker correlation: when two receivers are farther than 25 meters, the average correlation is below our threshold value of 0.40. Thus, nodes geographically close to each other tend to present strong correlation and may not provide a significant gain in terms of diversity.

5.2 Parent Selection for Anycast Routing

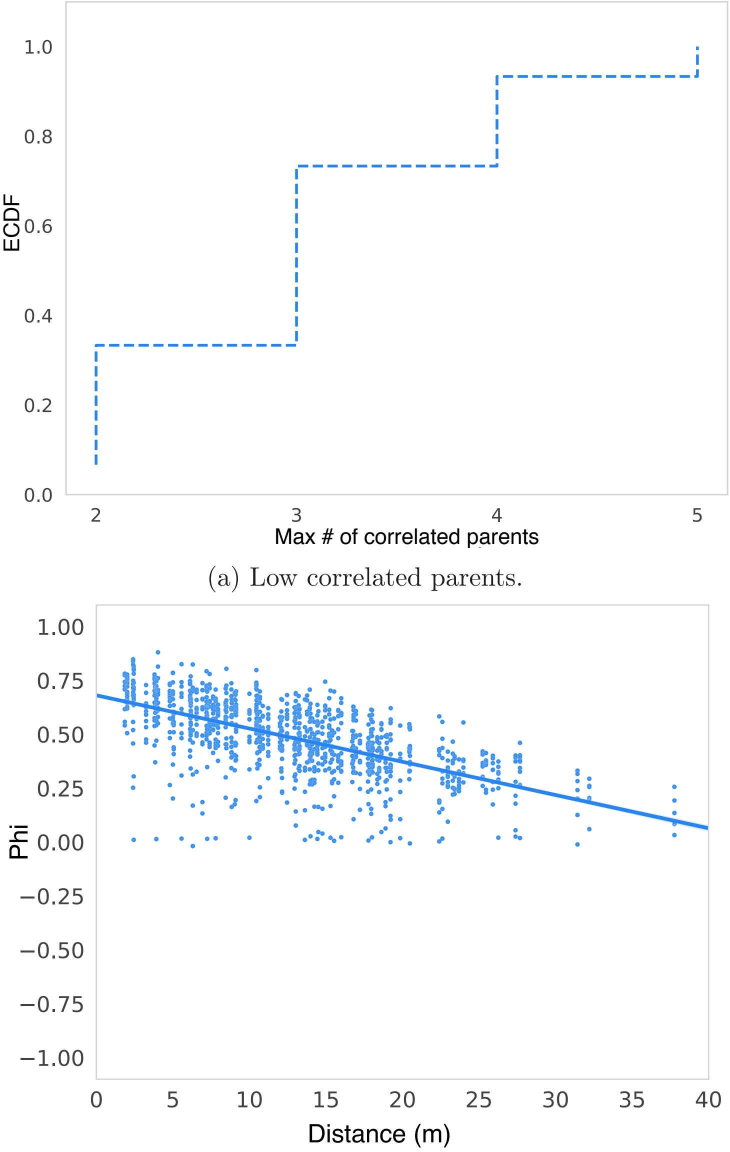
Inserting too many parents would waste energy: some of them would have to stay awake while never forwarding any packet (the other parents are sufficient). Inversely, we increase the number of retransmissions when too few parents are selected. Thus, we have to select the right set of parents.

We consider here a convergecast traffic pattern (see Chap. 2). Each device selects multiple parents, and forwards all its packets in anycast to them.

5.2.1 Reliable Parent Set Notification

All the nodes have to wake-up during the shared cells. However, only the authorized receivers have the right to acknowledge and to forward a packet. In particular, the receivers have to know their *priority* for a given transmitter. Thus, each transmitter has to notify its selected receivers with their respective priorities. This notification has to be reliable, to avoid deafness.

The nodes transmit the ordered list of receivers (a short address with 16 bits) in an Information Element, piggybacked in the Enhanced Beacons and routing control packets. To the list is also associated a timestamp (Absolute Sequence Number), at which the transmitter will switch to a novel list.



(a) Low correlated parents.

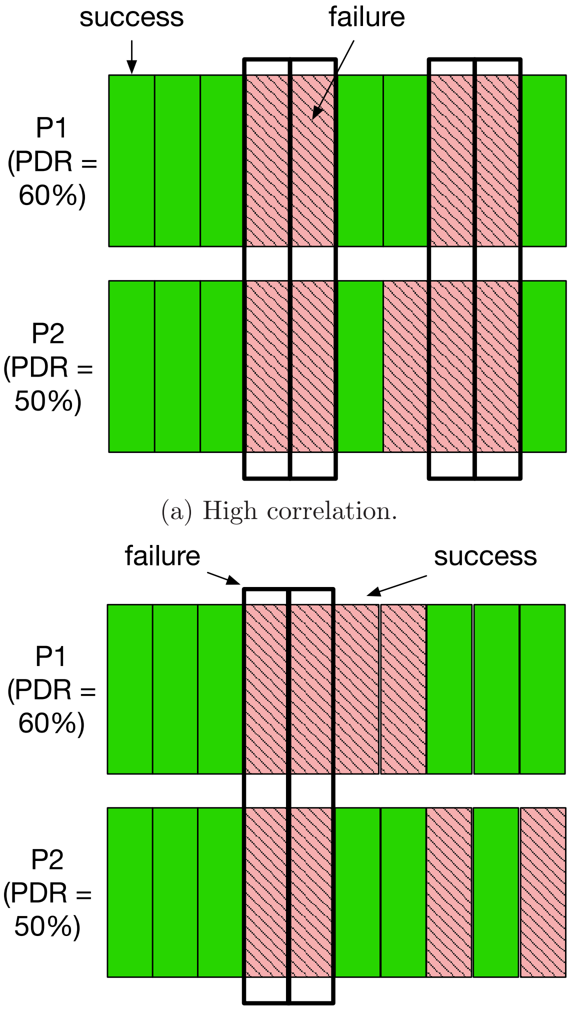
(b) Linear regression between distance and Phi (ϕ).

Figure 5.2: (a) Maximum number of parents with low correlation by transmitter and (b) linear regression between distance and Phi with 95% of confidence level.

5.2.2 Greedy PDR Parent Selection Limitation

Usually, we compute the Packet Delivery Ratio (PDR) to assess the reliability of a given transmission. In unicast communication, a node computes the PDR per link, i.e. the ratio of the number of `acks` and the number of packets transmitted to a specific neighbor.

If we assume independent packet losses, we can apply the optimal method described in [139]. Each node prioritizes its possible parents based on their RPL rank and their individual PDR. Thus, candidate neighbors with higher PDR and lower RPL rank are more likely to be chosen as parents. By using the RPL rank as con-



(a) High correlation.

(b) Low correlation.

Figure 5.3: Two links with different correlation values.

dition, we ensure a loop-free route. We assume that nodes compute their respective ranks based on link quality metrics, such as Expected Transmission Count (ETX), to exploit reliable paths. Finally, a node selects greedily high-priority neighbors in its parent set until the PDR reaches a maximum value.

Let $PDR(i,p)$ denote the PDR from the node i to its parent p, and $\mathcal{P}(i)$ denotes the set of parents for i. A packet is considered lost if none of its parents received it. If we consider independent packet losses, we obtain:

$$PDR\,(i \rightarrow P(i)) = 1 - \prod_{p \in \mathcal{P}} (1 - PDR(i,p)) \tag{5.1}$$

However, this equation does not hold anymore if packet losses are correlated among the different parents. To show the limitation of this approach, let us consider the Figure 5.3 illustrating the success / failure of 10 packets for two parents. In Figure 5.3a, we have two intermediate links with a PDR of 60% and 50% for parents

1 and 2 respectively. When we look at each individual transmissions, we observe that all packets received by Parent 2 were also received by Parent 1, exhibiting a strong correlation. The anycast delivery rate is in that case bounded by the highest PDR of the two links (60%). Formulated differently, anycast would not provide any gain, since the diversity is insufficient to decrease the packet losses.

Anycast scheduling can achieve higher reliability only if the receivers have weak or no correlation in terms of packet losses. This situation with the same PDR as in the previous example is highlighted in Figure 5.3b. In that case, anycast would achieve a PDR of 80%, as only the 4^{th} and 5^{th} packets are not received by any of the two parents.

5.2.3 Joint-Packet Delivery Ratio for anycast links

In realistic conditions, a packet may be lost because of e.g. external interference, which impacts all the receivers. In that case, we cannot anymore assume the packet losses are independent, and the equation 5.1 does not hold anymore.

When packet losses exhibit a strong correlation, the conditional probability is not equal to the product of the success probabilities: events are not anymore independent. Typically, if the packet toward the first parent has been lost, this is highly probable that it has **also** been lost for the secondary parent. Inserting the secondary parent in the forwarding set has no positive effect on the reliability. If Parent 1 was unable to decode the packet, it is highly probable that Parent 2 will never succeed to receive it either.

Since the parents are ordered by their PDR in the greedy-PDR strategy, the parents with a lower PDR have to be integrated in the parent set only if they provide independent results. Else, their reliability gain can be neglected.

Thus, we propose here the Joint Packet Delivery Ratio (**J-PDR**) metric to consider anycast transmissions with any packet losses correlation. J-PDR takes into account the fact that multiple receivers are listening for each transmission. Thus, we have to compute the multi-neighbor delivery ratio.

We use the concept of **transmission success sequences**. Each receiver stores independently the transmission success for the last k packets. This sequence of bits is reported regularly to the transmitter (within an Information Element as in [185]) so that it can compute the J-PDR. More precisely, each receiver reports its reception sequence to the transmitter, i.e. a bitmap, with one bit per packet transmission. Since the transmitter uses a sequence number incremented at each packet transmission, a receiver knows a packet has not been received by identifying the *voids* in the sequence numbers.

More formally, let us denote by $s_{ij}(k)$ the binary variable equal to 1 iif j has received the k^{th} packet from the node i. Thus $< s_{ij} >$ denotes the transmission success sequence from i to j. The node i computes J-PDR to its set of parents P as follows:

$$\text{J-PDR}(i \rightarrow P) = 1 - \frac{|\{k|\forall j, s_{ij}(k) = 0\}|_{k=1}^{z}}{|z|} \tag{5.2}$$

with z the number of packets in the sequence.

Algorithm 1: Selecting the n best neighbors according to their J-PDR.

```
   Data: Set of neighbors (Neighbors)
   Data: Number of neighbors (n)
   Result: Set of Parents(Parents)
1  $checked_{nb} \leftarrow 0$
2  $before_{jpdr} \leftarrow 0$
3  $after_{jpdr} \leftarrow 0$
4  $candidate \leftarrow \emptyset$
5  $rank\_by\_pdr(Neighbors)$
6  $Parents \leftarrow Neighbors[0]$
7  while $checked_{nb}$ < length(Neighbors) and length(Parents) < $n$ do
8      $before_{jpdr} \leftarrow compute\_jpdr(Parents)$
9      $candidate \leftarrow get\_next\_neighbor()$
10     $after_{jpdr} \leftarrow compute\_jpdr(Parents, candidate)$
11     if $after_{jpdr} > before_{jpdr}$ then
12         $Parents \leftarrow Parents \cup \{candidate\}$
```

In other words, a packet is considered lost if none of the receivers received it properly.

5.2.4 Greedy J-PDR Parent Selection

We propose rather to implement a greedy approach using the J-PDR metric, shown in Algorithm 1. Constructing the optimal set is computationally intensive: we have to test all the possible combinations.

The algorithm receives as inputs the set of all node's neighbors (*Neighbors*) and the maximum number of selected parents (n). The output is a sorted list of selected parents according to their packet losses correlation. We rank first the neighbors with a lower RPL rank according to a given metric (line 5). We select first the neighbor with the highest PDR, since it has a larger probability to increase the reliability for anycast (line 6). Next, we check for an additional neighbor that, when combined with the first selected neighbor, the J-PDR increases (lines 8-10). Typically, neighbors with weak packet loss correlation are *complementary*: when one fails, the other succeeds. The function *compute_jpdr* performs a logical OR operation using the transmission success sequences of all selected parents, resulting in a single sequence representing the current multi-neighbor delivery rate. We insert new neighbors in the parent set while the final J-PDR (selected parents and the candidate) for the link increases (line 12).

5.3 Experimental Evaluation

We base all our experiments on data traces obtained from a real deployment of 100 nodes spread over 3 floors in a research center building located in Lille (random topology). We collect statistics for all transmitted packets and their respective sender/receivers. We use the same approach as in Section 5.1.4, where a node takes a turn to broadcast a burst of packets periodically. The number of packets per burst and the broadcast periods are the same, 20 and 15 minutes respectively. The network operated for 25 hours during a regular working day, collecting approximately 7 million measurements. The complete dataset is freely available on GitHub[9].

Our objective is here to investigate the interest of using anycast at the link layer, apart from any specific protocol mechanism. Before spending some time to implement anycast on prototypes, we are convinced we need to prove anycast is efficient when exploiting real links, where packet losses may not be independent. Thus, we emulate a multihop topology using a custom IEEE 802.15.4-TSCH network simulator, exploiting the dataset. Indeed, we use directly the transmission successes / failures to determine if a packet is received by at least one receiver. By making our evaluation independent from any implementation, we aim to first prove this anycast mechanism needs to be investigated further. The code used for our experiments is also freely available on GitHub[10].

We base our implementation on the 6TiSCH-minimal [90] to schedule multiple receivers for a given transmitter. To avoid collisions, we schedule different shared cells for each transmitter. Thus, each node has only one opportunity to transmit per slotframe repetition. We assume here that the schedule is pre-installed at compilation time. We focus specifically on links with intermediate quality (e.g. < 75% of PDR) to demonstrate how anycast transmissions can improve the network reliability.

5.3.1 Multi-neighbor efficiency

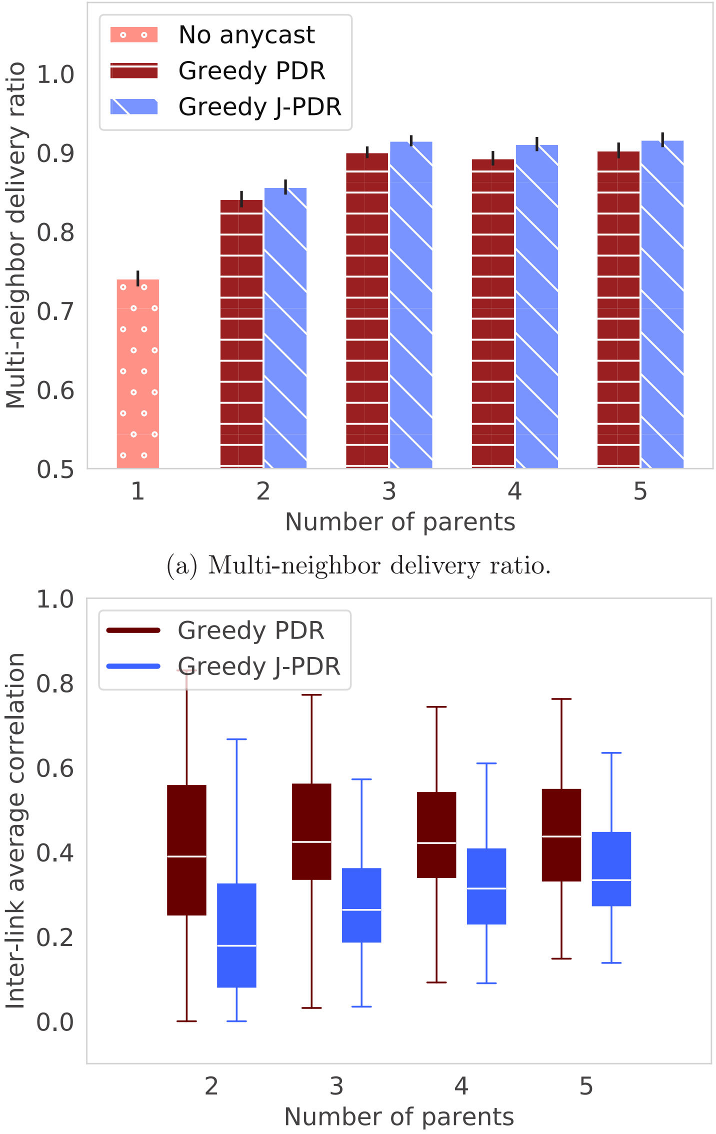
We first consider a scenario where a node has to build its parent set among multiple candidate neighbors. In this scenario, each one of the 100 deployed nodes transmits 1,000 packets in a row, while the others register all receptions. We use the first 12 bursts (240 transmissions) for training: a node computes the individual PDR and the transmissions success sequences for each neighbor. These metrics are used to select the parents. Finally, the remaining 51 bursts (1,020 packets) are used to assess the long-term performance when using this static parent set. All the experiment parameters are depicted in the Table 5.1.

We compare the performance of the two heuristics described in Section 5.2. Figure 5.4a reports the Packet Delivery Ratio obtained with the two heuristics (greedy PDR vs. j-PDR). Using a single parent (no anycast), allows the network to exploit links with a PDR of 75% on average. The two strategies perform exactly the same with one parent because they select as primary parent the neighbor with a lower rank, and with the highest PDR.

With two parents or more, anycast improves the PDR: the greedy PDR strategy achieves a PDR of 85%. Those results confirm that greedy J-PDR provides the highest reliability, since it selects the most independent set of parents. Greedy PDR tends to select neighbors that are geographically close to each other, since it considers only the PDR, and not the correlations. We can note that exploiting more than three parents has no benefit on the reliability, since only the first ones are effectively used.

[9] https://github.com/rodrigoth/anycast

[10] https://github.com/rodrigoth/anycast/tree/master/simulator



(a) Multi-neighbor delivery ratio.

(b) Average correlation.

Figure 5.4: Multi-delivery ratio and the average correlation for the two evaluated heuristics.

To measure the ability of the strategies to select independent parents, Figure 5.4b reports the correlation factors of the different parents for a given node. In approximately 75% of the cases, all the selected parents have a weak correlation average among themselves (below 0.4). This low correlation highlights that our algorithm improves the spatial diversity and consequently reduces the number of retransmissions. On the contrary, the greedy PDR strategy may select non-independent parents in some cases, leading to a poor diversity (and thus, a lower reliability gain).

Table 5.1: Experiments parameters.

Experiment	Traffic load	1000 packets
	# of nodes	100
	# Testbed	FIT-IoT Lab (Lille)
	Traffic pattern	Convergecast
TSCH	Slotframe length	101
	Timeslot duration	15 ms
	Transmissions attempts	4
	Schedule policy	6TiSCH minimal based
Radio	802.15.4 channels	11-26
	Transmission power	+3 dBm

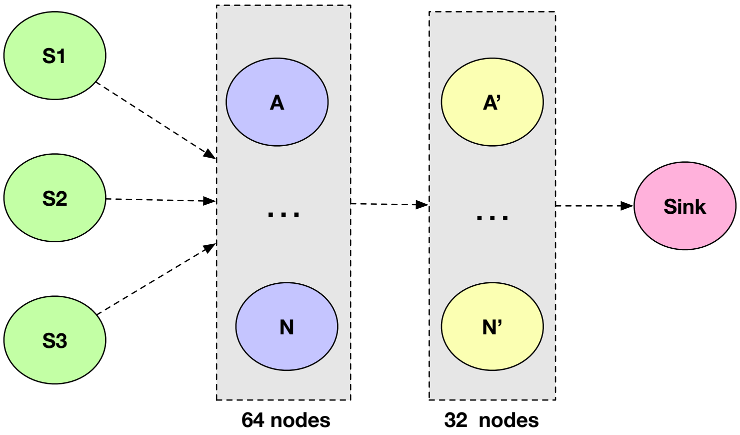


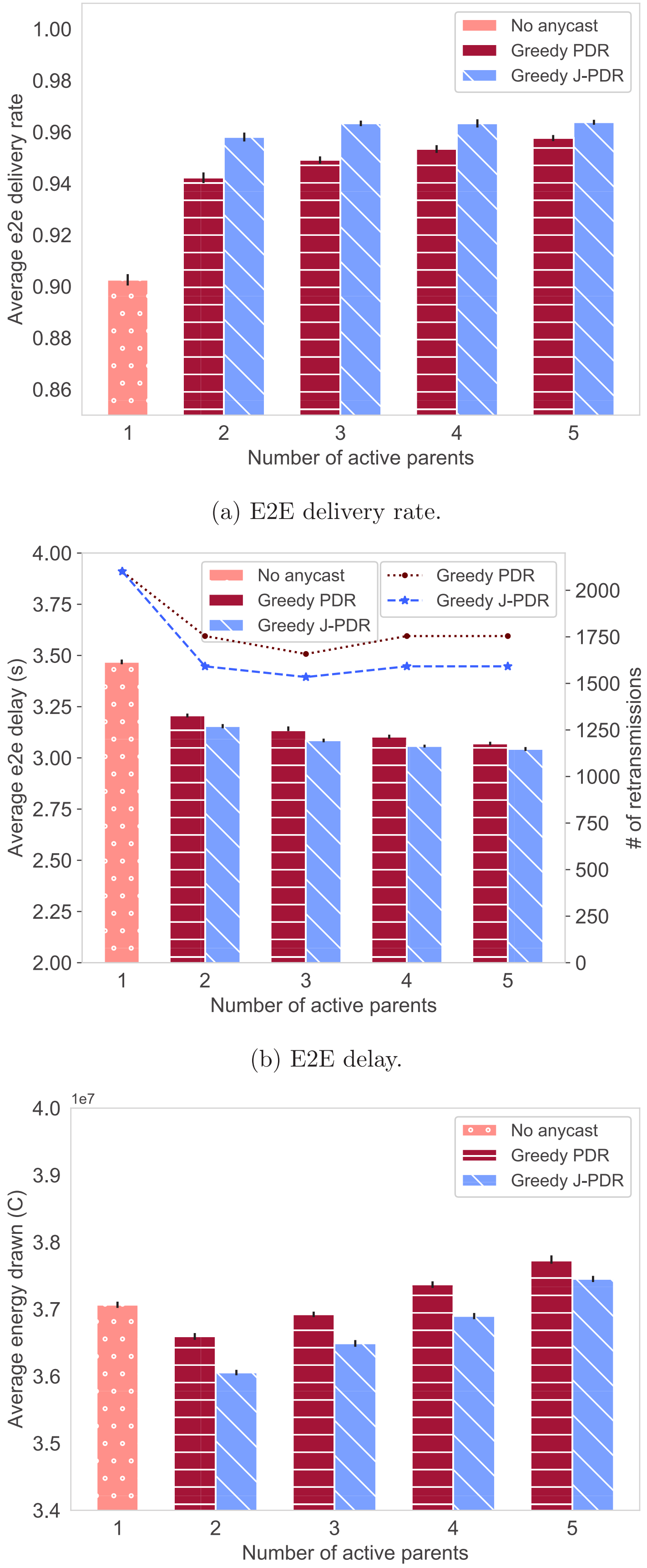
Figure 5.5: Multi-hop topology composed of 100 nodes across the 3 floors of the testbed. The leaves generate 1000 packets destined to the sink.

5.3.2 End-to-end performance

We now consider a multi-hop network to focus on end-to-end performance. We place the sink at the extreme corner of the testbed to allow multi-hop transmissions. We rely here on RPL to construct the routes and to assign ranks to each node. We obtain typically the topology illustrated in Figure 5.5. To measure more precisely the gain of using anycast in multi-hop networks, we report only the end-to-end results for the devices which are three hops away from the sink (i.e. S1, S2 and S3).

Figure 5.6a illustrates the end-to-end reliability achieved with anycast. Because anycast exploits multiple receivers, we increase significantly the end-to-end PDR when compared to the traditional unicast communication. Still, having more than three parents does only increase slightly the reliability. Greedy J-PDR keeps on providing the highest reliability, by selecting carefully independent parents, i.e. with independent packet losses. Here, anycast is efficient, and provides an end-to-end reliability of 96% when each device is authorized to retransmit at most four times the same packet.

Figure 5.6b reports the end-to-end delay. In particular, with only one parent, a



(a) E2E delivery rate.

(b) E2E delay.

(c) Energy.

Figure 5.6: End-to-end performance on a multi-hop network for 30 simulations repetitions and 95% of confidence interval.

packet is delivered after a large number of retransmissions, increasing mechanically the end-to-end delay. Indeed, with multiple parents, the nodes can exploit different links with variable conditions. Thus, a packet progresses on average *farther* with the same number of transmissions. With two parents, we reduce significantly the number of retransmissions and the delay. The difference between both heuristics becomes more evident now: the higher spatial diversity added by the J-PDR heuristic reduces the number of retransmissions and consequently the end-to-end delay.

Finally, we measure the energy cost of using anycast, reported in Figure 5.6c. We use the energy model proposed in [186], which also targets IEEE 802.15.4-TSCH networks. This model computes the charge drawn per slot taking into account the amount of time that the CPU and radio are used. We assume that the radio is the main source of energy consumption and we consider the energy spent in transmissions/receptions only. Additionally, we adjust the current values accordingly to our hardware (AT86RF231 radio). We use the timers from OpenWSN as reference to compute how long the radio is used in each timeslot.

Using anycast reduces the number of retransmissions and consequently the network saves energy. A retransmission impacts heavily the energy consumption, since in a shared cell all nodes wake-up simultaneously for a short time to listen for incoming packets, even if they are not scheduled. Thus a retransmission impacts not only the sender and the receiver(s), but all the nodes in the network. Additionally, we can remark that the energy consumption is higher for the greedy PDR strategy: the parents are not independent, and some of them stay awake uselessly.

5.4 Summary

We quantified here the interest of anycast to improve the reliability of IEEE 802.15.4-TSCH in a realistic indoor environment. In particular, we used experimental data to study in depth the correlation between packet losses. We proposed a heuristic to select a proper set of independent parents, able to improve the reliability. This greedy heuristic relies on the Joint Packet Delivery Ratio (J-PDR), denoting the actual PDR a link can provide, when considering non independent packet losses. This way, we reduce both the number of retransmissions, and the delay, when the source of the flow is several hops away from the destination. Our experimental results highlighted that each device in our indoor environment had at least two parents with independent packet losses, i.e. with a ϕ-factor inferior than 0.4.

Therefore, we conjecture that anycast transmissions can be used to restrict routing reconfigurations, discussed in Chap. 4, caused by temporary oscillations on the link quality. Indeed, short-time variations should not affect all receivers equally, if they have independent packet losses. Thus, a node can use any of its alternative routes when individual links start performing poorly without switching its preferred parent.

After having demonstrated the efficiency of anycast in terms of network reliability, a novel anycast scheduling algorithm able to work with dedicated cells in 6TiSCH still need to be proposed. One possibility is to adapt any centralized scheduling algorithm, so that several receivers can be associated to a given cell (with the same transmitter). Another possibility is to adapt a distributed scheduling function, such as SF0 [87], to take benefit from anycast. In particular, it is necessary to obtain an accurate estimation of the packet losses, using a combination of passive measurements and probes, such as the link estimator shown in Chapter 3.

It is also necessary to implement the acknowledgements in the IEEE 802.15.4-TSCH stack, so that several (ordered) receivers can be associated with a single transmission. In addition, a solution to deal with the hidden terminal problem of neighbors that cannot see each other need to be proposed. Otherwise, these nodes may forward the same packet, i.e. a duplicated packet is created.

So far, we have employed only networks with fixed devices. However, we expect more industrial applications relying on mobile devices to enhance the manufacturing process. The challenge consists in handling several routing reconfigurations caused by nodes joining and leaving the network continuously. Can we effectively support mobile devices in highly reliable low-power wireless networks? We start to address mobility aspects in the next chapter.

Chapter 6

Attachment Delay of Mobile Devices

Contents

The Internet of Mobile Things is now emerging [187], where smart objects can move independently. Smart cities now integrate more and more mobile devices, for e.g. transport and logistic applications [188]. Similarly, the healthcare industry has to support users able to move independently while assuming ultra-reliable networks.

Although mobility plays an increasingly important role for many industrial deployments [189], the IEEE 802.15.4-TSCH standard does not propose a clear approach to handle a high rate of topology changes due to the association/dissociation of mobile devices. Thus, the challenge consists in handling a set of mobile devices inside a static wireless network infrastructure Additionally, the slow-channel hopping mechanism introduces a new layer of complexity: a joining node has to wait for receiving the synchronization beacon on its active listening channel, delaying its association to the network. A fast association is a key factor to enable mobility over low-power wireless networks [190].

As shown in Chap. 2, the use of mobile devices in wireless industrial networks has already been investigated in the past [189, 191, 192]. They mainly focus on proposing mechanisms to reduce the attachment delay. Indeed, discovering the network is particularly challenging in multichannel environments, since the discovering node has to find the right channel to listen to [193]. Besides, the novel device has to reserve some transmission opportunities, using control packets. Unfortunately, these control packets are prone to collisions since they are transmitted through contention-based cells [89]. Mechanically, these collisions increase the attachment delay.

Here, we analyze this attachment delay through a Discrete-time Markov chain (DTMC), comprising both the synchronization and the negotiation of dedicated cells. In particular, since the control frames (EB and 6P) have a strong impact on the convergence, our proposed model carefully integrates the collision probability of these packets.

Contribution

1. We propose here an analytical Markov chain to model the first association of a mobile node in a multi-hop network. We consider both the discovery of a neighboring device, and the negotiation of cells;
2. We evaluate the gain of transmitting Enhanced Beacons (EB) on multiple channels in order to reduce the synchronization delay. Using multiple channels allows to spread the load on shared cells, reducing the collision probability;
3. We quantify the impact of the network density on the discovery and negotiation time. More neighbors mean also more collisions, very prejudicial to the synchronization.

6.1 Scenario and Challenges

We focus here on a network topology where the sink and a collection of relay nodes are static. Only a few devices (e.g. robots) are mobile and represent the *leaves* of the network infrastructure (Fig. 6.1). Thus, a mobile device sends its packet to a neighboring relay node, which forwards them through a path of relays to the sink. Each static node has a collection of dedicated cells in its schedule, maintained by a scheduling function such as SF0 [87]. Thus, each relay node can forward the packets from mobile devices without any collision.

Mobile devices constitute the leaves and have to identify a single neighboring relay node to send their packets. They need to capture its Enhanced Beacons, to adjust their clock and know when are the next shared cells, to be able to transmit their first messages. After selecting a next hop, a mobile node engages a 6P two-way handshake [31] to reserve dedicated cells for its transmissions. Thus, supporting mobility presents the following challenges:

Lossy links: since each device use wireless transmissions, unreliability is the norm. Even co-located networks exploiting the same ISM band may interfere, and

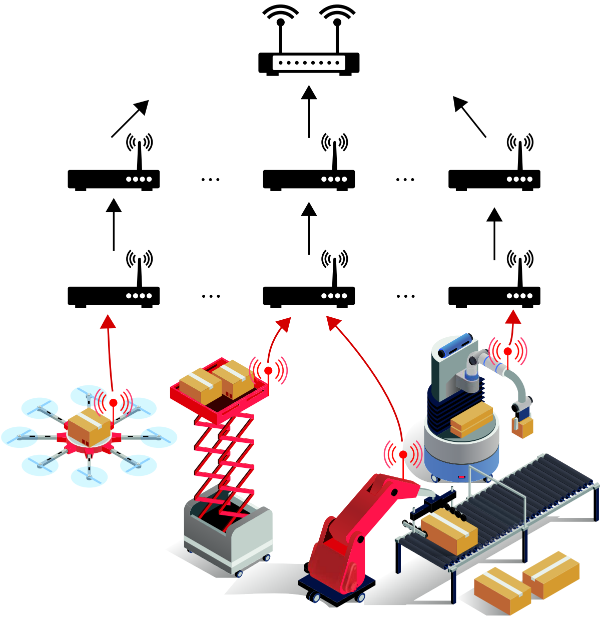


Figure 6.1: An industrial network composed of an infrastructure and a set of mobile leaves.

thus impact heavily its performance [169]. Mobile devices tend to join/leave the network or change its points of attachment frequently. In a 6TiSCH network, this topology instability generates a large number of control packets while the network re-converge. During the convergence, a network tends to present higher latency and lower reliability;

Short transmission range: for frequency re-use, and to increase the network capacity [194], a device uses a limited transmission power. However, this short range tends to increase the number of handoffs: a mobile device has to find a novel point of attachment when the link quality toward its previous parent starts to deteriorate;

Discovery delay: initially, a mobile device is unaware of the frequency hopping sequence used by the network, and has to listen passively to beacons. Unfortunately, multichannel increases very significantly the joining time, even for static devices [193]. Besides, scanning the network requires the mobile device to maintain its radio on, wasting energy;

Efficient next hop selection: a mobile device has to select a next hop with a sufficient link quality. Unfortunately, the link quality estimation is practically very expensive, and increases the attachment delay;

Negotiation delay: only dedicated cells are robust to collisions. Unfortunately, as seen in Chap. 4, negotiating cells with the next hop is expensive and takes a significant time. In particular, several handshakes are required to verify a cell is available for both the transmitter and the receiver.

6.2 Joining time model

We analyze here the joining time, i.e. the time interval between a mobile device wakes-up, and it can start transmitting data packets through dedicated cells. We focus on the discovery that a node has to trigger when it is unsynchronized. This procedure comprises:

synchronization: the joining node has to receive an Enhanced Beacon (EB) to synchronize itself with the network. Then, it gets the frequency hopping sequence and the shared cells for broadcast packets;

negotiation: the node has selected the source of the EB as parent, and then negotiates a set of dedicated cells to use to transmit its data packets.

6.2.1 Markov chain

We define here a discrete time Markov chain (Fig. 6.2) to represent the joining process of a new (mobile) node, hereafter denoted as *joining node*, when it joins the network for the first time. The joining node is initially in the *Unsynchronized* state, listening for EB sent by neighboring fixed nodes. In IEEE 802.15.4-TSCH, all synchronized nodes broadcast EB periodically to announce the existence of the network.

Upon receiving an EB on its active channel, the joining node starts negotiating dedicated cells through the 6P protocol to the infrastructure. In our model, we represent this negotiation through the state *6P Negotiation*. In addition, the model takes into account both 6P-Request and 6P-Response used by 6P (see Chap. 2). After reserving dedicated cells, the joining node finally attaches to the network.

We detail next the different parts of our model.

6.2.2 Synchronization

We make here a distinction between the two factors that impact directly the synchronization time of the joining node: EB collision and the channel hopping mechanism. Since IEEE 802.15.4-TSCH adopts a slotted Aloha mechanism for shared cells, the collision probability may be quite high. Indeed, an EB packet is enqueued until the next shared cell. Thus, when multiple nodes enqueue EB packets simultaneously between consecutive shared cells, their transmissions collide. Because of the channel hopping characteristic, an EB is successfully received by the joining node only if the latter is listening to the *right* channel.

All nodes in a IEEE 802.15.4-TSCH network enqueue EB at the same frequency after they synchronize. To reduce the amount of collisions among EB, we consider adding jitters before EB transmissions, which represents the default behavior of OpenWSN [195]. The jitter increases the time window in which a node enqueues Enhanced Beacons. For instance, for a beacon period β and jitter γ, the generation time of the next Enhanced Beacon will be randomly selected within the interval $[\beta - \gamma, \beta + \gamma]$. Therefore, collisions are less repetitive, since the nodes have now the possibility to transmit beacons at different moments.

Let us model the EB generation as a Poisson Process. Let us consider λ as the expected number of EB queued by all nodes during a given time interval of length

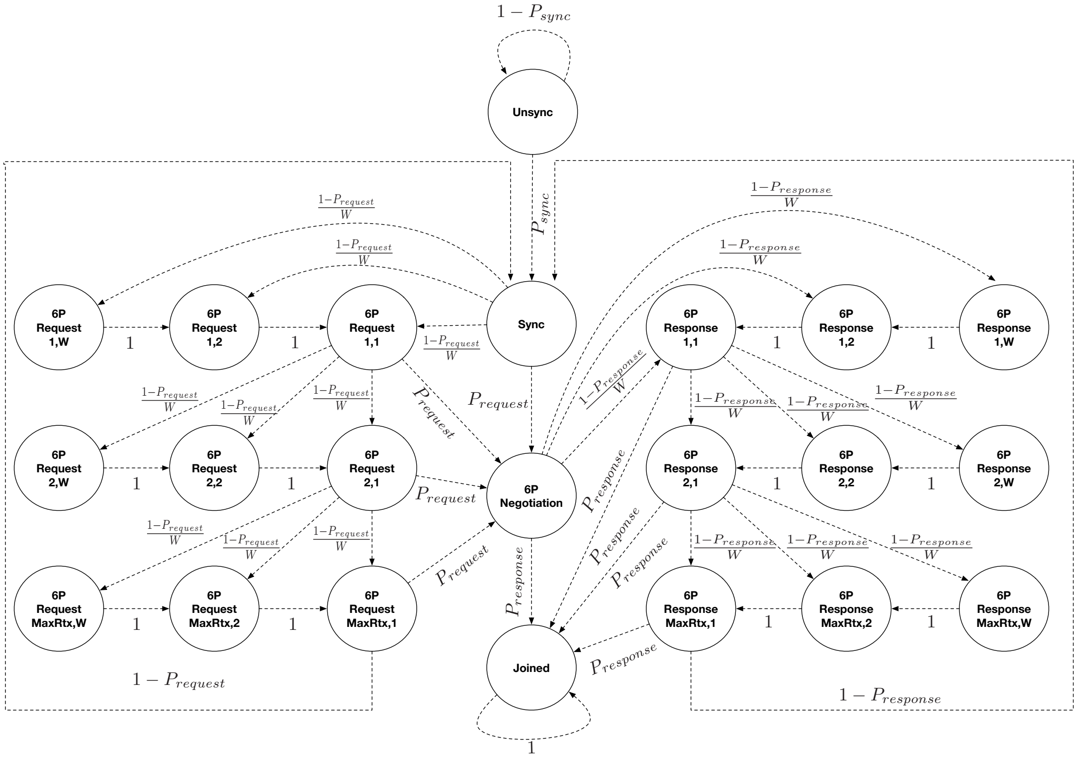


Figure 6.2: Model for the association time of a joining node.

L. Let Δt be the time between consecutive shared cells, in a such way that $L \geq \Delta t$. During the period L, the nodes have $\left\lfloor \frac{L}{\Delta t} \right\rfloor$ possibilities to enqueue their respective EB packets between consecutive shared cells. Since we assume that the rate λ is constant over the time, we can compute the rate of beacons to be enqueued during any Δt interval as:

$$\mu = \lambda * \left\lfloor \frac{L}{\Delta t} \right\rfloor^{-1} \tag{6.1}$$

The transmission is only successful when a single device enqueues an EB during a given Δt interval. For instance, Figure 6.3 depicts two colliding transmissions (from nodes B and C). It also shows that those from A and D are successful since enqueued during different Δt periods. From the Poisson distribution, the probability of having a single node generating an EB for any Δt interval is:

$$P_{beacon} = P(X = 1) = \mu e^{-\mu} \tag{6.2}$$

Additionally, we need to account the probability that the joining node is listening to the right channel. Since the frequency hopping sequence uses all the channels uniformly, the joining node has a uniform probability of matching the channel of the EB transmission. Thus, the probability of synchronization (P_{sync} in Fig. 6.2) is finally:

$$P_{sync} = P_{beacon} * \left(\frac{1}{N_{ch}} \right) \tag{6.3}$$

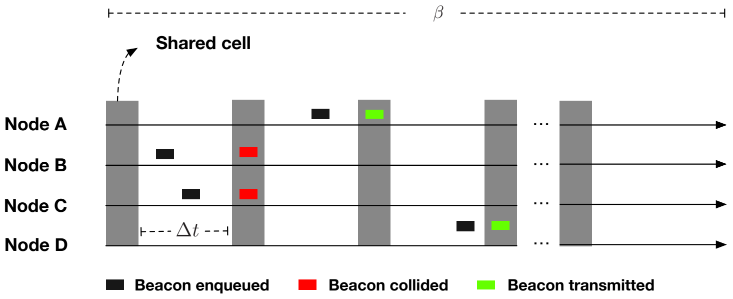


Figure 6.3: Beacon queuing over the time. A collision occurs when two or more nodes enqueue simultaneously between consecutive shared cells (nodes B and C).

where N_{ch} is the number of channels (up to 16 in 2.4 GHz). Since we consider that a packet is lost only due to collisions, using less channels decreases the synchronization time.

6.2.3 Negotiation

After having received an EB, the joining node transits to the state *6P Negotiation* (Fig. 6.2) and it starts negotiating with its parent. However, no bandwidth is yet available: it has to send 6P-Request packets and to wait for a confirmation before starting using dedicated cells for communication. Unfortunately, collisions are frequent in shared cells, since EB, routing control packets (i.e. DIO used by IETF RPL) and 6P control packets compete for the same resource.

6P uses a two-way handshake mechanism: both the request and the response are subject to transmission failures. The negotiation is successful if both the request and response are transmitted without collision. Thus, we can employ here Eq. 6.2 with $X = 0$ to compute the probability of success.

$$P_{request} = P_{response} = e^{-\mu} \tag{6.4}$$

In case of collision, the transmitter selects a random backoff value and skips the corresponding number of shared cells. We represent the backoff states for the 6P-Request and 6P-Response (Fig. 6.2) as a 2-tuple (r, w), where r is the current transmission attempt and w is the backoff counter. The probability of reaching any subsequent backoff state after a collision is equally likely. For all states (i, w), where $w > 1$, the transmitter does not try to retransmit and it transits to state $(i, w-1)$ with probability 1. After reaching a maximum number of attempts $MaxRtx$, the node discards the current packet and starts over the negotiation, i.e. go back to the *Sync* state.

6.2.4 Handover

Since mobile devices constantly move around the environment, the link between the device and its point of attachment may eventually start presenting a low reliability due to the long distance between them. In that case, the mobile device has to select a more reliable relay node to forward its packets, i.e. to perform a handover.

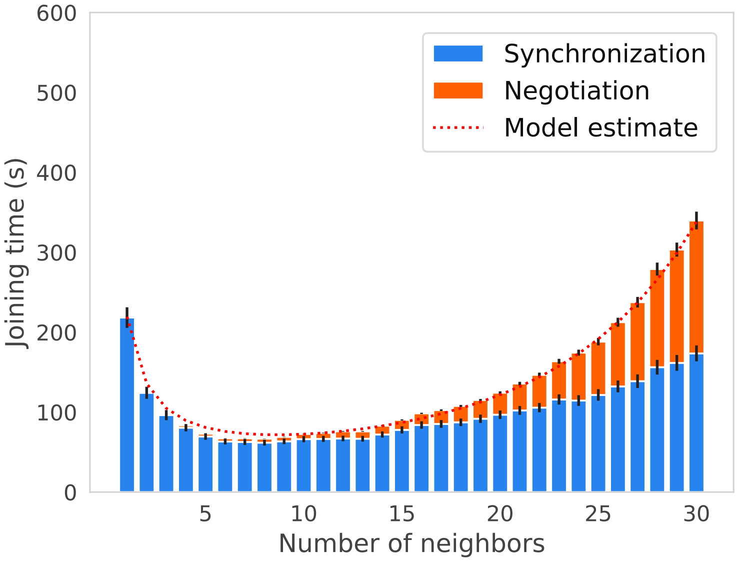


Figure 6.4: Comparison of the average joining time given by the model and simulations with an EB period of 15s, and a jitter of 200 ms

Specifically, the mobile device deallocates the cells toward its previous next hop, and negotiates novel cells with its novel relay node.

We can neglect the deallocation time, as the 6P packets will be transmitted without contention, i.e. dedicated cells already exist in their schedule. On the other hand, novel cells have to be negotiated with the novel relay node. Since the mobile node is already synchronized (*Sync* state in Fig. 6.2), it can immediately engage a negotiation. Thus, we can employ Equation 6.4 to compute the probability that the negotiation will succeed.

6.2.5 Estimating the joining time

Since we rely on an absorbing Markov chain, we can estimate the joining time by computing the average number of steps to reach the absorbing state from the initial state *Unsync*. In our DTMC depicted in Figure 6.2, the *Joined* state is the absorbing state. Every step in our model represents the interval between two consecutive shared cells (i.e. Δt). We rely on the Fundamental Matrix to compute the average absorbing time: i.e. number of transitions from the initial state (*Unsync*) to the absorbing state (*Joined*).

6.3 Numerical Analysis

We propose first to verify the accuracy of our DTMC model when estimating the joining time in 6TiSCH networks. Then, we will analyze the joining time for a joining node, as well as assessing the gain of using multiple channels for EB transmissions.

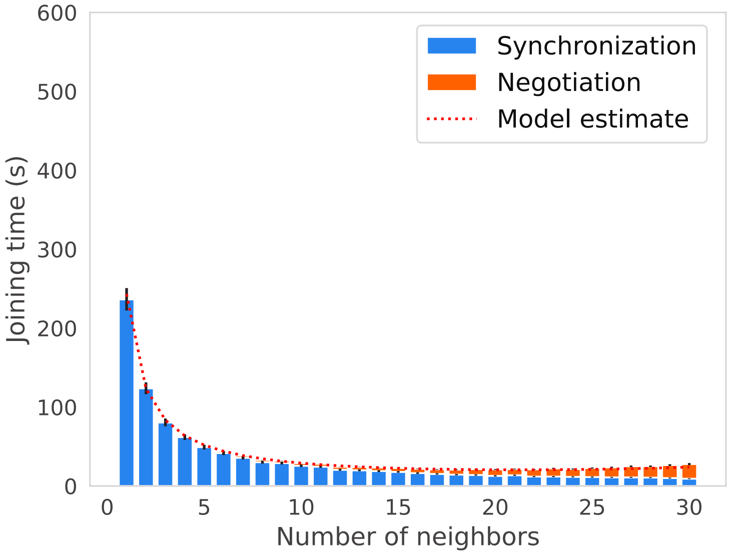


Figure 6.5: Impact of the multi-channel EB broadcasting on the joining time.

6.3.1 Model validation

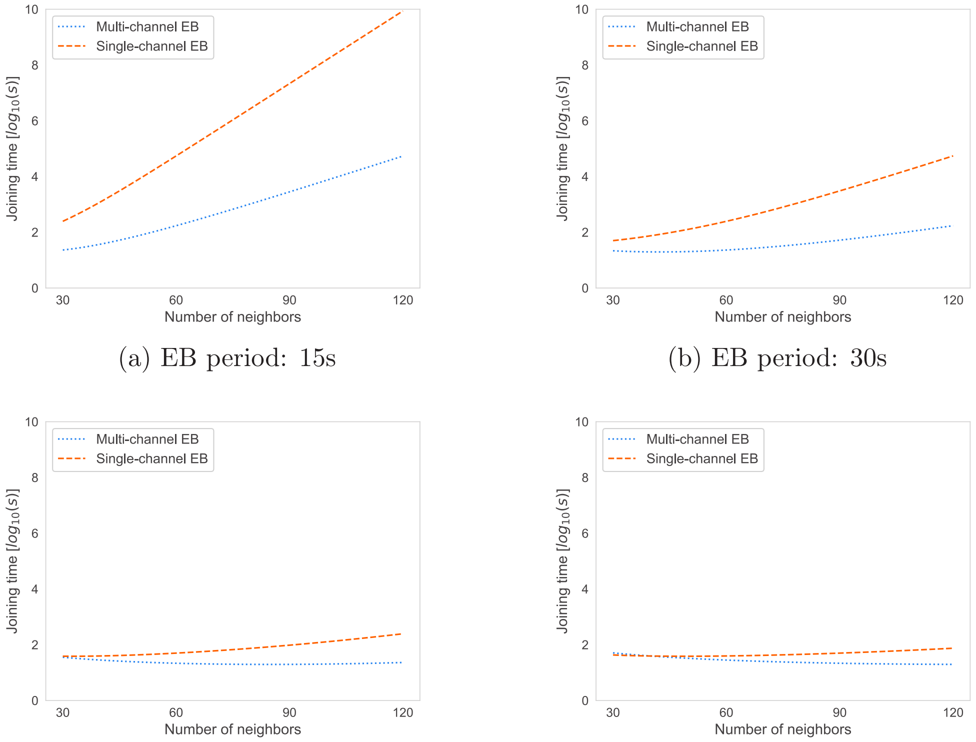
We rely on simulations to validate our DTMC model. We implement a lightweight 6TiSCH simulator written in Python focusing exclusively on the joining procedure of a mobile node. Our simulator is freely available on GitHub[11]

In our scenarios, we consider an existing network composed of fixed nodes (i.e. the infrastructure) and one joining node. The fixed nodes broadcast EB and DIO regularly during shared cells. For sake of simplicity, we assume that the infrastructure has enough bandwidth to accommodate the novel flows. Thus, only the joining node and its point of attachment have to negotiate dedicated cells. Additionally, we assume perfect links conditions. Thus, collisions are the only causes of packet drops.

We employ a slotframe composed of 101 timeslots, and 26 channel offsets, with two shared cells placed uniformly in the slotframe. The joining node selects randomly one channel to listen for EB. All nodes in the infrastructure enqueue EB and DIO every 15 seconds in average, considering a jitter of 200 ms. We plot systematically the 95% confidence intervals.

The comparison between our analytical model and simulation results is depicted in Figure 6.4. We perform 1,000 repetitions for each number of neighbors to make our results more representative. We observe that the analytical values fit very well the simulation results. As expected, with few nodes, the synchronization takes longer, since the joining node has a smaller probability to receive a valid EB. On the other hand, the negotiation is fast, since there are less competition in the shared cells. Increasing the number of neighbors improves the joining time to a certain extent (i.e. 9 neighbors in our scenario). For higher values, the probability of collision increases impacting both the synchronization and negotiation times. Thus, the joining time presents an exponential growth.

[11] https://github.com/rodrigoth/Simulator/tree/optimized



(a) EB period: 15s

(b) EB period: 30s

(c) EB period: 60s

(d) EB period: 90s

Figure 6.6: Impact of the EB period on different network densities.

6.3.2 Multi-channel EB to reduce the attachment delay

By definition, all transmissions in shared cells occur exclusively in a single channel offset. Therefore, the probability of collision increases, since only one cell is used for transmissions. In addition, the slow channel hopping mechanism increases the channel diversity, making the network announcements to occur on a single and different frequency each time. We propose to assess the gain of the Random Filling approach [196] to transmit EB on multiple channels, and thus, to reduce the synchronization time. More precisely, they allocate exclusive timeslots for EB transmissions, where the channel offsets are selected randomly among the available ones.

We redefine Equation 6.3 to account simultaneous transmissions on different channels. Now, on each channel, EB arrival follows a Poisson Process of parameter μ/N_{ch}, since the EB are uniformly distributed on all N_{ch} channels. Hence,the probability that only one EB is transmitted on the channel where the joining node is listening to is:

$$P_{sync} = \left(\frac{\mu}{N_{ch}}\right) e^{\left(-\frac{\mu}{N_{ch}}\right)} \tag{6.5}$$

Figure 6.5 highlights the gain of broadcasting EB on multiple channels. The synchronization time decreases heavily compared to the single channel case depicted in Figure 6.4. In a general way, we can expect less EB arrivals, but the EB are now transmitted on multiple channels simultaneously. Thus, the probability that the joining node is listening to any of the transmitted channels increases. In addition, spreading EB on multiple channel impacts directly the negotiation time, since the EB and the 6P packets are transmitted mostly on different channels. We can now accommodate much larger densities with a very reasonable attachment delay.

6.3.3 Large scale performance

Finally, we analyze the impact of the EB period on the joining time with different densities. For the sake of simplicity, we consider that the EB and DIO are transmitted at the same frequency.

Figure 6.6 compares the joining time when EB are transmitted on single vs. multiple channels. As expected the EB frequency has a crucial importance on the joining time on large scale deployments. However, shorter EB periods increase the probability of collisions when using a single channel. Using multiple channels allows to spread the load, and thus reduces significantly the collision probability.

We can note that the optimal EB period, minimizing the joining time, depends on the density. More nodes mean a larger number of EB transmissions, and thus a larger optimal EB period.

6.4 Summary

In this chapter, we focused on the joining time of a mobile node when it joins the network for the first time. Mobile devices have to fast attach to the network, and reserve some resources for their critical flows. We modeled the joining process using a Discrete Time Markov Chain. Our model takes into account both the synchronization and negotiation times in 6TiSCH networks.

Our simulations demonstrate the accuracy of our DTMC model to estimate finely the synchronization and negotiation time. Obviously, dense networks mean a larger number of collisions, which impact very negatively the synchronization time. Even worse, negotiating dedicated cells is also very expensive, since the collision rate for control packets is very high. We also use our DTMC model to assess the gain of using multiple channels. By spreading the EB on the different channels, the collision rate is significantly reduced, improving the scalability.

After having demonstrated the accuracy of our model, an extension to consider a scenario with multiple mobile nodes joining at the same time still need to be proposed. In this case, we need to recompute the probabilities by considering that EB and DIO can now collide with 6P packets. In addition, since multiple nodes can start negotiating at the same time, 6P packets can also collide with other 6P packets from neighbors nodes.

Chapter 7

Conclusion and Future Research Directions

We conclude now this thesis, highlighting our main scientific contributions and pointing out future research directions related to the work presented here.

7.1 Conclusion

The goal of this work was to improve the reliability of low-power wireless networks in indoor scenarios, where obstacles and external interference are the rule. In particular, we proposed improvements for standards widely used in IIoT to increase the network robustness to abrupt variations on the radio conditions. Since 6TiSCH has become the main standard for IIoT, we focused our research efforts on it. However, to a certain extent, our solutions can be adapted to other standards like WirelessHART and ISA 100.11a.

We started by demonstrating an existing correlation between the unicast Packet Delivery Ratio (PDR) and the reception rate of control packets to propose a passive link quality estimator. Differently from existing works, which rely on probing to estimate the link quality to an inactive neighbor (i.e. no direct connection), our solution used only existing traffic to identify good candidates. Our approach was based on the ranking of nodes according to the reception rate of their control packets. Thus, top ranked neighbors are considered as the best candidates. Our experiments on a large scale testbed demonstrated that this classification scheme is enough for achieving an accurate selection of a next-hop neighbor in the presence of link quality degradation.

Next, we performed an experimental analysis of the performance stability of 6TiSCH networks in two indoor testbeds with different channel characteristics. We noticed that the Routing Protocol for Low-Power and Lossy Networks (RPL) protocol introduces frequent routing reconfiguration to deal with time-varying link conditions. We showed also how these routing reconfigurations can be prejudicial to the network reliability when employing a reservation-base MAC layer. In addition, we exploited the existing correlation between the control packets reception rate and the unicast PDR to create a two-step parent selection, avoiding bad choices leading to instabilities. Finally, we simplified the schedule consistency management between two nodes to reduce the instability caused by renegotiating from scratch all the cells when they detect a schedule inconsistency. We obtained finally a network that converges faster and that reacts accurately during moments of instabilities.

In Chap. 5, we evaluated the interest of employing anycast at the link-layer. We first discussed the main limitation of employing a classical unicast transmissions, and its impacts on the network performance. We also detailed how to implement anycast transmissions in 6TiSCH using only shared cells. In addition, we proposed a heuristic to select a set of independent parents, increasing the chance of reception by at least one of them. Our experiments based on data traces collected in an indoor environment demonstrated that our heuristic was able to reduce the number of retransmissions, and consequently the transmission delay.

Finally, we proposed a Discrete-Time Markov Chain (DTMC) to model the association time of a mobile node when it joins the network for the first time. We discussed the main difficulties of employing mobility in highly reliable low-power wireless networks. In particular, the impact of the slow-channel hopping and the collision of control packets (Enhanced Beacons and 6P) on the association time. Additionally, we verified the accuracy of our model to estimate finely both the synchronization and negotiation times in networks with different densities. We also employed our model to assess the gain of transmitting synchronization beacons on multiple-channels. By doing this, we demonstrated that the synchronization becomes much faster, and the collision rate is significantly reduced.

7.2 Research Directions

We discuss now some research directions to continue the work in this thesis.

7.2.1 Standard evaluation methodology

In this work, we spent most of our research efforts by performing real experiments on testbeds. Although testbeds allow researchers to test their solutions in realistic conditions, they bring some issues related to the validation of the results. The main challenge is how can we make our experiments reproducible so other researchers can replicate them? Usually, in testbed deployments we do not have any control of the environment, making our results to present some level of variability. Indeed, uncontrolled and unpredictable factors (people, interference) perform like random variables affecting differently our experiments.

In general, we gave our best to provide the maximum amount of information for replication and validation purposes. Although providing detailed experiment description, making the source code and the dataset available improve the reproducibility, we still need to define a standard way for performing experiments when facing uncontrolled environments. For instance, which metrics should be computed? How long an experiment should last? How many repetitions until getting representative results? The main challenge is that very few works provide detailed description of their experimental campaigns [150], complicating the reproducibility and comparability. Therefore, initiatives like IoT Bench [197] are fundamental for defining a set of tools and practices for accurate performance evaluations. However, more research effort should be devoted for a larger adoption of standard evaluation methods.

7.2.2 Improve mobility support

We gave the first step to investigate mobility aspects in highly reliable low-power wireless networks, considering both the synchronization and negotiation of new dedicated cells. While we targeted only the attachment delay, an accurate scheduling method to deal both with static and mobile nodes has to be proposed. In particular, we envision reserving dedicated cells to relay the traffic among the wireless routers, while a different set of dedicated or even shared cells being only used for mobile devices. In addition, we believe that the scheduling function has to pre-reserve cells for the new flows. This pre-reservation scheme would make the system more scalable, avoiding negotiation delays. The main challenge here is how to define the amount of bandwidth to pre-reserve without introducing too many idle cells. Indeed, idle cells reduce the network capacity, and they also increase the energy consumption.

Additionally, we need to propose an handover scheme, so that a mobile device can maintain several next hops. In particular, we need an autonomous solution where a node can precisely decide the best moment for performing the handover, before it starts dropping packets due to the link degradation with its relay node. For this, we have to use passive mechanisms to evaluate in real-time the link quality of all potential next hop nodes. For instance, our passive method discussed in Chapter 3 can be employed to rank the most appropriate neighbors. On the other hand, we may use anycast transmissions so that multiple fixed nodes can be used as potential receivers, without the need of performing handover.

7.2.3 Millimeter wave bands

So far, we focused only on using the ISM band (sub-6GHz) for communicating, since it represents an easy and costless way of deploying low-power wireless networks. As we showed in this thesis, the ISM band requires the use of mechanisms to mitigate the impact of external interference caused by a large and increasing number of devices that rely on this band for communicating. We envision even more devices and wireless technologies relying on the ISM band in the coming years. Additionally, more advanced kinds of industrial applications requiring vision technologies are expected in the future [198]. Unfortunately, sub-6GHz can not provide the required data rate and latency constraints for such applications [199].

Therefore, we believe that Millimeter wave (mmWave) can potentially open a new way for future industrial networks due to the large bandwidth available at this spectrum. Indeed, migrating a IEEE 802.15.4-TSCH network from 2.4 GHz to 60 GHz would make the channel hopping mechanism to be implemented over 225 channels, having a bit rate of 10 Mb/s per channel [200]. Besides, we expect much less interference at mmWave bands when compared to sub-6GHz bands. However, mmWave brings additionally challenges to handle. For instance, a higher energy consumption due to multiple antennas to send and receive signals, a high signal attenuation caused by obstacles and a shorter communication range. Therefore, beamforming mechanisms are needed to provide an efficient mmWave communication, allowing directional transmissions. Directional transmissions may reduce the signal strength at nodes not involved in the current transmission, limiting the broadcasting. We believe that scheduling algorithms can exploit this reduced interference to increase the spatial diversity, i.e. multiple nodes can be scheduled without interfering with each other.

Bibliography

[1] Biljana L Risteska Stojkoska and Kire V Trivodaliev. “A review of Internet of Things for smart home: Challenges and solutions”. In: *Journal of Cleaner Production* 140 (2017), pp. 1454–1464.

[2] Luis Sanchez et al. “SmartSantander: IoT experimentation over a smart city testbed”. In: *Computer Networks* 61 (2014), pp. 217–238.

[3] Mark Hung. “Leading the iot, gartner insights on how to lead in a connected world”. In: *Gartner Research* (2017), pp. 1–29.

[4] Brahmjit Singh. “The Internet of Things: A Vision for Smart World”. In: *Advances in Signal Processing and Communication*. Springer, 2019, pp. 165–172.

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

[6] A Larsan Aro Brian, L Arockiam, and PDSK Malarchelvi. “An IOT based secured smart library system with NFC based book tracking”. In: *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)* 11.5 (2014).

[7] Jaime Lloret, Miguel Garcia, Diana Bri, and Sandra Sendra. “A wireless sensor network deployment for rural and forest fire detection and verification”. In: *sensors* 9.11 (2009), pp. 8722–8747.

[8] Milica Pejanović Đurišić, Zhilbert Tafa, Goran Dimić, and Veljko Milutinović. “A survey of military applications of wireless sensor networks”. In: *2012 Mediterranean conference on embedded computing (MECO)*. IEEE. 2012, pp. 196–199.

[9] 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.

[10] Jaime Lloret, Sandra Sendra, Miguel Garcia, and Ginés Lloret. “Group-based underwater wireless sensor network for marine fish farms”. In: *IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE. 2011, pp. 115–119.

[11] *IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Std 802.15.4. 2006.

[12] Gabriel Montenegro, Nandakishore Kushalnagar, Jonathan Hui, and David Culler. *Transmission of IPv6 packets over IEEE 802.15. 4 networks*. Tech. rep. 2007.

[13] Ridha Soua and Pascale Minet. "A survey on energy efficient techniques in wireless sensor networks". In: *2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)*. IEEE. 2011, pp. 1–9.

[14] Tim Winter et al. *RPL: IPv6 routing protocol for low-power and lossy networks*. Tech. rep. 2012.

[15] Oana Iova, Pietro Picco, Timofei Istomin, and Csaba Kiraly. "Rpl: The routing standard for the internet of things... or is it?" In: *IEEE Communications Magazine* 54.12 (2016), pp. 16–22.

[16] Tao Liu and Alberto E Cerpa. "TALENT: Temporal adaptive link estimator with no training". In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM. 2012, pp. 253–266.

[17] 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.

[18] B. Galloway and G. P. Hancke. "Introduction to Industrial Control Networks". In: *IEEE Communications Surveys Tutorials* 15.2 (2013), pp. 860–880. ISSN: 1553-877X. DOI: 10.1109/SURV.2012.071812.00124.

[19] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman. "Communication systems for building automation and control". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1178–1203. ISSN: 0018-9219. DOI: 10.1109/JPROC.2005.849726.

[20] Borja Martinez, Cristina Cano, and Xavier Vilajosana. "A square peg in a round hole: The complex path for wireless in the manufacturing industry". In: *IEEE Communications Magazine* 57.4 (2019), pp. 109–115.

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

[22] Andrew Chatha. "Fieldbus: The foundation for field control systems". In: *Control Engineering* 41.6 (1994), pp. 77–80.

[23] Gang Zhao. "Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey." In: *Network Protocols & Algorithms* 3.1 (2011), pp. 46–63.

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

[25] Jianping Song, Song Han, A.K. Mok, Deji Chen, M. Lucas, and M. Nixon. "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control". In: *Proceedings of Real-Time and Embedded Technology and Applications Symposium (RTAS)*. Apr. 2008, pp. 377–386. DOI: 10.1109/RTAS.2008.15.

[26] ANSI/ISA-100.11a-2011. *Wireless systems for industrial automation: Process control and related applications.* ISA. 2011.

[27] Kristofer S. J. Pister and Lance Doherty. "TSMP: Time Synchronized Mesh Protocol". In: *Proceedings of IASTED International Symposium on Distributed Sensor Networks (DSN).* 2008, pp. 391–398.

[28] Thomas Watteyne, Ankur Mehta, and Kris Pister. "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense". In: *Proceedings of ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN).* Tenerife, Canary Islands, Spain, 2009, pp. 116–123.

[29] *IPv6 over the TSCH mode of IEEE 802.15.4e.* `https://datatracker.ietf.org/wg/6tisch`.

[30] Zach Shelby, Klaus Hartke, and Carsten Bormann. *The constrained application protocol (CoAP).* Tech. rep. 2014.

[31] Qin Wang, Xavier Vilajosana, and Thomas Watteyne. *6TiSCH Operation Sublayer (6top) Protocol (6P).* Tech. rep. 2018.

[32] J. Ye, B. Chen, Q. Liu, and Y. Fang. "A precision agriculture management system based on Internet of Things and WebGIS". In: *International Conference on Geoinformatics.* IEEE. 2013, pp. 1–5. DOI: `10.1109/Geoinformatics.2013.6626173`.

[33] "Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications". In: *ETSI TC ERM, TR 103 055, v1.1.1* (Sept. 2011).

[34] Soo Young Shin. "Throughput analysis of IEEE 802.15.4 network under IEEE 802.11 network interference". In: *International Journal of Electronics and Communications* 67.8 (2013), pp. 686 –689. ISSN: 1434-8411. DOI: `10.1016/j.aeue.2013.02.007`.

[35] Razvan Musaloiu; E. and Andreas Terzis. "Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks". In: *Int. J. Sen. Netw.* 3.1 (Dec. 2008), pp. 43–54. ISSN: 1748-1279. DOI: `10.1504/IJSNET.2008.016461`. URL: `http://dx.doi.org/10.1504/IJSNET.2008.016461`.

[36] A. Gonga, O. Landsiedel, P. Soldati, and M. Johansson. "Revisiting Multichannel Communication to Mitigate Interference and Link Dynamics in Wireless Sensor Networks". In: *Proceedings of IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS).* 2012, pp. 186–193. DOI: `10.1109/DCOSS.2012.15`.

[37] Vasileios Kotsiou, Georgios Z Papadopoulos, Dimitrios Zorbas, Periklis Chatzimisios, and Fabrice Theoleyre. "Blacklisting-Based Channel Hopping Approaches in Low-Power and Lossy Networks". In: *IEEE Communications Magazine* 57.2 (2019), pp. 48–53.

[38] Victor Sarker, Jorge Queralta, Tuan Nguyen, Hannu Tenhunen, and Tomi Westerlund. "A Survey on LoRa for IoT: Integrating Edge Computing". In: *International Workshop on Smart Living with IoT, Cloud and Edge Computing.* IEEE. 2019.

[39] Evgeny Khorov, Andrey Lyakhov, Alexander Krotov, and Andrey Guschin. "A survey on IEEE 802.11 ah: An enabling networking technology for smart cities". In: *Computer Communications* 58 (2015), pp. 53–69.

[40] Juan Carlos Zuniga and Benoit Ponsard. "Sigfox system description". In: *LPWAN IETF97, Nov. 14th* 25 (2016).

[41] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. "NB-IoT system for M2M communication". In: *IEEE Wireless Communications and Networking Conference*. 2016, pp. 1–5.

[42] 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.

[43] Aravind Iyer, Catherine Rosenberg, and Aditya Karnik. "What is the right model for wireless channel interference?" In: *IEEE Transactions on Wireless Communications* 8.5 (2009), pp. 2662–2671.

[44] Ugo Maria Colesanti, Carlo Crociani, and Andrea Vitaletti. "On the accuracy of omnet++ in the wireless sensornetworks domain: simulation vs. testbed". In: *Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM. 2007, pp. 25–31.

[45] 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.

[46] V C Gungor and G P Hancke. "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches". In: *IEEE Transactions on Industrial Electronics* 56.10 (2009), pp. 4258–4265.

[47] Marco Zimmerling, Pratyush Kumar, Federico Ferrari, Luca Mottola, and Lothar Thiele. *Energy-efficient Real-time Communication in Multi-hop Low-power Wireless Networks*. Tech. rep. 356. ETH Zurich, Laboratory TIK, 2014.

[48] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, and M. Gidlund. "Assignment of Segmented Slots Enabling Reliable Real-Time Transmission in Industrial Wireless Sensor Networks". In: *IEEE Transactions on Industrial Electronics* 62.6 (2015), pp. 3966–3977. ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2402642.

[49] A. Willig, K. Matheus, and A. Wolisz. "Wireless Technology in Industrial Networks". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1130–1151. ISSN: 0018-9219. DOI: 10.1109/JPROC.2005.849717.

[50] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. "A High-throughput Path Metric for Multi-hop Wireless Routing". In: *Wirel. Netw.* 11.4 (July 2005), pp. 419–434. ISSN: 1022-0038. DOI: 10.1007/s11276-005-1766-z. URL: http://dx.doi.org.insis.bib.cnrs.fr/10.1007/s11276-005-1766-z.

[51] 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: *International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM. Urbana-Champaign, IL, USA, 2005, pp. 414–425. ISBN: 1-59593-004-3. DOI: 10.1145/1062689.1062741.

[52] M. Eriksson and T. Olofsson. "On Long-Term Statistical Dependences in Channel Gains for Fixed Wireless Links in Factories". In: *IEEE Transactions on Communications* 64.7 (2016), pp. 3078–3091. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2016.2563431.

[53] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. "An Empirical Study of Low-power Wireless". In: *ACM Transactions on Sensor Networks* 6.2 (Mar. 2010), 16:1–16:49. ISSN: 1550-4859. DOI: 10.1145/1689239.1689246.

[54] Bilel Romdhani, Dominique Barthel, and Fabrice Valois. "Exploiting Asymmetric Links in a Convergecast Routing Protocol for WSNs". In: (2011).

[55] Simone Grimaldi, Mikael Gidlund, Tomas Lennvall, and Filip Barać. "Detecting communication blackout in industrial Wireless Sensor Networks". In: *2016 IEEE World Conference on Factory Communication Systems (WFCS)*. IEEE. 2016, pp. 1–8.

[56] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005. ISBN: 0-52184527-0.

[57] S. Wijetunge, U. Gunawardana, and R. Liyanapathirana. "Wireless Sensor Networks for Structural Health Monitoring: Considerations for communication protocol design". In: *Proceedings of International Conference on Telecommunications*. 2010, pp. 694–699. DOI: 10.1109/ICTEL.2010.5478798.

[58] Liyang Yu, Neng Wang, and Xiaoqiao Meng. "Real-time forest fire detection with wireless sensor networks". In: *Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing*. Vol. 2. 2005, pp. 1214–1217. DOI: 10.1109/WCNM.2005.1544272.

[59] Pei Huang, Li Xiao, Soroor Soltani, Matt W Mutka, and Ning Xi. "The evolution of MAC protocols in wireless sensor networks: A survey". In: *IEEE communications surveys & tutorials* 15.1 (2012), pp. 101–120.

[60] Joseph Polastre, Jason Hill, and David Culler. "Versatile low power media access for wireless sensor networks". In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM. 2004, pp. 95–107.

[61] Vasileios Kotsiou, Georgios Z Papadopoulos, Periklis Chatzimisios, and Fabrice Tholeyre. "Is local blacklisting relevant in slow channel hopping low-power wireless networks?" In: *IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6.

[62] T. Watteyne, S. Lanzisera, A. Mehta, and K. S. J. Pister. "Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks". In: *Proceedings of IEEE International Conference on Communications (ICC)*. 2010, pp. 1–5. DOI: 10.1109/ICC.2010.5502548.

[63] A. Farrel, J.-P. Vasseur, and J. Ash. *A Path Computation Element (PCE)-Based Architecture*. RFC 4665. IETF, 2006.

[64] A. Koubaa, A. Cunha, and M. Alves. “A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks”. In: *Proceedings of 19th Euromicro Conference on Real-Time Systems (ECRTS)*. 2007, pp. 125–135. DOI: 10.1109/ECRTS.2007.82.

[65] N. Abdeddaim, F. Theoleyre, F. Rousseau, and A. Duda. “Multi-Channel Cluster Tree for 802.15.4 Wireless Sensor Networks”. In: *Proceedings of IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2012, pp. 590–595. DOI: 10.1109/PIMRC.2012.6362853.

[66] N. Abdeddaim, F. Theoleyre, M. Heusse, and A. Duda. “Adaptive IEEE 802.15.4 MAC for Throughput and Energy Optimization”. In: *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2013, pp. 223–230. DOI: 10.1109/DCOSS.2013.44.

[67] *IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. 2012.

[68] W. y. Lee, K. i. Hwang, Y. A. Jeon, and S. Choi. “Distributed Fast Beacon Scheduling for Mesh Networks”. In: *Proceedings of IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*. 2011, pp. 727–732. DOI: 10.1109/MASS.2011.79.

[69] Mark Nixon and T Round Rock. “A Comparison of WirelessHART and ISA100. 11a”. In: *Whitepaper, Emerson Process Management* (2012), pp. 1–36.

[70] J. Lee, T. Kwon, and J. Song. “Group Connectivity Model for Industrial Wireless Sensor Networks”. In: *IEEE Transactions on Industrial Electronics* 57.5 (2010), pp. 1835–1844. ISSN: 0278-0046. DOI: 10.1109/TIE.2009.2033089.

[71] Marcelo Nobre, Ivanovitch Silva, and Luiz Affonso Guedes. “Routing and Scheduling Algorithms for WirelessHART Networks: A Survey”. In: *MDPI Sensors* 15.5 (2015), p. 9703. ISSN: 1424-8220. DOI: 10.3390/s150509703.

[72] Sinem Coleri Ergen and Pravin Varaiya. “TDMA scheduling algorithms for wireless sensor networks”. In: *Springer Wireless Networks* 16.4 (2010), pp. 985–997. ISSN: 1572-8196. DOI: 10.1007/s11276-009-0183-0.

[73] John N. Tsitsiklis and Kuang Xu. “On the Power of (Even a Little) Centralization in Distributed Processing”. In: *ACM SIGMETRICS Performance Evaluation Review* 39.1 (June 2011), pp. 121–132. ISSN: 0163-5999. DOI: 10.1145/2007116.2007131.

[74] 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: *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2012, pp. 327–332. ISBN: 9781467325691. DOI: 10.1109/PIMRC.2012.6362805.

[75] Mr Palattella and N Accettura. “On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH”. In: *IEEE Sensors Journal* (2013), pp. 1–21. ISSN: 1530-437X. DOI: `10.1109/JSEN.2013.2266417`.

[76] Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, and Fabrice Valois. “High-Reliability Scheduling in Deterministic Wireless Multi-hop Networks”. In: *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Sept. 2016, pp. 1–6. DOI: `10.1109/PIMRC.2016.7794839`.

[77] Atis Elsts, Xenofon Fafoutis, James Pope, George Oikonomou, Robert Piechocki, and Ian Craddock. “Scheduling High-Rate Unpredictable Traffic in IEEE 802.15.4 TSCH Networks”. In: *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2017.

[78] Ridha Soua, Pascale Minet, and Erwan Livolant. “MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks”. In: *Proceedings of IEEE 31st International Performance Computing and Communications Conference (IPCCC)*. 2012, pp. 91–100. ISBN: 9781467348812. DOI: `10.1109/PCCC.2012.6407742`.

[79] Carlene E.-A. Campbell, Shafiullah Khan, Dhananjay Singh, and Kok-Keong Loo. “Multi-Channel Multi-Radio Using 802.11 Based Media Access for Sink Nodes in Wireless Sensor Networks”. In: *MDPI Sensors* 11.5 (2011), p. 4917. ISSN: 1424-8220. DOI: `10.3390/s110504917`.

[80] Ruan D. Gomes, Diego V. Queiroz, Abel C. Lima Filho, Iguatemi E. Fonseca, and Marcelo S. Alencar. “Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes”. In: *Elsevier Ad Hoc Networks* 59 (2017), pp. 116 –133. ISSN: 1570-8705. DOI: `https://doi.org/10.1016/j.adhoc.2017.02.007`.

[81] Sergio Montero, Javier Gozalvez, Miguel Sepulcre, and Gonzalo Prieto. “Impact of mobility on the management and performance of WirelessHART industrial communications”. In: *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2012, pp. 1–4. ISBN: 9781467347372. DOI: `10.1109/ETFA.2012.6489704`.

[82] Sirajum Munir, Shan Lin, Enamul Hoque, S. M. Shahriar Nirjon, John A. Stankovic, and Kamin Whitehouse. “Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks”. In: *Proceedings of IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*. 2010, pp. 303–314. DOI: `10.1145/1791212.1791248`.

[83] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. “IEEE 802.15.4e: A survey”. In: *Elsevier Computer Communications* 88 (2016). ISSN: 0140-3664. DOI: `http://dx.doi.org/10.1016/j.comcom.2016.05.004`.

[84] 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. ISSN: 23274662. DOI: `10.1109/JIOT.2015.2476915`. eprint: `9605103` (cs).

[85] Antoni Morell, Xavier Vilajosana, José López Vicario, and Thomas Watteyne. "Label switching over IEEE802.15.4e networks". In: *Transactions on Emerging Telecommunications Technologies* 24.5 (2013), pp. 458–475. DOI: `10.1002/ett.2650`.

[86] Kieu-Ha Phung, Bart Lemmens, Marnix Goossens, Ann Nowe, Lan Tran, and Kris Steenhaut. "Schedule-based multi-channel communication in wireless sensor networks: A complete design and performance evaluation". In: *Elsevier Ad Hoc Networks* 26 (2015), pp. 88 –102. ISSN: 1570-8705. DOI: `10.1016/j.adhoc.2014.11.008`.

[87] Diego Dujovne, Luigi Alfredo Grieco, Maria Rita Palattella, and Nicola Accettura. *6TiSCH 6top Scheduling Function Zero (SF0)*. Internet-Draft draft-ietf-6tisch-6top-sf0-05. Internet Engineering Task Force, Jan. 2018. URL: `https://tools.ietf.org/html/draft-ietf-6tisch-6top-sf0-05`.

[88] Esteban Municio and Steven Latré. "Decentralized Broadcast-based Scheduling for Dense Multi-hop TSCH Networks". In: *Proceedings of ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. New York City, New York, 2016, pp. 19–24. ISBN: 978-1-4503-4257-5. DOI: `10.1145/2980137.2980143`.

[89] 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 ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. Malta, Malta, 2016, pp. 102–110. DOI: `10.1145/2988287.2989133`.

[90] X. Vilajosana and K. Pister. *Minimal 6TiSCH Configuration*. draft 17. IETF, 2016.

[91] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH". In: *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Seoul, South Korea, 2015, pp. 337–350. DOI: `10.1145/2809695.2809714`.

[92] T. Chang, M. Vucinic, X. Vilajosana, S. Duquennoy, and D. Dujovne. *6TiSCH Minimal Scheduling Function (MSF)*. Internet-Draft. draft-chang-6tisch-msf-03. IETF, 2019.

[93] Nouha Baccour, Anis Koubâa, Habib Youssef, and Mário Alves. "Reliable Link Quality Estimation in Low-power Wireless Networks and Its Impact on Tree-routing". In: *Elsevier Ad Hoc Networks* 27 (2015), pp. 1–25. ISSN: 1570-8705. DOI: `10.1016/j.adhoc.2014.11.011`.

[94] Bogdan Pavkovic, Fabrice Theoleyre, Dominique Barthel, and Andrzej Duda. "Experimental analysis and characterization of a wireless sensor network environment". In: *Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM. 2010, pp. 25–32.

[95] Luca Mottola, Gian Pietro Picco, Matteo Ceriotti, Ştefan Gună, and Amy L Murphy. "Not all wireless sensor networks are created equal: A comparative study on tunnels". In: *ACM Transactions on Sensor Networks (TOSN)* 7.2 (2010), p. 15.

[96] Ana Bildea. "Link quality in wireless sensor networks". PhD thesis. 2013.

[97] Alec Woo, Terence Tong, and David Culler. "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks". In: *International Conference on Embedded Networked Sensor Systems (SenSys)*. ACM. Los Angeles, California, USA, 2003, pp. 14–27. DOI: `10.1145/958491.958494`.

[98] Alexander Becher, Olaf Landsiedel, Georg Kunz, and Klaus Wehrle. "Towards Short-Term Wireless Link Quality Estimation". In: *Workshop on Embedded Networked Sensors (Hot EmNetS)*. ACM. June 2008, pp. 1–5.

[99] O. Iova, F. Theoleyre, and T. Noel. "Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks". In: *Proceedings of IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2013, pp. 2098–2102. DOI: `10.1109/PIMRC.2013.6666490`.

[100] J. Ko and M. Chang. "MoMoRo: Providing Mobility Support for Low-Power Wireless Applications". In: *IEEE Systems Journal* 9.2 (2015), pp. 585–594. ISSN: 1932-8184. DOI: `10.1109/JSYST.2014.2299592`.

[101] A. Bildea, O. Alphand, F. Rousseau, and A. Duda. "Link quality estimation with the Gilbert-Elliot model for wireless sensor networks". In: *Proceedings of IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2015, pp. 2049–2054. DOI: `10.1109/PIMRC.2015.7343635`.

[102] Carlo Vallati, Emilio Ancillotti, Raffaele Bruno, Enzo Mingozzi, and Giuseppe Anastasi. "Interplay of Link Quality Estimation and RPL Performance: An Experimental Study". In: *PE-WASUN*. ACM. Malta, Malta, 2016, pp. 83–90. ISBN: 978-1-4503-4505-7. DOI: `10.1145/2989293.2989299`. URL: `http://doi.acm.org.insis.bib.cnrs.fr/10.1145/2989293.2989299`.

[103] L. Pradittasnee, S. Camtepe, and Y. C. Tian. "Efficient Route Update and Maintenance for Reliable Routing in Large-Scale Sensor Networks". In: *IEEE Transactions on Industrial Informatics* 13.1 (2017), pp. 144–156. ISSN: 1551-3203. DOI: `10.1109/TII.2016.2569523`.

[104] D. Liu, Z. Cao, Y. Zhang, and M. Hou. "Achieving Accurate and Real-Time Link Estimation for Low Power Wireless Sensor Networks". In: *IEEE/ACM Transactions on Networking* 25.4 (2017), pp. 2096–2109. ISSN: 1063-6692. DOI: `10.1109/TNET.2017.2682276`.

[105] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. "Four-Bit Wireless Link Estimation". In: *HotNets-VI*. ACM. 2007.

[106] T. Winter and P. Thubert and A. Brandt and J. Hui and R. Kelsey and P. Levis and K. Pister and R. Struik and JP. Vasseur and R. Alexander. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. rfc 6550. IETF, 2012. DOI: `10.17487/RFC6550`.

[107] Jerry Martocci, Pieter De Mil, Nicolas Riou, and Wouter Vermeylen. *Building automation routing requirements in low-power and lossy networks*. Tech. rep. 2010.

[108] Farid Touati, Rohan Tabish, and Adel Ben Mnaouer. “Towards u-health: An indoor 6LoWPAN based platform for real-time healthcare monitoring.” In: *WMNC*. 2013, pp. 1–4.

[109] Yibo Chen, Jean-Pierre Chanet, and Kun Mean Hou. “RPL Routing Protocol a case study: Precision agriculture”. In: *First China-France Workshop on Future Computing Technology (CF-WoFUCT 2012)*. 2012, 6–p.

[110] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. “Collection tree protocol”. In: *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM. 2009, pp. 1–14.

[111] Jad Nassar, Matthieu Berthomé, Jérémy Dubrulle, Nicolas Gouvy, Nathalie Mitton, and Bruno Quoitin. “Multiple Instances QoS Routing in RPL: Application to Smart Grids”. In: *Sensors* 18.8 (2018), p. 2472.

[112] Joakim Eriksson, Niclas Finne, Nicolas Tsiftes, Simon Duquennoy, and Thiemo Voigt. “Scaling RPL to Dense and Large Networks with Constrained Memory”. In: *International Conference on Embedded Wireless Systems and Networks (EWSN 2018)*. 2018.

[113] Philip Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. *The trickle algorithm*. Tech. rep. 2011.

[114] Karel Heurtefeux and Hamid Menouar. “Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study”. In: *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE. 2013, pp. 1–4.

[115] Simon Duquennoy, Joakim Eriksson, and Thiemo Voigt. *Five-Nines Reliable Downward Routing in RPL*. `http://arxiv.org/abs/1710.02324` 1710.02324v1. arXiv, 2017.

[116] Pascal Thubert. *Objective function zero for the routing protocol for low-power and lossy networks (RPL)*. Tech. rep. 2012.

[117] Omprakash Gnawali and Philip Levis. *The minimum rank with hysteresis objective function*. Tech. rep. 2012.

[118] Mustapha Boushaba, Abdelhakim Hafid, and Michel Gendreau. “Node stability-based routing in Wireless Mesh Networks”. In: *Journal of Network and Computer Applications* 93 (2017), pp. 1 –12. DOI: `10.1016/j.jnca.2017.02.010`.

[119] T. Clausen, U. Herberg, and M. Philipp. “A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)”. In: *WiMob*. Shanghai, China: IEEE, 2011, pp. 365–372. DOI: `10.1109/WiMOB.2011.6085374`.

[120] H. Kermajani and C. Gomez. “On the Network Convergence Process in RPL over IEEE 802.15.4 Multihop Networks: Improvement and Trade-Offs”. In: *Sensors* 14 (2014), pp. 11993–12022. ISSN: 1424-8220. DOI: `10.3390/s140711993`.

[121] H. S. Kim, H. Kim, J. Paek, and S. Bahk. “Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks”. In: *IEEE Transactions on Mobile Computing* 16.4 (2017), pp. 964–979. ISSN: 1536-1233. DOI: 10.1109/TMC.2016.2585107.

[122] H.S. Kim, J. Paek, D. Culler, and S. Bahk. “Do not lose bandwidth: adaptive transmission power and multihop topology control”. In: *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. Ottawa, Canada: IEEE, 2017, pp. 99–108. DOI: 10.1109/DCOSS.2017.23.

[123] S. A. Alvi, F. u. Hassan, and A. N. Mian. “On the energy efficiency and stability of RPL routing protocol”. In: *IWCMC*. IEEE. 2017, pp. 1927–1932. DOI: 10.1109/IWCMC.2017.7986578.

[124] Junyang Shi, Mo Sha, and Zhicheng Yang. “DiGS: distributed graph routing and scheduling for industrial wireless sensor-actuator networks”. In: *IEEE Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2018, pp. 354–364.

[125] Georgios Z Papadopoulos, Tadanori Matsui, Pascal Thubert, Geraldine Texier, Thomas Watteyne, and Nicolas Montavont. “Leapfrog collaboration: Toward determinism and predictability in industrial-IoT applications”. In: *IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6.

[126] Pascale Minet, Ines Khoufi, and Anis Laouiti. “Increasing reliability of a TSCH network for the industry 4.0”. In: *IEEE International Symposium on Network Computing and Applications (NCA)*. IEEE. 2017, pp. 1–10.

[127] Tomas Lagos Jenschke, Remous-Aris Koutsiamanis, Georgios Z Papadopoulos, and Nicolas Montavont. “Multi-path Selection in RPL Based on Replication and Elimination”. In: *International Conference on Ad-Hoc Networks and Wireless*. Springer. 2018, pp. 15–26.

[128] Kui Dang, Ji-Zhong Shen, Li-Da Dong, and Yong-Xiang Xia. “A Graph Route-Based Superframe Scheduling Scheme in WirelessHART Mesh Networks for High Robustness”. In: *Wireless Personal Communications: An International Journal* 71.4 (Aug. 2013), pp. 2431–2444. DOI: 10.1007/s11277-012-0946-2.

[129] S. Zhang, G. Zhang, A. Yan, Z. Xiang, and T. Ma. “A highly reliable link scheduling strategy for WirelessHART networks”. In: *Proceedings of 2013 International Conference on Advanced Technologies for Communications*. 2013, pp. 39–43. DOI: 10.1109/ATC.2013.6698073.

[130] Guangchao Gao, Heng Zhang, and Li Li. “A reliable multipath routing strategy for wirelesshart mesh networks using subgraph routing”. In: *Journal of Computational Information Systems* 5 (2013), pp. 2120–2124. ISSN: 15539105.

[131] J. Lee, W. C. Jeong, and B. C. Choi. “A multi-channel timeslot scheduling algorithm for link recovery in wireless multi-hop sensor networks”. In: *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. 2016, pp. 871–876. DOI: 10.1109/ICTC.2016.7763319.

[132] H. Wu and D. J. Lee. "Robust QoS Scheduling using alternate path for recovery from link failures in IEEE 802.15.4e". In: *International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 2014, pp. 99–100. DOI: `10.1109/ICMU.2014.6799076`.

[133] Venkata P Modekurthy, Abusayeed Saifullah, and Sanjay Madria. "Distributed graph routing for wirelesshart networks". In: *Proceedings of the 19th International Conference on Distributed Computing and Networking*. ACM. 2018, p. 24.

[134] Sanjit Biswas and Robert Morris. "ExOR: opportunistic multi-hop routing for wireless networks". In: *ACM SIGCOMM computer communication review* 35.4 (2005), pp. 133–144.

[135] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. "Low power, low delay: opportunistic routing meets duty cycling". In: *ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2012, pp. 185–196.

[136] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. "Let the tree bloom: Scalable opportunistic routing with orpl". In: *Proceedings of the Conference on Embedded Networked Sensor Systems*. ACM. 2013.

[137] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. "Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments". In: *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*. IEEE Press. 2016, p. 19.

[138] Thong Huynh, Fabrice Theoleyre, and Won-Joo Hwang. "On the interest of opportunistic anycast scheduling for wireless low power lossy networks". In: *Computer Communications* 104 (2017), pp. 55 –66. DOI: `10.1016/j.comcom.2016.06.001`.

[139] Inès Hosni and Fabrice Théoleyre. "Adaptive k-cast Scheduling for High-Reliability and Low-Latency in IEEE802. 15.4-TSCH". In: *International Conference on Ad-Hoc Networks and Wireless*. Springer. 2018, pp. 3–14.

[140] K. Srinivasan, Mayank Jain, Jung Il Choi, Tahir Azim, Edward S Kim, Philip Levis, and Bhaskar Krishnamachari. "The κ Factor: Inferring Protocol Performance Using Inter-link Reception Correlation". In: *MobiCom*. ACM. 2010, pp. 317–28.

[141] Zhengguo Sheng, Shusen Yang, Yifan Yu, Athanasios V Vasilakos, Julie A McCann, and Kin K Leung. "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities". In: *IEEE Wireless Communications* 20.6 (2013), pp. 91–98.

[142] *Deterministic Networking Architecture*. `https://datatracker.ietf.org/doc/draft-ietf-detnet-architecture/`.

[143] Pascal Thubert, Maria Rita Palattella, and Thomas Engel. "6TiSCH centralized scheduling: When SDN meet IoT". In: *Proc. of IEEE Conf. on Standards for Communications & Networking (CSCN'15)*. 2015.

[144] P. Thubert. *An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4*. draft 12. IETF, 2017.

[145] Diego Dujovne, Thomas Watteyne, Xavier Vilajosana, and Pascal Thubert. "6TiSCH: deterministic IP-enabled industrial internet (of things)". In: *IEEE Communications Magazine* 52.12 (2014), pp. 36–41.

[146] Lawrence G Roberts. "ALOHA packet system with and without slots and capture". In: *ACM SIGCOMM Computer Communication Review* 5.2 (1975), pp. 28–42.

[147] 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.

[148] Thang Phan Duy, Thanh Dinh, and Younghan Kim. "Distributed cell selection for scheduling function in 6TiSCH networks". In: *Computer Standards & Interfaces* 53 (2017), pp. 80 –88. ISSN: 0920-5489. DOI: `10.1016/j.csi.2017.03.008`.

[149] Nicolas Privault. "Understanding Markov Chains". In: *Examples and Applications, Publisher Springer-Verlag Singapore* (2013).

[150] Kosmas Kritsis, Georgios Z Papadopoulos, Antoine Gallais, Periklis Chatzimisios, and Fabrice Theoleyre. "A tutorial on performance evaluation and validation methodology for low-power and lossy networks". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 1799–1825.

[151] Elhadi M Shakshuki, Haroon Malik, and Tarek R Sheltami. "A comparative study on simulation vs. real time deployment in wireless sensor networks". In: *Journal of Systems and Software* 84.1 (2011), pp. 45–54.

[152] Kévin Roussel, Ye-Qiong Song, and Olivier Zendra. "Using Cooja for WSN simulations: Some new uses and limits". In: 2016.

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

[154] Ugo Maria Colesanti, Carlo Crociani, and Andrea Vitaletti. "On the Accuracy of Omnet++ in the Wireless Sensor Networks Domain: Simulation vs. Testbed". In: *Proceedings of ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*. Chania, Crete Island, Greece, 2007, pp. 25–31. ISBN: 978-1-59593-808-4. DOI: `10.1145/1298197.1298203`.

[155] Yik-Chung Wu, Qasim Chaudhari, and Erchin Serpedin. "Clock synchronization of wireless sensor networks". In: *IEEE Signal Processing Magazine* 28.1 (2010), pp. 124–138.

[156] Ulf Kulau, Sebastian Schildt, Stephan Rottmann, Björn Gernert, and Lars Wolf. "PotatoNet–Outdoor WSN Testbed for Smart Farming Applications". In: *14. GI/ITG KuVS Fachgespräch Sensornetze* (), p. 11.

[157] 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.

[158] Georgios Z Papadopoulos, Julien Beaudaux, Antoine Gallais, Thomas Noel, and Guillaume Schreiner. “Adding value to WSN simulation using the IoT-LAB experimental platform”. In: *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2013, pp. 485–490.

[159] Julien Vandaele. “Poster: FIT IoT-LAB testbed, to interoperability and beyond”. In: *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing. 2018, pp. 181–182.

[160] Manjunath Doddavenkatappa, Mun Choon Chan, and Akkihebbal L Ananda. “Indriya: A low-cost, 3D wireless sensor network testbed”. In: *International conference on testbeds and research infrastructures*. Springer. 2011, pp. 302–316.

[161] Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. “A survey on (mobile) wireless sensor network experimentation testbeds”. In: *IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE. 2014, pp. 263–268.

[162] Paramasiven Appavoo, Ebram Kamal William, Mun Choon Chan, and Mobashir Mohammad. “Indriya2: A Heterogeneous Wireless Sensor Network (WSN) Testbed”. In: *International Conference on Testbeds and Research Infrastructures*. Springer. 2018, pp. 3–19.

[163] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. “Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems”. In: *Proceedings of the 12th international conference on Information processing in sensor networks*. ACM. 2013, pp. 153–166.

[164] Leonardo S Cardoso, Guillaume Villemaud, Tanguy Risset, and Jean-Marie Gorce. “Cortexlab: A large scale testbed for physical layer in cognitive radio networks”. In: *IC1004 Meeting, Lyon, France*. 2012.

[165] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. “A survey on spectrum management in cognitive radio networks”. In: (2008).

[166] F. Righetti, C. Vallati, G. Anastasi, and S. Das. “Performance Evaluation the 6top Protocol and Analysis of its Interplay with Routing”. In: *Proceedings of IEEE International Conference on Smart Computing (SMARTCOMP)*. 2017, pp. 1–6. DOI: 10.1109/SMARTCOMP.2017.7947029.

[167] Emilio Ancillotti, Carlo Vallati, Raffaele Bruno, and Enzo Mingozzi. “A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management”. In: *Elsevier Computer Communications* 112 (2017), pp. 1 –13. ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/j.comcom.2017.08.005. URL: http://www.sciencedirect.com/science/article/pii/S0140366417305704.

[168] O. Iova, F. Theoleyre, T. Watteyne, and T. Noel. “The Love-Hate Relationship between IEEE 802.15.4 and RPL”. In: *IEEE Communications Magazine* 55.1 (2017), pp. 188–194. ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.1300687RP.

[169] V. Kotsiou, G. Papadopoulos, P. Chatzimisios, and F. Theoleyre. “LABeL: Link-based Adaptive BLacklisting Technique for 6TiSCH Wireless Industrial Networks”. In: *International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM. Miami, FL, US, 2017.

[170] Hyung-Sin Kim, Hosoo Cho, Hongchan Kim, and Saewoong Bahk. “DT-RPL: Diverse Bidirectional Traffic Delivery through RPL Routing Protocol in Low Power and Lossy Networks”. In: *Elsevier Computer Networks* 126 (Oct. 2017), pp. 150–161.

[171] David M. Corey, William P. Dunlap, and Michael J. Burke. “Averaging Correlations: Expected Values and Bias in Combined Pearson rs and Fisher’s z Transformations”. In: *The Journal of General Psychology* 125.3 (1998), pp. 245–261. DOI: 10.1080/00221309809595548. URL: http://dx.doi.org/10.1080/00221309809595548.

[172] J.D. Evans. *Straightforward Statistics for the Behavioral Sciences*. Brooks/Cole Publishing Company, 1996. ISBN: 9780534231002.

[173] D. Dujovne, L.A. Grieco, M.R. Palattella, and N. Accettura. *6TiSCH 6top Scheduling Function Zero / Experimental (SFX)*. Internet-Draft. draft-ietf-6tisch-6top-sfx-00. IETF, 2017.

[174] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. “Monitoring KPIs in synchronized FTDMA multi-hop wireless networks”. In: *Proceedings of Wireless Days (WD)*. 2016, pp. 1–6. DOI: 10.1109/WD.2016.7461516.

[175] P. Agrawal, A. Ahlén, T. Olofsson, and M. Gidlund. “Long Term Channel Characterization for Energy Efficient Transmission in Industrial Environments”. In: *IEEE Transactions on Communications* 62.8 (2014), pp. 3004–3014. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2014.2332876.

[176] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das. “Improving Network Formation in 6TiSCH Networks”. In: *IEEE Transactions on Mobile Computing* 18.1 (2019), pp. 98–110. ISSN: 1536-1233. DOI: 10.1109/TMC.2018.2828835.

[177] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, and K. S. J. Pister. “Simple Distributed Scheduling With Collision Detection in TSCH Networks”. In: *IEEE Sensors Journal* 16.15 (2016), pp. 5848–5849. DOI: 10.1109/JSEN.2016.2572961.

[178] Filip Barać, Mikael Gidlund, and Tingting Zhang. “Ubiquitous, yet deceptive: Hardware-based channel metrics on interfered WSN links”. In: *IEEE Transactions on Vehicular Technology* 64.5 (2015), pp. 1766–1778.

[179] K. Heurtefeux and H. Menouar. “Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study”. In: *WMNC*. Dubai, United Arab Emirates: IFIP, 2013, pp. 1–4. DOI: 10.1109/WMNC.2013.6548990.

[180] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister. “A Realistic Energy Consumption Model for TSCH Networks”. In: *IEEE Sensors Journal* 14.2 (2014), pp. 482–489. ISSN: 1530-437X. DOI: 10.1109/JSEN.2013.2285411.

[181] Diego V. Queiroz, Marcelo S. Alencar, Ruan D. Gomes, Iguatemi E. Fonseca, and Cesar Benavente-Peces. “Survey and systematic mapping of industrial Wireless Sensor Networks”. In: *Journal of Network and Computer Applications* 97 (2017), pp. 96 –125. DOI: 10.1016/j.jnca.2017.08.019.

[182] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. “Bursty Traffic over Bursty Links”. In: *SenSys*. ACM. Berkeley, California, 2009, pp. 71–84. ISBN: 978-1-60558-519-2. DOI: 10.1145/1644038.1644046.

[183] Felix Dobslaw, Tingting Zhang, and Mikael Gidlund. “End-to-End Reliability-aware Scheduling for Wireless Sensor Networks”. In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 758–767.

[184] S. Ben Yaala, F. Théoleyre, and R. 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. DOI: 10.1016/j.adhoc.2017.07.002.

[185] Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, and Fabrice Valois. “Kausa: KPI-aware Scheduling Algorithm for Multi-flow in Multi-hop IoT Networks”. In: *Proceedings of International Conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW)*. 2016, pp. 47–61. DOI: 10.1007/978-3-319-40509-4_4.

[186] G. Daneels, et al. “Accurate Energy Consumption Modeling of IEEE 802.15. 4e TSCH Using Dual-Band OpenMote Hardware”. In: *Sensors* 18.2 (2018), p. 437.

[187] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. d. S. e. Silva. “The Mobile Hub concept: Enabling applications for the Internet of Mobile Things”. In: *Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 2015, pp. 123–128. DOI: 10.1109/PERCOMW.2015.7134005.

[188] L. D. Xu, W. He, and S. Li. “Internet of Things in Industries: A Survey”. In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2300753.

[189] R. Silva, J. S. Silva, and F. Boavida. “Infrastructure-supported mobility in wireless sensor networks - A case Study”. In: *Proceedings of IEEE International Conference on Industrial Technology (ICIT)*. 2015, pp. 1895–1900. DOI: 10.1109/ICIT.2015.7125373.

[190] Pranesh Sthapit, Yeon-Sang Choi, Goo-Rak Kwon, SS Hwang, and JY Pyun. “A fast association scheme over IEEE 802.15. 4 based mobile sensor network”. In: *Proc. ICWMC* (2013).

[191] Y. Al-Nidawi and A. H. Kemp. “Mobility Aware Framework for Timeslotted Channel Hopping IEEE 802.15.4e Sensor Networks”. In: *IEEE Sensors Journal* 15.12 (2015), pp. 7112–7125. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2472276.

[192] B. Dezfouli, M. Radi, and O. Chipara. “Mobility-aware real-time scheduling for low-power wireless networks”. In: *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*. 2016, pp. 1–9. DOI: 10.1109/INFOCOM.2016.7524594.

[193] N. Karowski, A. C. Viana, and A. Wolisz. “Optimized Asynchronous Multichannel Discovery of IEEE 802.15.4-Based Wireless Personal Area Networks”. In: *IEEE Transactions on Mobile Computing* 12.10 (2013), pp. 1972–1985. ISSN: 1536-1233. DOI: 10.1109/TMC.2012.169.

[194] Laura Marie Feeney and Per Gunningberg. “Avoiding an IoT ’Tragedy of the Commons’”. In: *Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM. Munich, Germany, 2018, pp. 495–497. DOI: 10.1145/3210240.3223572.

[195] Thomas Watteyne et al. “OpenWSN: a standards-based low-power wireless development environment”. In: *Transactions on Emerging Telecommunications Technologies* 23.5 (2012), pp. 480–493. DOI: 10.1002/ett.2558.

[196] Elvis Vogli, Giuseppe Ribezzo, L Alfredo Grieco, and Gennaro Boggia. “Fast network joining algorithms in industrial IEEE 802.15.4 deployments”. In: *Ad Hoc Networks* 69 (2018), pp. 65–75.

[197] Romain Jacob and Markus Schuss. *IoTBench - Past, Present, and Future of a Community-Driven Benchmarking Initiative*. Apr. 15, 2019. URL: https://www.research-collection.ethz.ch/handle/20.500.11850/339242. published.

[198] Michael Cheffena. “Industrial wireless communications over the millimeter wave spectrum: opportunities and challenges”. In: *IEEE Communications Magazine* 54.9 (2016), pp. 66–72.

[199] Chiara Pielli, Tanguy Ropitault, and Michele Zorzi. “The Potential of mmWaves in Smart Industry: Manufacturing at 60 GHz”. In: *International Conference on Ad-Hoc Networks and Wireless*. Springer. 2018, pp. 64–76.

[200] Sergio Saponara, Filippo Giannetti, Bruno Neri, and Giuseppe Anastasi. “Exploiting mm-wave communications to boost the performance of industrial wireless networks”. In: *IEEE Transactions on Industrial Informatics* 13.3 (2017), pp. 1460–1470.

List of Abbreviations

6TiSCH IPv6 over the TSCH mode of IEEE 802.15.4e

ASN Absolute Sequence Number

CAP Contention Access Period

CBR Constant Bit Rate

CCA Clear Channel Assessment

CSMA Carrier Sense Multiple Access

DAO-ACK Destination Advertisement Object Acknowledgment

DAO Destination Advertisement Object

DetNet Deterministic Networking

DIO DODAG Information Object

DIS DODAG Information Solicitation

DODAG Destination Oriented Directed Acyclic Graph

DSME Deterministic and Synchronous Multi-channel Extension

DTMC Discrete Time Markov Chain

EB Enhanced Beacon

ECDF Empirical Cumulative Distribution Function

ETSI European Telecommunications Standards Institute

ETX Expected Transmission Count

Free Space Path-Loss FSPL

GMPLS Generalized Multi-Protocol Label Switching

HART Highway Addressable Remote Transducer

IETF Internet Engineering Task Force

IIOT Industrial Internet of Things

IOT Internet of Things

ISM The Industrial, Scientific and Medical Band

J-PDR Joint Packet Delivery Rate

KA Keep-Alive

LLDN Low Latency Deterministic Network

LLN Low-power and Lossy Networks

LoRa Long Range

LQE Link Quality Estimator

LQI Link Quality Indicator

MAC Media Access Control

mmWave Millimeter Wave

MRHOF Minimum Rank with Hysteresis Objective Function

MTU Maximum Transmission Unit

NB-IoT Narrowband Internet of Things

NFC Near-Field Communication

OF Objective Function

OF0 Objective Function 0

OSPF Open Shortest Path First

PCE Path Computation Element

PDR Packet Delivery Rate

PLC Programmable Logic Controllers

PRR Packet Reception Ratio

QoS Quality of Service

RFID Radio Frequency Identification System

RPL IPv6 Routing Protocol for Low-Power and Lossy Networks

RSSI Received Signal Strength Indicator

RSVP Resource Reservation Protocol

SDN Software Defined Network

SF0 Scheduling Function 0

SHM Structural Health Monitoring

TDMA Time Division Multiple Access

TDMA Time Division Multiple Access

TSCH Time Slotted Channel Hopping

WLAN Wireless Local Area Network

WMEWMA Window-Mean Exponentially Weighted Moving Average

WPAN Wireless Personal Area Network

WSN Wireless Sensor Networks

List of Figures

List of Tables

Améliorations des Normes pour la Performance Prévisible dans l'Internet des Objets Industriel dans les Environnements Intérieurs

Introduction

Les réseaux industriels ont été largement déployés pour une multitude d'utilisations. Dans les usines intelligentes, un grand une collection de capteurs et d'actionneurs sont connectés à des automates programmables pour prendre des décisions en temps réel. Ces réseaux industriels sont largement utilisés aussi pour d'autres applications afin de fournir de nouveaux services à haute valeur ajoutée. Par exemple, un ensemble de capteurs et d'actionneurs peut détecter des intrusions ou contrôler le système de chauffage, de ventilation et de climatisation dans bâtiments.

Pour réduire les coûts de déploiement tout en maximisant la flexibilité, les réseaux industriels commencent à devenir sans fil. Cependant, les réseaux radio sont connus pour être très difficiles, car ils présentent des comportements variables dans le temps, en raison d'obstacles et d'interférences externes. Néanmoins, le réseau doit toujours transmettre bien les paquets. Par exemple, une fiabilité élevée (par exemple$>$ 99%) et un délai prévisible sont désormais une priorité, ou une nécessité pour la plupart des applications industrielles.

Heureusement, plusieurs normes sans fil ont été proposées pour améliorer la fiabilité et l'efficacité énergétique des transmissions. Ils s'appuient sur l'ordonnancement strict des transmissions, associé à un saut de fréquence pour améliorer les performances du réseau. Ces protocoles sont souvent déterministes pour fournir des performances stables et faciliter le diagnostic en cas de défaillance.

En règle générale, IEEE 802.15.4-TSCH est projeté sous forme de trame périodique composée du slot de temps et du décalage de canal, ou de cellules. Une paire de nœuds peut négocier un ensemble de cellules dédiées pour son propre usage. Étant donné que cette trame est répétée indéfiniment, ces cellules indiquent une quantité de bande passante réservée pour ce lien spécifique. Le surapprovisionnement aide à faire face aux environnements peu fiables: des cellules supplémentaires sont réservées pour une éventuelle retransmission.

Le saut de fréquence lent permet de lutter contre les interférences externes: en cas d'échec d'une transmission, le paquet est retransmis via un canal radio différent, afin de rendre les pertes de paquets moins répétitives. Toutefois, les réseaux sans fil colocalisés utilisant la même bande ISM (par exemple, les réseaux Wi-Fi) peuvent tout de même nuire à la fiabilité. De plus, les interférences externes présentent des caractéristiques variables dans le temps. Cela implique que le réseau doit gérer efficacement ces variations: il doit fournir suffisamment de ressources pour gérer les rafales de pertes de paquets.

Le groupe de travail 6TiSCH définit actuellement un ensemble de protocoles pour exécuter IPv6 sur les réseaux IEEE 802.15.4-TSCH. 6TiSCH accorde la connectivité IPv6 en fournissant une pile de protocoles de faible puissance qui colle les protocoles de couche supérieure (6LoWPAN, RPL, CoAP) et TSCH via la sous-couche 6P. 6TiSCH s'appuie également sur une fonction de planification pour décider du nombre de cellules à réserver pour chaque lien. Comme les cellules sont allouées à la volée, les conditions variables dans le temps peuvent entraîner des oscillations, ce qui augmente le trafic de contrôle.

Motivation

La bande ISM représente un groupe de fréquences radio sans licence réservées au niveau international pour le fonctionnement d'équipements industriels, scientifiques et médicaux. En raison de sa nature à faible consommation d'énergie, un réseau IEEE802.15.4 est particulièrement désavantagé lorsqu'il est colocalisé avec d'autres réseaux sans fil à plus forte consommation, tels que les réseaux IEEE802.11. La puissance de transmission des dispositifs IEEE 802.11 est de 30 dBm, ce qui est nettement supérieur au 0 dBm utilisé par les dispositifs IEEE 802.15.4. Cette asymétrie a un impact direct sur les performances et la fiabilité des réseaux IEEE802.15.4. Alors que le saut de canal lent utilisé par TSCH tend à accroître la fiabilité, les pertes de paquets continuent de se produire.

Nous assistons à la montée en puissance de nouvelles technologies de réseau sans fil à faible consommation d'énergie pour l'Internet des objets. Par exemple, Long Range (LoRa) et IEEE 802.11ah ont reçu beaucoup d'attention au cours des dernières années pour les déploiements à longue portée. Sigfox et *Narrowband Internet of Things* (NB-IoT) sont d'autres technologies sans fil à faible consommation concurrentes sur le marché de l'IoT. Ainsi, nous nous attendons à ce que dans les années à venir, davantage de technologies de réseau sans fil (et de nouveaux appareils) cohabitent dans une bande de communication déjà encombrée, provoquant des interférences mutuelles.

De plus, les caractéristiques de l'environnement sont variables dans le temps et certaines sources externes d'interférences peuvent s'arrêter ou démarrer de façon aléatoire. Par exemple, plus de personnes en utilisant des appareils sans fil ou un obstacle (un objet, une personne en mouvement) peuvent temporairement affecter les liaisons sans fil. Il a également été signalé que des variations des conditions climatiques affectaient la fiabilité des liaisons sans fil. Ainsi, les métriques de qualité de lien ont un rôle crucial à jouer pour distinguer les variations à court et à long terme.

Une reconfiguration de route est particulièrement coûteuse pour les couches MAC basées sur des réservations comme TSCH: la bande passante de la route précédente doit être désallouée puis allouée dans la nouvelle route. Naturellement, l'allocation de bande passante nécessite la transmission de paquets de contrôle, augmentant la consommation d'énergie du réseau. Pire, lors du réacheminement, la fiabilité du réseau peut être compromise pendant que les nœuds tentent de trouver de nouveaux chemins fiables.

Par conséquent, nous mettons en évidence le défi de recherche suivant dans cette thèse:

Défi Scientifique

Comment assurer une haute fiabilité avec un délai limite supérieur, alors que des facteurs incontrôlés et imprévisibles peuvent entraîner des variations sur les canaux sans fil?

L'objectif principal de cette thèse est d'étudier les performances des protocoles de communication utilisés dans les applications IIoT lorsqu'ils sont déployés dans des environnements intérieurs. En particulier, nous nous concentrons sur l'identification des facteurs qui affectent la fiabilité du réseau lorsque les liaisons sans fil présentent des conditions variant dans le temps. Du plus, nous rendons le réseau plus robuste en proposant des améliorations qui garantissent une grande fiabilité même lorsque les conditions radio changent de manière inattendue. Par conséquent, nous nous appuyons principalement sur des déploiements réels à des fins d'évaluation.

Bien que les simulations fournissent un moyen rapide de tester une hypothèse avant une implémentation longue, elles dépendent fortement de la précision des modèles et tendent à sous-estimer les problèmes qui peuvent survenir dans les scénarios pratiques. Par conséquent, les résultats de simulation peuvent diverger des résultats expérimentaux obtenus à partir de déploiements réels. Ainsi, nous utilisons des dispositifs physiques déployés sur des testbeds réseaux.

Estimation passive de la qualité des liens pour les réseaux 6TiSCH

L'estimation de la qualité des liens est cruciale pour la création de topologies de routage efficaces. En particulier, les liens avec un taux d'erreur élevé ne doivent pas être exploitées car elles sont moins économes en énergie (plus de retransmissions sont nécessaires) et ont un impact négatif sur la fiabilité. Malheureusement, l'estimation de la qualité des liens n'est pas simple. De nombreuses approches reposent sur un sondage actif pour évaluer la qualité de la liaison vers tous les voisins. Bien que la période de sondage puisse être adaptée dynamiquement pour réduire la surcharge, des paquets de contrôle sont toujours nécessaires pour réagir rapidement aux changements.

Nous proposons une approche purement passive pour 6TiSCH, où un nœud identifie les meilleurs parents possibles sans tester individuellement chaque lien. Pour sélectionner le parent préféré le plus précis, des nœuds classent leur voisins en fonction de la quantité de paquets de contrôle reçus de chacun. Ce travail est le premier à faire valoir qu'un classement des voisins est suffisant: un nœud n'a pas besoin d'estimer avec précision la qualité du lien vers tous ses voisins. Nous vérifions expérimentalement que cette métrique peut constituer un bon estimateur même si les paquets de contrôle sont sujets à des collisions.

Un nœud N doit estimer la qualité du lien vers:

- ses voisins actifs, avec lesquels il échange des paquets de données. Typiquement, un voisin actif est un nœud pour lequel il transfère les paquets, ou vers lequel il envoie son propre trafic. Une méthode passive est facile à mettre en œuvre: il suffit de mesurer le taux de livraison des paquets;
- ses voisins inactifs, avec lesquels aucun paquet de donée n'est échangé. Néanmoins, un voisin inactif peut être sélectionné comme parent préféré lorsque les qualités de la liaison radio changent ou si le parent principal tombe en panne.

Le problème devient encore plus délicat à gérer avec des variations temporelles, très courantes pour le type de scénario pris en considération pour cette thèse. Pour les voisins inactifs, la qualité de lien a été évaluée il y a longtemps et ne reflète pas la qualité actuelle. De facto, ces voisins ne seront plus jamais considérés comme le parent préféré, sauf si l'actuel tombe en penne ou que la qualité de son lien devient non fiable.

Nous nous concentrons ici sur un réseau multihop convergecast, car le trafic bidirectionnel n'est toujours pas pris en charge efficacement par 6TiSCH. Notre hypothèse est qu'il existe une corrélation entre le taux de réception des paquets de contrôle d'un voisin et la qualité de la liaison qu'il fournirait en tant que parent préféré.

En effet, la qualité du lien unicast et le taux de diffusion des paquets de contrôle ont une très forte corrélation, et nous avons démontré expérimentalement que nous pouvons classer en toute sécurité les qualités de liaison en utilisant cette métrique. Nous avons intégré ce mécanisme dans la pile 6TiSCH, et nous avons démontré expérimentalement la pertinence de cette méthode passive.

La stabilité des réseaux 6TiSCH dans les environnements intérieurs

Bien que tsch offre de solides garanties de livraison, l'environnement radio présente toujours des caractéristiques variables dans le temps. Ainsi, le réseau doit fournir des ressources suffisantes (bande passante) pour faire face aux variations à court terme: des ressources supplémentaires permettent au réseau de fonctionner dans la pire des situations. Cependant, le surapprovisionnement diminue la capacité du réseau, car davantage de cellules sont réservées aux retransmissions.

Le *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) essaie de gérer les instabilités des canaux en ajustant la topologie de manière dynamique et en s'assurant que chaque nœud a un voisin approprié pour communiquer. Malheureusement, il a été prouvé que RPL réagit de manière excessive pour lier les changements de qualité. Ainsi, le routage dynamique peut être source d'instabilités.

Nous étudions la stabilité des performances d'un réseau 6TiSCH dans les déploiements intérieurs. En particulier, nous effectuons une analyse multicouche identifiant les opérations inefficaces de la pile 6TiSCH qui entravent la topologie de routage et le protocole TSCH de convergence. Nous nous concentrons particulièrement sur les déploiements à long terme, où le réseau devrait atteindre l'état stable avec tous les nœuds synchronisés et capables de communiquer. L'environnement radio est variable dans le temps, en raison par exemple de l'interférences externes et d'obstacles. Cependant, nous soutenons que le réseau devrait continuer à fournir un comportement déterministe.

Dans les déploiements industriels, il n'est pas rare de trouver différentes technologies sans fil cohabitant dans le même environnement. En raison de sa faible consommation d'énergie, un réseau 6TiSCH est particulièrement désavantagé lorsqu'il partage la même bande avec d'autres réseaux de puissance supérieure tels que le Wi-Fi. De plus, des obstacles tels que des murs en béton et des machines à métaux rendent le canal sans fil non stationnaire à long terme.

Une reconfiguration de la topologie n'est pas gratuite et elle est généralement suivie d'une rafale de transmissions de paquets de contrôle. En particulier, 6TiSCH est basé sur la réservation et les cellules doivent être négociées tout au long du chemin vers le nœud-puits. Pire encore, le nombre de paquets perdus tend à augmenter, générant des dépassements de tampon. Par conséquent, nous avons besoin de protocoles efficaces qui garantissent des reconfigurations uniquement lorsque cela est vraiment nécessaire.

Dans cette contribuition, nous avons montré que le réseau ne parvient pas à converger vers un état stable sur deux testbeds différent, même dans des situations statiques. Pour les deux testbeds, nous pouvons identifier les moments d'instabilité des performances dus aux oscillations des conditions radio provoquées par des interférences externes et des obstacles. Nous avons identifié les causes des instabilités et proposé une solution pour chacune des couches de la pile 6TiSCH.

Tout d'abord, nous avons démontré qu'un réarrangement des cellules partagées dans le slotframe réduit la probabilité de collisions pour les paquets de contrôle, ouvrant la voie à une négociation plus rapide lors des reconfigurations de la topologie. Ensuite, nous avons simplifié la gestion du l'ordonnancement des transmissions entre deux nœuds pour réduire l'instabilité du réseau causée par la renégociation de toutes les cellules à partir de zéro, lorsqu'elles détectent une incohérence d'ordonnancement. Nous avons également exploité la corrélation existante entre le taux de réception des paquets de contrôle et la qualité de liens unicast pour créer une sélection parent en deux étapes, en évitant les mauvais choix conduisant à des instabilités. Nous obtenons enfin un réseau qui converge plus rapidement et qui réagit avec précision lors des moments d'instabilités.

La pertinence d'anycast pour les réseaux 6TiSCH

L'anycast est une technique pour améliorer la fiabilité lors de l'utilisation de liens vulnérable. Cette technique a été proposée pour autoriser plusieurs récepteurs à être actifs en même temps, de sorte qu'une transmission est considérée comme erronée quand aucun des récepteurs n'a pu la décoder et l'acquitter. Puisqu'une transmission n'est perdue que si tous les récepteurs ne reçoivent pas un paquet, la fiabilité du réseau et l'efficacité énergétique peuvent être considérablement améliorées.

Alors que les modèles de propagation radio fournissent des caractéristiques très stables, les évaluations expérimentales prouvent que la réalité est beaucoup plus complexe. Un récepteur peut réussir à décoder une séquence de paquets, puis arrêter de recevoir pendant longtemps. En d'autres termes, les pertes de paquets ne sont pas indépendantes, en raison, par exemple, d'interférences externes. L'identification de liens stables à long terme peut aider à éviter les oscillations, mais réduirait également la diversité de routage.

Le 6TiSCH a été initialement conçu pour les transmissions unicast: dans le programme, une cellule est affectée à une paire de récepteur / émetteur. L'utilisation d'un chemin unique n'est pas tolérante aux pannes. Un seul nœud défectueux suffit pour rompre la route et aucun paquet n'est finalement reçu par le nœud-puits. Certains peuvent affirmer que le protocole de routage est en charge de trouver une autre route après avoir détecté le défaut. Cependant, une telle reconfiguration est particulièrement coûteuse dans les réseaux synchronisés: la bande passante doit être réallouée sur le nouveau chemin. De multiples chemins anycast devraient rendre les routes beaucoup plus robustes et tolérantes aux pannes, tout en évitant des coûts de reconfiguration. Anycast réduirait le nombre de transmissions en affectant plusieurs récepteurs à un seul émetteur. Nous visons à fournir le même niveau de fiabilité avec moins de retransmissions.

Dans des conditions réels, un paquet peut être perdu en raison, par exemple, d'interférences externes, qui affectent tous les récepteurs. Lorsque les pertes de paquets présentent une forte corrélation, la probabilité conditionnelle n'est pas égale au produit des probabilités de succès: les événements ne sont plus indépendants. En règle générale, si le paquet vers le premier parent a été perdu, il est hautement probable qu'il a également été perdu pour le parent secondaire.

Ainsi, nous avons proposé dans cette contribution la métrique *Joint Packet Delivery Ratio* (J-PDR) pour prendre en compte les transmissions anycast avec toute corrélation de pertes de paquets. J-PDR prend en compte le fait que plusieurs récepteurs écoutent chaque transmission. Ainsi, nous devons calculer le taux de livraison multi-voisins. Nous utilisons le concept de *transmission sucess sequences*. Chaque récepteur stocke indépendamment le succès de transmission pour les k derniers paquets. Cette séquence de bits est rapportée régulièrement à l'émetteur afin qu'il puisse calculer le J-PDR. Plus précisément, chaque récepteur rapporte sa séquence de réception à l'émetteur, c'est-à-dire un bitmap, avec un bit par transmission de paquet. Étant donné que l'émetteur utilise un numéro de séquence incrémenté à chaque transmission, un récepteur sait qu'un paquet n'a pas été reçu en identifiant les vides dans les numéros dans les séquences.

Dans cette contribution, nous avons quantifié ici l'intérêt de anycast pour améliorer la fiabilité du 6TiSCH dans un environnement intérieur réaliste. En particulier, nous avons utilisé des données expérimentales pour étudier en profondeur la corrélation entre les pertes de paquets. Nous avons proposé une heuristique pour sélectionner un ensemble approprié de parents indépendants, capables d'améliorer la fiabilité. De cette façon, nous avons réduit à la fois le nombre de retransmissions et le retard, lorsque la source du flux est à plusieurs sauts de la destination. Nos résultats expérimentaux ont mis en évidence que chaque nœud de notre environnement intérieur avait au moins deux parents avec des pertes de paquets indépendantes.

Délai d'attachement des nœuds mobiles

L'Internet des objets mobiles est en train d'émerger où les objets intelligents peuvent se déplacer indépendamment. Bien que la mobilité joue un rôle de plus en plus important pour de nombreux déploiements industriels, la norme 6TiSCH ne propose pas une approche claire pour gérer un taux élevé de changements de topologie en raison de l'association / dissociation des appareils mobiles. Ainsi, le défi consiste à gérer un ensemble d'appareils mobiles au sein d'une infrastructure de réseau sans fil statique. De plus, le mécanisme de saut de canal introduit une nouvelle couche de complexité: un nœud doit attendre de recevoir le paquet de synchronisation sur son canal d'écoute actif, retardant son association au réseau. Une association rapide est un facteur clé pour permettre la mobilité sur des réseaux sans fil de faible puissance.

Nous nous concentrons ici sur une topologie de réseau où le nœud-puits et une collection de nœuds de relais sont statiques. Seuls quelques appareils (par exemple des robots) sont mobiles et représentent les feuilles de l'infrastructure réseau. Ainsi, un nœud mobile envoie son paquet à un nœud de relais voisin, qui les transfère à travers un chemin de relais vers le nœud-puits. Les nœud mobiles constituent les feuilles et doivent identifier un seul nœud relais voisin pour envoyer leurs paquets. Ils doivent capturer de paquet de synchronisation, ajuster leur horloge et savoir où se trouvent les prochaines cellules partagées, pour pouvoir transmettre leurs premiers messages.

Après avoir sélectionné un voisin relais, un nœud mobile engage une négociation pour réserver des cellules dédiées à ses transmissions. Ainsi, la mobilité présente les défis suivants:

Liens avec perte: comme chaque appareil utilise des transmissions sans fil, le manque de fiabilité est la norme. Même les réseaux colocalisés exploitant la même bande ISM peuvent interférer et donc avoir un impact important sur ses performances. Les appareils mobiles ont tendance à rejoindre / quitter le réseau ou à changer fréquemment de points de connexion. Dans un réseau 6TiSCH, cette instabilité de la topologie génère un grand nombre de paquets de contrôle pendant que le réseau converge à nouveau. Pendant la convergence, un réseau a tendance à présenter une latence plus élevée et une fiabilité plus faible;

Portée de transmission courte pour la réutilisation des fréquences et pour augmenter la capacité du réseau, un nœud utilise une puissance d'émission limitée. Cependant, cette courte portée tend à augmenter le nombre de transferts: un appareil mobile doit trouver un nouveau point d'attache lorsque la qualité du lien vers son parent précédent commence à se détériorer;

Délai de découverte: au départ, un nœud mobile n'est pas au courant de la séquence de sauts de fréquence utilisée par le réseau et doit écouter passivement les paquets de synchronisation. Malheureusement, le multicanal augmente très considérablement le temps de connexion, même pour les appareils statiques De plus, de nœud mobile doit maintenir sa radio allumée, ce qui gaspille de l'énergie;

Délai de négociation: seules les cellules dédiées sont résistantes aux collisions. Malheureusement, la négociation des cellules avec un voisin coûte cher et prend beaucoup de temps. En particulier, plusieurs poignées de main sont nécessaires pour vérifier qu'une cellule est disponible à la fois pour l'émetteur et le récepteur.

Nous avons défini une chaîne de Markov à temps discret pour représenter le processus de jonction d'un nouveau nœud (mobile), lorsqu'il rejoint le réseau pour la première fois. Notre modèle prend en compte à la fois les temps de synchronisation et de négociation dans les réseaux 6TiSCH. Nos simulations démontrent la précision de notre modèle pour estimer finement le temps de synchronisation et de négociation. De toute évidence, les réseaux denses signifient un plus grand nombre de collisions, ce qui a un impact très négatif sur le temps de synchronisation. Pire encore, la négociation de cellules dédiées est également très coûteuse, car le taux de collision des paquets de contrôle est très élevé. Nous utilisons également notre modèle DTMC pour évaluer le gain d'utilisation de plusieurs canaux. En répartissant les paquet synchronisation de sur différents canaux, le taux de collision est considérablement réduit.

Conclusion

L'objectif de ce travail était d'améliorer la fiabilité des réseaux sans fil à faible puissance dans les scénarios intérieurs, où les obstacles et les interférences externes sont la règle. En particulier, nous avons proposé des améliorations pour les normes largement utilisées dans l'Internet des Objets Industriel afin d'augmenter la robustesse du réseau aux variations brusques des conditions radio. Étant donné que 6TiSCH est devenu la principale norme pour l'Internet des Objets Industriel, nous avons concentré nos efforts de recherche sur celuici. Cependant, dans une certaine mesure, nos solutions peuvent être adaptées à d'autres normes.

Dans cette thèse, nous avons passé la plupart de nos efforts de recherche en réalisant de réelles expériences sur des testbeds. Bien que les testbeds permettent aux chercheurs de tester leurs solutions dans des conditions réalistes, ils posent certains problèmes liés à la validation des résultats. Le principal défi est de savoir comment rendre nos expériences reproductibles afin que d'autres chercheurs puissent les reproduire. Habituellement, dans les déploiements sur tesbed, nous n'avons aucun contrôle sur l'environnement, ce qui fait nos résultats présentent un certain niveau de variabilité. En effet, des facteurs incontrôlés et imprévisibles (des personnes, des interférences) fonctionnent comme des variables aléatoires affectant différemment nos expériences.

En général, nous avons fait de notre mieux pour fournir le maximum d'informations à des fins de réplication et de validation. Bien que la description détaillée de l'expérience, la mise à disposition du code source et de l'ensemble de données améliorent la reproductibilité, nous devons encore définir une méthode standard pour effectuer des expériences face à des environnements non contrôlés.