Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Kommunikationsnetze

Radio Resource Management of Industrial Wireless Sensor Networks for Networked Control Systems

Samuele Zoppi, M.Sc.

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:		Prof. Dr.-Ing. Sandra Hirche
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Wolfgang Kellerer
	2.	Prof. Dr.-Ing. James Gross

Die Dissertation wurde am 11.01.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 07.05.2021 angenommen.

Radio Resource Management of Industrial Wireless Sensor Networks for Networked Control Systems

Samuele Zoppi, M.Sc.

07.05.2021

Abstract

Recent technological advancements in low-power Wireless Sensor Networks (WSN) opened the door to their application in industrial operations. In this new paradigm, Industrial Wireless Sensor Networks (IWSN) must fulfil, despite processing and energy limitations, the challenging communication requirements of feedback-based industrial processes such as Networked Control Systems (NCS). NCS are control systems that operate in a communication network. In this scenario, a key challenge for IWSN is to provide communication Quality of Service (QoS) in the presence of random packet loss introduced by interference and the harsh wireless industrial environment. This doctoral thesis delves into this problem following a bottom-up approach.

First, in order to determine analytical bounds on the achievable physical-layer performance of IWSN transmissions, the communication Packet Error Rate (PER) is evaluated in propagation environments with different attenuation and multipath fading characteristics. Furthermore, the impact of external interference on the PER is evaluated thanks to measurement- and software-based estimators. Results demonstrate their suitability in estimating PER in static and dynamic interference scenarios.

Secondly, noting that existing QoS provisioning methods do not exploit interfered channels, two link-layer Radio Resource Management (RRM) schemes are presented in order to provide QoS in the presence of interference. Each scheme is based on a stochastic model of the IWSN packet transmissions, relies on the estimation of PER, and is designed to provide bounded latencies and reliability to data packets. The whitening hopping sequence design is developed in order to provision reliability over consecutive re-transmissions in the presence of static interference. The reliability-based dynamic scheduler is derived in order to provide link-layer QoS in the presence of dynamic interference.

Then, network-layer scheduling policies are investigated to support the transmission of NCS over a two-hop network path representing the communication of a control loop. Given the delay requirements of the NCS, a transient model of the network is developed to characterize the Delay Violation Probability (DVP) of individual packets traversing it. Furthermore, scheduling policies are developed to optimize the network resources in order to minimize

the DVP of individual control packets. Static and dynamic scheduling policies exploit, respectively, initial and continuous information of the queue states. Results show that static policies allow a DVP reduction of one order of magnitude and dynamic policies of several orders of magnitude.

Lastly, this thesis characterizes the relationship between the communication QoS and the Quality of Control (QoC) of NCS. This is achieved via the derivation of a delay-reliability model that defines the End-to-End (E2E) delay distribution of packets in a two-hop IWSN and relates it to QoC. Experimental measurements of an IWSN testbed and a simulated Networked Inverted Pendulum (NIP) validate the model and prove that, given the communication parameters of the IWSN, the achievable QoC is fully determined. Additionally, the relationship between communication QoS and QoC is investigated via measurements of an experimental NCS. This is achieved via the implementation of an open-source NCS platform, the definition of an NCS architecture, and the delay-analysis of a control loop. Measurement of the NCS platform provide a direct interconnection between QoC and the delays arising from control, computation, and communication, which can be used to develop novel RRM schemes and NCS.

Kurzfassung

Die technologischen Fortschritte bei Wireless Sensor Networks (WSN) ermöglichen ihre Anwendung im industriellen Betrieb. Industrial Wireless Sensor Networks (IWSN) müssen in diesem neuen Paradigma trotz Verarbeitungs- und Energiebeschränkungen die anspruchsvollen Kommunikationsanforderungen von rückgekoppelten industriellen Prozessen wie Networked Control Systems (NCS) erfüllen. NCS sind Regelungssysteme, die mit Hilfe eines Kommunikationsnetzes arbeiten. In diesem Szenario besteht die zentrale Herausforderung der IWSN in der Bereitstellung von Quality of Service (QoS) bei zufälligen Paketverlusten, die durch die drahtlose Industrieumgebung und Interferenzen verursacht werden.

Die vorliegende Dissertation geht dieser Problematik mit einem Bottom-up-Prinzip nach. Zuerst wird die Packet Error Rate (PER) in Umgebungen mit verschiedenen Dämpfungs- und Schwundcharakteristiken ausgewertet, um die analytischen Grenzen für die erreichbare Leistung der Bitübertragungsschicht von IWSN-Übertragungen zu bestimmen. Darüber hinaus wird der Einfluss von Interferenzen auf die PER mittels messtechnischer und softwarebasierter Schätzer bewertet. Die Ergebnisse bestätigen die Eignung der verwendeten Schätzer für die Schätzung der PER in statischen und dynamischen Interferenzszenarien.

Mit der Kenntnis, dass bestehende QoS-Bereitstellungsmethoden gestörte Kanäle nicht berücksichtigen, werden zwei Sicherungsschicht Radio Resource Management (RRM) Strategien vorgestellt, um QoS in Anwesenheit von Interferenzen bereitzustellen. Jedes Schema basiert auf einem stochastischen Modell der IWSN-Paketübertragungen, beruht auf der Schätzung von PER und wird entwickelt, um begrenzte Latenzen und Zuverlässigkeiten der Datenpakete zur Verfügung zu stellen. Insbesondere wird die Whitening Hopping Sequenz vorgestellt, die bei aufeinanderfolgenden Übertragungen in Anwesenheit statischer Interferenzen Zuverlässigkeit bietet. Außerdem wird ein zuverlässigkeitsbasierter dynamischer Scheduler entwickelt, der Kommunikationspläne an dynamische Interferenzen anpasst.

Daraufhin werden Vermittlungsschicht Scheduling-Strategien untersucht, um die NCS-Übertragung in einem Zwei-Hop-IWSN zu unterstützen, welches die Kommunikation in einer Regelschleife darstellt. Angesichts der Verzögerungsanforderungen des NCS wird ein transientes Modell des Netzwerks entwickelt, um die Verzögerungswahrscheinlichkeit, Delay

Violation Probability (DVP), der einzelnen Kontrollpakete, die das Netzwerk durchlaufen, zu charakterisieren. Außerdem werden Scheduling-Strategien entwickelt, um die Netzwerkressourcen so zu optimieren, dass die DVP der einzelnen Kontrollpakete minimiert werden. Die anfänglichen und laufenden Informationen über den Zustand der Warteschlangen werden von statischen und dynamischen Scheduling-Strategien genutzt. Die Ergebnisse zeigen, dass statische Richtlinien eine Reduktion um eine Größenordnung und dynamische Strategien um mehrere Größenordnungen erlauben.

Schließlich charakterisiert die Dissertation die Beziehung zwischen der QoS und der Quality of Control (QoC) von NCS. Dies wird durch die Herleitung eines Verzögerungs-Zuverlässigkeits-Modells erreicht, das die End-to-End (E2E)-Verzögerung von Paketen in einem Zwei-Hop-IWSN definiert und mit der QoC in Verbindung bringt. Experimentelle Ergebnisse eines IWSN-Prüfstands und eines simulierten Networked Inverted Pendulum (NIP) validieren das Modell und beweisen, dass bei gegebenen Kommunikationsparametern des IWSN die erreichbare QoC vollständig bestimmt wird. Außerdem wird der Zusammenhang zwischen QoS und QoC zusätzlich durch Messungen an einem experimentellen NCS untersucht. Durch die Implementierung einer Open-Source NCS-Plattform, sowie die Definition einer Architektur für NCS und die Verzögerungsanalyse einer Regelschleife wurde eine vollständige Charakterisierung der NCS-Plattform durchgeführt. Die Messergebnisse der zuvor genannten NCS-Plattform zeigen einen direkten Zusammenhang zwischen QoC und den Verzögerungen, die durch Steuerung, Berechnung und Kommunikation entstehen. Dies kann zur Entwicklung von RRM-Strategien und NCS verwendet werden.

Acknowledgment

Throughout the doctorate and while writing this dissertation, I have received considerable support from colleagues, family, and friends, which deserve this acknowledgement. Thanks to all of you, I could find the energy and motivation to succeed in this project.

I first want to thank Prof. Kellerer for constantly supporting me throughout all the challenges of my doctorate. His guidance and trust have considerably helped me in exploring and pursuing exciting research projects, which are included in this dissertation and not only. I want to devote a major acknowledgement to Prof. Gross and Dr. Champati for hosting me during my research visit at KTH and for their wise and continuous support over the last years of my doctorate.

I want to thank all my former colleagues in LKN for the engaging and exciting discussions. In particular, I want to thank Dr.-Ing. Gürsu and Dr.-Ing. Vilgelm for inspiring my research directions and guiding me through my first research project. I am also thankful to Dr.-Ing. Guck and Dr.-Ing. Van Bemten for the creative discussions and to Dr.-Ing. Klügel for mentoring my further studies and experiments in wireless communications. I would also like to thank Mr. Ayan, Dr.-Ing. Molinari and Dr.-Ing. Gallenmüller for their close collaboration on our cross-disciplinary research project. A big "Thank You" goes to Ms. Giunta, Mr. Krapf and Dr.-Ing. Maier for their friendliness and continuous support with students, lectures, and research projects.

I want to thank my family for believing in my goals and constantly supporting me during all my studies. In particular, I would like to dedicate this dissertation to my grandparents. Their wisdom and life-achievements have greatly inspired me. I am extremely grateful to my girlfriend Žana for her daily support during the last and most challenging years of my doctorate. Your positive and supportive energy gave me the strength to complete this endeavour.

Finally, I want to thank all my friends. Your regular and constructive sharing of opinions and points of view has strongly helped me during the doctorate. The several happy moments spent together have helped me to rest and regenerate outside my research.

Contents

Chapter 1

Introduction

Wireless Sensor Networks (WSN) consist of multiple low-power embedded devices equipped with sensors communicating via wireless links. Since their introduction, WSN have been employed in a plethora of applications. Traditional WSN are used for sensing and monitoring of natural environments, such as mountains and forests, and physical constructions, such as buildings and bridges. For this reason, WSN are deployed in large-scale geographical areas and transmit information over multiple hops to a central entity, the sink, which collects and processes the measurements. Sensors measure natural phenomena, such as outdoor temperature and humidity, and generate small and sporadic traffic in the network. Infrequent transmissions, combined with several energy- and processing-constrained devices, stimulated the research community to develop energy-efficient mechanisms for WSN [11, 69]. In fact, energy efficiency is a fundamental requirement for WSN, which allows them to operate independently for several years.

Long lifetime, flexible deployments, and scalability attracted the application of WSN to industrial processes in so-called Industrial Wireless Sensor Networks (IWSN). In IWSN, sensors and actuators regularly communicate with a remote control logic and are deployed to monitor and steer physical systems. IWSN can be applied, for instance, for real-time monitoring of a production plant or involved in the production processes. In all these applications, packets must be reliably delivered to fulfil the specific timings of industrial tasks. For instance, a sudden event may cause sensors to generate time-critical packets to be delivered to the controller or actuators in order to react in time. For this reason, IWSN represents a paradigm shift for WSN, where Quality of Service (QoS) performances such as low latency and reliability have more priority than energy efficiency [76, 109, 160]. To achieve these requirements, novel IWSN communication standards have been released, such as WirelessHART [124], ISA 100.11a [15], and WIA-PA [158]. In all standards, the Network Coordinator (NC) centrally coordinates the network resources in order to meet the application requirements. In this way,

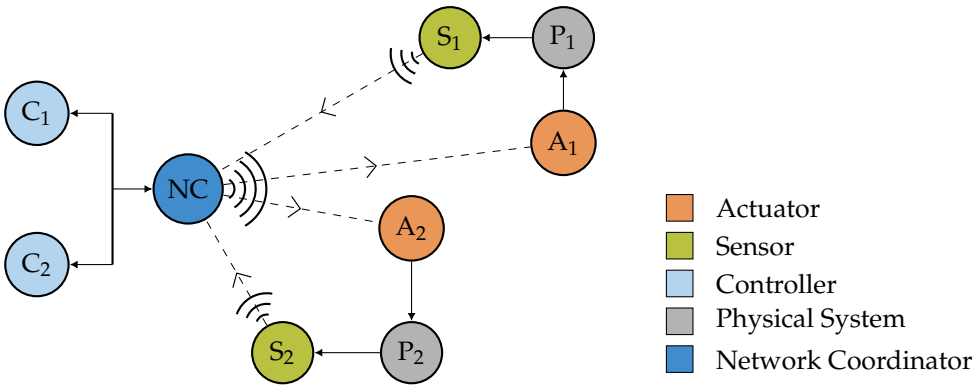

Figure 1.1: IWSN network architecture of feedback-based communication of industrial applications.

radio resources of the entire network can be efficiently coordinated at both data link and network layers.

This work investigates the operation of feedback-based industrial applications in IWSN with a specific focus on Networked Control Systems (NCS), which generate a significant portion of industrial traffic and represent the current research trend of industrial systems [157]. In NCS, the control loop of an automatic control system is established between distributed sensors, controllers, and actuators, and allows to remotely steer a physical system. NCS are well integrated in the industrial environment and can be employed to control the movement of production machines and robotic arms, or to remotely steer robots in the factory, such as automatic guided vehicles transporting materials or drones for inspections [101]. As represented in Fig. 1.1, each NCS operates according to a control loop that is periodically executed each sampling period. Measurements of the physical system are transmitted by the sensor (S) to the remote control logic (C), which computes and transmits commands to the actuator (A) in order to steer it. In order to correctly operate, both messages from sensor and controller must be delivered within the end of the sampling period. Losing a packet or exceeding this deadline severely degrades the performance of the NCS, denoted in this work as Quality of Control (QoC), and can cause its instability. In the context of industrial applications, low QoC can cause the interruption of production activities, damage to physical machines, or, in the worst case, harm human operators [47, 64, 119].

Deploying NCS over a wireless network introduces several advantages, such ease of installation, maintenance, and flexibility [101] and enables mobility and operation in hostile environments. NCS messages generate low data-rate traffic, i.e., few kbps are needed to transmit the messages of sensor and controller, however, they require stringent *QoS performance*. In fact, to achieve a desired QoC, NCS pose stringent *delay* and *reliability* requirements to the communication network. In IWSN, low latency operation, which for industrial automation

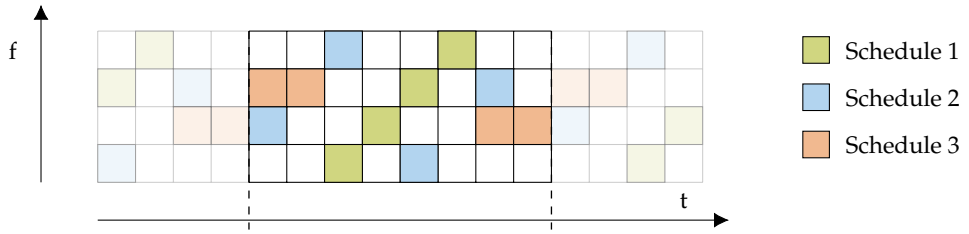

Figure 1.2: Link-layer radio resource model of the TSCH medium access.

systems is in the range of a few hundreds of ms [160], is achieved thanks to a star network topology. In this way, sensors and actuators directly communicate to the NC, which is connected to the control logic (cf. Fig. 1.1). Furthermore, the Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) scheme of IWSN standard enables reliable and timely communication. In TSCH, as depicted in Fig. 1.2, radio resources are organized in a time-frequency grid – time is divided in time slots grouped in Time Division Multiple Access (TDMA) frames and channel hopping is performed over the available frequencies. This combination enables the deterministic allocation of transmission opportunities and enhanced reliability via frequency diversity. Furthermore, it allows the NC to allocate *schedules*, i.e., sets of radio resources from the grid, to the devices. Schedules are computed according to the latency and reliability requirements of industrial applications and are regularly distributed by the NC via broadcast messages.

Although IWSN embody all the capabilities to support industrial applications, achieving the QoS required by NCS is challenging due to the presence of *packet loss* introduced by unreliable wireless links. The industrial environment is particularly harsh for wireless communication. It is characterized by metallic objects, such as machines and structures, that introduce a high attenuation and a rich scattering [28, 71], which lower the signal quality. Furthermore, differently than other wireless communication technologies, IWSN operate in the shared Industrial Scientific and Medical (ISM) unlicensed band and are strongly affected by interference from coexisting wireless technologies, such as Wireless Local-Area Network (WLAN) and Bluetooth, and electromagnetic emissions of industrial machines. These effects introduce stochastic variations in the received signal, which degrade the communication quality and cause packets to be lost. This problem is further intensified by the presence of mobile nodes in the factory that introduce time-varying shadowing and interference effects on the received signal. Therefore, it is crucial to develop effective Radio Resource Management (RRM) techniques to ensure the correct operation of feedback-based systems. This task, however, requires overcoming several engineering challenges in the design of RRM schemes for interfered IWSN and in the performance analysis of NCS subject to packet loss and delays.

This doctoral thesis tackles these problems following a vertical approach, thereby presenting own solutions at the physical, link, networking, and application layers of IWSN.

1.1 Research Challenges

Several challenges arise in the design of RRM mechanisms for IWSN in order to support a desired level of QoC in the presence of random packet loss. The NC, which is in charge of computing the schedules of all links, must not only take into account the link-layer QoS performances of a single transmitter, but also its implications with respect to the End-to-End (E2E) network-layer performance of both sensor and controller links. To achieve this, the NC must combine the QoC requirements of NCS with information about the instantaneous quality of the wireless channels in a proper manner. This thesis tackles the complexity of this problem via a bottom-up approach, it investigates the QoS performance of the physical, link, and network layers and their interaction with the application layer. At each layer, different challenges with respect to the current state-of-the-art are tackled to enable NCS communication in IWSN, which motivated the contributions presented in this work. In the following sections, individual research problems at the physical, link, networking, and application layers are described together with the solutions introduced by the own contributions. To avoid ambiguities between QoS provisioning methods at the different layers, QoS metrics are defined at each layer.

1.1.1 C1: Estimation of PER in Interfered IWSN

The first research challenge addresses the nature of the packet loss in the industrial environment. In fact, in order to understand the QoS performance of each IWSN transmission, the causes of packet loss must be closely investigated. As QoS performance metric at the physical-layer, the Layer 1 (L1) reliability r^{L1} of a frequency channel in a given environment is considered, which is described by its average Packet Error Rate (PER).

Although the propagation characteristics of wireless industrial environments have been largely investigated in terms of attenuation and scattering [28, 71], few works focused on the impact of industrial interference. A common approach is to perform measurements of IWSN transmissions in different scenarios [38, 132, 139, 150] and analyse the PER achieved by the devices on each frequency channel. Most of the existing measurement campaigns, however, did not focus on the evaluation of external interference. Measurements of PER in an IWSN subject to external interference were performed in collaboration with Mr. Gürsu and Mr. Vilgelm for an aircraft scenario [57]. While existing works measured the PER of a particular environment, they did not provide insights in the nature of PER in an industrial

scenario (C1a). For this reason, the following questions remain open. What is the Signal-to-Interference-and-Noise-Ratio (SINR) achieved by the channel? What is the impact of the propagation environment? And of interference?

In a factory, time-varying effects, such as mobility of devices and machines, may affect the communication PER. Thus, it is of paramount importance to quantify and estimate the time-varying PER of IWSN during the operation of the network. To this end, several link quality estimators have been proposed in the literature [16]. Among them, however, only few methods estimated the link PER, which can be clustered in hardware- and software-based methods. On the one hand, hardware-based estimators [107, 161] rely on measurements of Received Signal Strength Indicator (RSSI) to estimate the amount of received power and compute the communication SINR. On the other hand, software-based estimators [25, 146] solely rely on the observation of received packets to estimate PER. A major limitation of existing hardware- and software-based estimators is that only interference-free scenarios have been investigated (C1b). Therefore, the impact of time-varying external interference on PER estimation has not been studied.

With respect to the above challenges, the following **contributions** are presented:

- S1a: Measurements of RSSI are performed for both signal and interference in different scenarios. Distinct measurements of received power enable the separate analysis of propagation characteristics and interference on PER values.
- S1b: A measurement-based estimator is evaluated in the presence of external interference.
- S1c: The performances of software-based PER estimators in a time-varying interference scenario are investigated via simulations. Simulations enable the accurate comparison of the estimators' performances and the optimal selection of their parameters.

Thanks to these contributions, a detailed investigation of the effect of interference on the PER of IWSN transmissions is performed. Computing an accurate estimation of physical-layer PER is a fundamental requirement of the link- and network-layer RRM schemes presented in this work. In fact, all the methods rely on a stochastic model of the IWSN transmissions that relates to PER.

1.1.2 C2: Interference-Aware Link-Layer Scheduling of IWSN

Packet loss interrupts the operation of the control loop and severely degrades the performances of feedback-based applications. Therefore, given the physical-layer PER of a link, it is

important to develop RRM mechanisms that achieve desired link-layer QoS performances. In this work, the link-layer QoS performance is defined by the Layer-2 *reliability* r^{L2}, i.e., the percentage of received messages, and *delay* d^{L2}. The NC of the IWSN is in charge of allocating schedules in order to fulfil the QoS requirements. This is achieved by allocating time slots and frequency channels in order to select the transmission PER and the number of transmission attempts of data packets.

Several works tackled the problem of link-layer QoS provisioning in IWSN. Channel hopping is a widespread solution to reduce the effect of unpredictable, external interference between consecutive transmissions [52, 57, 139, 142]. Given a specific interference scenario, however, IWSN protocols do not specify how to construct the hopping sequence. The state-of-the-art technique of optimizing channel hopping is blacklisting – excluding highly interfered channels from the hopping sequence [40, 50, 67, 75, 159]. Although blacklisting is able to improve the communication reliability, it comes with the price of sacrificing radio resources (C2a). Considering the scarcity of radio resources and the stringent QoS requirements of feedback-based applications, it is necessary to develop RRM schemes that make use off all the resources efficiently.

While modifying the hopping sequence can effectively improve reliability, each modification requires the reconfiguration of the IWSN. Therefore, hopping sequence RRM techniques are employed when the PER of the channels does not considerably vary over time. Alternatively, time and frequency resources can be individually scheduled to the devices. The definition of appropriate scheduling methods, however, is not covered by the standards. To this end, several autonomous [42, 72, 112], distributed [9, 58, 60, 90, 98], and centralized [66, 97, 103, 131] scheduling schemes for the TSCH medium access were developed in the literature. Existing methods computed conflict-free schedules in large multi-hop networks in order to support the traffic of several flows. Their focus was to reduce internal interference and the issue of time-varying external interference was not taken into account (C2b). Few methods tackled the problem of compensating packet loss [10, 34, 66], however, slots are allocated without taking into account the physical-layer performance of the channels (C2c). Furthermore, in all these methods, link-layer delay and reliability requirements are not taken into account while constructing schedules (C2d). Although several scheduling methods were developed for IWSN, the problem of dynamic scheduling based on the time-varying PER of the channels to fulfil link-layer QoS performances was not investigated.

For this reason, this work presents methods to solve the above-mentioned challenges providing the following **contributions**:

- S2a: Based on the physical-layer communication PER r^{L1}, a time-varying stochastic model of each resource block of the TSCH scheduling grid is provided.

- S2b: Under statically interfered channels, an algorithm is derived to compute a hopping sequence that exploits all the available channels in order to fairly distribute reliability r^{L2} to the devices within a given deadline d^{L2}. The hopping sequence is constructed allocating re-transmissions over different channels according to their PER.
- S2c: A dynamic reliability-based scheduler is developed to fulfil QoS requirements d^{L2}, r^{L2} in the presence of time-varying interference. At each TDMA frame, the scheduler estimates the up-to-date PER of the channels employing a software-based estimator and computes new schedules accordingly.

Thanks to these contributions, given the physical-layer PER performance r^{L1}, the desired link-layer reliability and delay performances r^{L2}, d^{L2} can be achieved by communication links. Furthermore, by measuring and estimating the static and time-varying PER of the channels, the presented RRM schemes make use of all the frequency channels in the presence of interference. These are fundamental achievements in order to enable the communication of NCS in IWSN. In fact, relying on the link-layer QoS performance, multi-hop RRM schemes are presented in this work in order to support the transmission of individual messages in a feedback loop.

1.1.3 C3: Packet-Based Network-Layer Scheduling of IWSN

In order to support the communication of NCS, IWSN must reliably convey the messages of each feedback control loop. As represented in Fig 1.1, a feedback loop can be modelled as a two-hop network path including a sensor-to-controller and controller-to-actuator links. When steering a physical system, the delay between the generation of sensor measurements and the reception of actuation commands is vital for the application. In fact, actuators must react in time to sudden events captured by the sensor. This requires that messages are processed and delivered to the actuator within a given *deadline*. Failure in delivering the messages on time can significantly impact the performance of the application, i.e., the stability/safety of the controlled physical system. This is particularly challenging in IWSN, where, due to severe fading and interference, packets are randomly dropped and must be re-transmitted. In this scenario, the performance of feedback-based applications is characterized by the latency distribution of each packet traversing the network. Given a maximum latency defined by the application, the probability that the latency of network packets exceeds it is captured by the Delay Violation Probability (DVP). In this thesis, the network-layer delay d^{L3} and DVP r^{L3} of feedback messages are used to describe the network QoS performance.

Given a specific application deadline, in order to ensure the correct operation of the feedback loop, the available resources must be optimized to minimize the DVP of each packet

traversing the network. Characterizing DVP for a finite sequence of packets is challenging, as it depends on fluctuating system parameters – such as backlogs in the queues at the time of arrival and temporal variability of the service offered by wireless links – and calls for the analysis of the transient behaviour of the network [27]. Resource allocation in the transient domain can significantly benefit from the availability of network state information [133, 134], i.e., the evolution of the network queues until the deadline. Therefore, this work investigates network-layer resource allocation methods that exploit network state information in order to minimize the DVP of feedback messages.

Several related works tackled the problem of resource allocation in a multi-hop wireless network to support time-critical applications. Most of them, however, focused on the steady-state performance of the application flows [37, 46, 62, 122] (C3a). Furthermore, existing works that considered per-packet delay and reliability requirements [22, 30, 51] allocated a maximum number of re-transmissions (C3b) and did not optimize the network resources to minimize the DVP of time-critical packets (C3c). Maximizing the per-packet reliability in a multi-hop network was attempted by Soldati et al. [123], however, assuming independent resource allocations for consequent transmitters (C3d). This is different from a DVP-based approach that considers the correlation of transmission outcomes over consecutive transmitters.

The above-mentioned research challenges are tackled in this work via the following **contributions**:

- S3a: Two upper bounds to compute the DVP of individual packets traversing a two-hop network based on the initial network conditions are derived.
- S3b: Novel heuristic scheduling policies that compute a static resource allocation based on the initial network state are presented.
- S3c: Noting that DVP cannot be directly used for dynamic resource allocation, a dynamic heuristic scheduling policy that maximizes the network's throughput is derived.
- S3d: The performance of the derived dynamic scheduling policy is compared to the classical Backpressure (BP) [134], Max Weight (MW) [94] and Weighted-Fair Queuing (WFQ) [99] scheduling policies.
- S3e: Via simulations, the impact of network state information on the network-layer QoS performance of the derived schemes is evaluated.

By deriving DVP and developing scheduling policies to minimize it, the performances of a lossy two-hop network that transmits the packets of a feedback loop is fully characterized and optimized. In this way, it is possible to accurately describe the impact on the network

performances of each packet generated by the application. For this reason, the last research challenge of this work closely analyses the relationship between DVP and the performances of NCS.

1.1.4 C4: Impact of Network QoS on the NCS Performance

When transmitting the packets of sensor and controller, IWSN directly expose NCS to network delays and packet loss. In fact, although suitable RRM schemes can achieve desired link- and network-layer QoS performances, with a given probability, individual packets are lost and the control loop is temporarily interrupted. Failure in delivering actuation commands to NCS introduces a performance penalty and can cause instability [101]. The application-layer performances of NCS are described in this work by the term QoC. Therefore, in order to evaluate the QoC achieved by NCS in IWSN, the following questions must be answered. What is the relationship between QoC and the network-layer QoS? What is the QoC achieved by the link- and network-layer QoS provisioning schemes of IWSN?

Although NCS have been extensively investigated over the last decades [56, 101, 152, 156, 157], a full characterization of NCS and IWSN is missing for several reasons. First, most of control-related works [101, 108, 157] adopted high-level models of the communication network that do not capture the effect of physical, link, and networking layers (C4a). Second, existing works that jointly considered control and communication systems [36, 100, 102, 104, 113, 114] focused on the design of NCS and did not provide insights in the delay and reliability achieved by the network communication (C4b). Further, network-related approaches mainly investigated WLAN communication [18, 136–138, 145], thus a contention-based medium access (C4c). Additionally, network-related works developed methods for QoS provisioning at the link and networking layers that model NCS as real-time traffic [37, 62, 79, 115, 147, 151], thereby not characterizing the impact of QoS on QoC (C4d). Finally, the vast majority of existing methods investigated the impact of wireless communication on NCS via simulations, thus neglecting the effects of a real implementation. Existing practical research works that investigated experimental NCS [17, 19, 26, 81] only focused on specific aspects of the NCS and did not provide a characterization of QoC for the entire NCS in terms of delay and reliability (C4e).

This work closely investigates the relationship between IWSN communication and NCS performances, and presents solutions to the above challenges via the following **contributions**:

- S4a: The QoC of a simulated NCS is evaluated via the Loop Success Probability (LSP), the probability that NCS messages are correctly delivered at each control loop, which

abstracts the impact of network imperfections. LSP models the network-layer QoS and corresponds to $1-$DVP when the control loop defines the deadline of messages.

- S4b: An analytical delay-reliability model of the E2E delay distribution of packets in a two-hop network path is derived and related to LSP.
- S4c: Via measurements of a two-hop IWSN and a simulated NCS, the delay-reliability model is validated and its relationship with LSP and QoC confirmed.
- S4d: A complete analysis of experimental NCS was performed thanks to a model of the NCS architecture and the delay-analysis of a control loop.
- S4e: An open-source NCS benchmarking platform was implemented for experimental performance measurements. Measurement results provide a direct interconnection between QoC and the delays arising from control, computation, and communication.

By characterizing the QoC of NCS in real wireless networks, the above contributions conclude the layer-based investigation of IWSN and NCS tackled by this thesis. The model and measurements describe the relationship between the network-layer QoS and QoC in several scenarios, and can aid the design of RRM and NCS in lossy wireless networks.

1.2 Thesis Outline

This thesis investigates the problem of RRM of IWSN for NCS tackling the research challenges described in Sec. 1.1 following a bottom-up approach. It studies the QoS performance of the physical, link, and network layers and their interaction with the application layer. As depicted in Fig. 1.3, each chapter tackles the research challenges of a specific layer and provides a comprehensive description of the contributions provided by the own works.

Chapter 2 tackles the research challenge C1 investigating the detailed physical-layer reliability of IWSN. In Sec. 2.1 the physical-layer characteristics of IWSN transmissions are described. In Sec. 2.2 an analytical model of the wireless propagation environment is presented together with the analysis of its PER performances. In particular, Sec. 2.3 presents the contributions of the chapter describing the results of measurement-based estimation (S1a, S1b) in Sec. 2.3.1 and of software-based estimation (S1c) in Sec. 2.3.2.

Tackling challenge C2, Chapter 3 presents link-layer RRM schemes for the provisioning of link-layer QoS to IWSN link. A stochastic-model of the link-layer radio resources of IWSN (S2a) is presented in Sec. 3.2.2. Sec. 3.3 describes the reliable design of the hopping sequence in the presence of static interference (S2b), while Sec. 3.4 a reliability-based dynamic scheduler for time-varying interference (S2c).

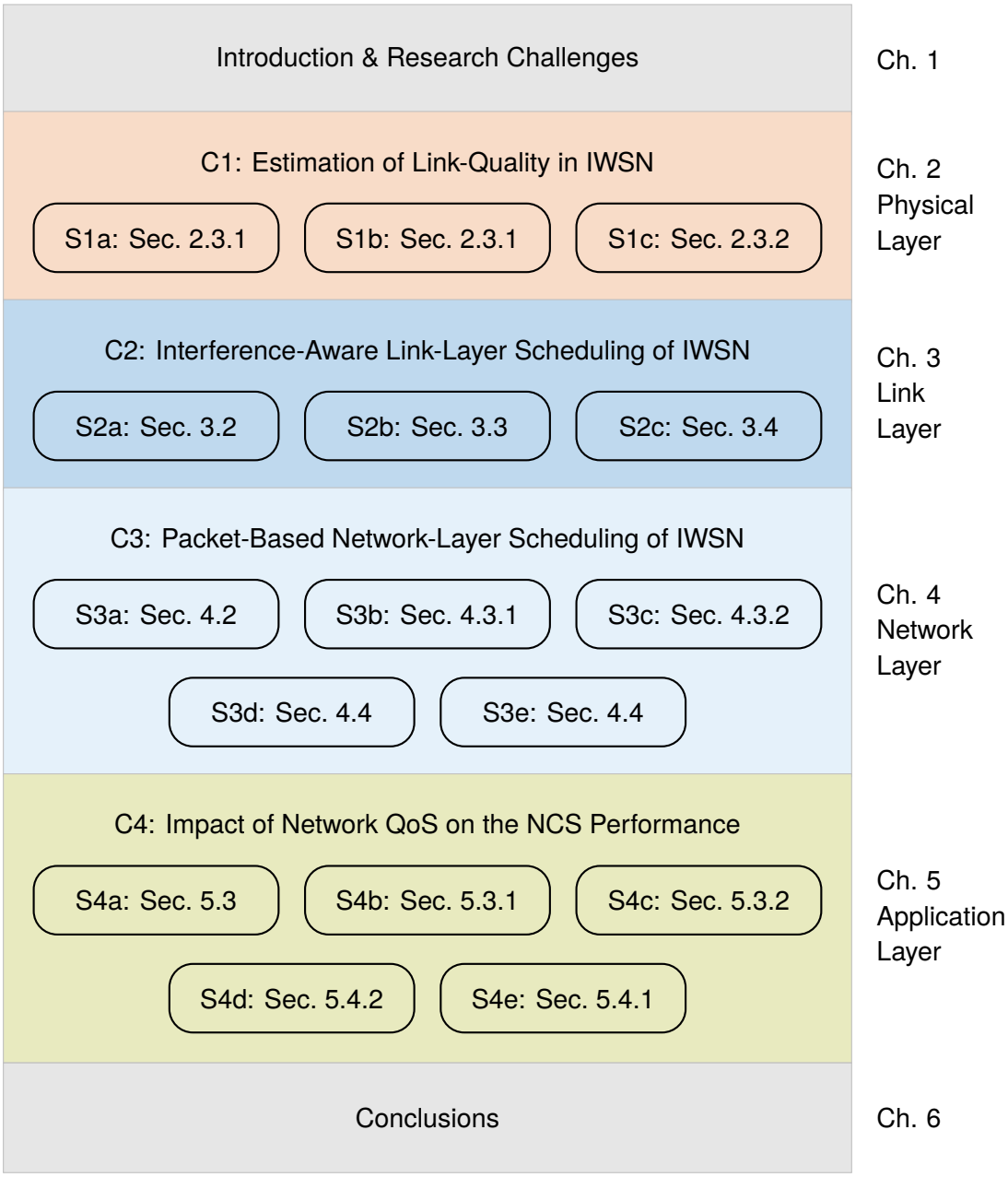

Figure 1.3: Outline of the thesis. For each chapter, contributions are listed together with the section that describes them.

Chapter 4 tackles challenge C3 via the definition of network-layer scheduling policies that minimize the DVP of individual packets in feedback-based applications. Sec. 4.2 describes the network model and the analytical derivation of DVP (S3a). Static (S3b) and dynamic (S3c) scheduling policies are presented, respectively, in Sec. 4.3.1 and Sec. 4.3.2 and their performance is comprehensively evaluated (S3d, S3e) in Sec. 4.4.

Chapter 5 presents several solutions to challenge C4, which tackles the performance characterization of NCS operating in wireless networks affected by delays and packet loss. A

model of the NCS under study and the definition of QoC is provided in Sec. 5.2. Sec. 5.3 evaluates QoC in terms of LSP (S4a). It presents in Sec. 5.3.1 the delay-reliability model that relates LSP to the communication parameters of IWSN (S4b) and validates it (S4c) in Sec. 5.3.2. Sec. 5.4 presents the evaluation of network-layer QoS and QoC of an experimental NCS. In particular, Sec. 5.4.2 describes the details of the open-source NCS platform (S4d), while Sec. 5.4.1 presents the model of its architecture and the delay-analysis of the control loop (S4e).

Chapter 6 concludes the thesis summarizing the main findings and discussing possible directions for future work.

Chapter 2

Industrial Wireless Sensor Networks: A Physical Layer Perspective

This chapter investigates the reliability of individual Industrial Wireless Sensor Networks (IWSN) transmissions in the industrial environment. This is a fundamental building block of Radio Resource Management (RRM) schemes for Quality of Service (QoS) provisioning, which is achieved by characterizing the physical-layer performance of IWSN. The average communication Packet Error Rate (PER) is chosen as a metric to describe the reliability of individual IWSN transmissions. The PER indicates the expected performance achieved by a single device and reflects the characteristics of its propagation environment. Thus, it is a suitable metric to describe the physical-layer performance of the devices in the network.

Differently than existing works that characterize PER in large, outdoor Wireless Sensor Networks (WSN) deployments [28, 71], this chapter describes the PER of IWSN in an indoor industrial environment. The industrial scenario is substantially different from traditional WSN deployments for several reasons. In fact, it is characterized by higher attenuation and stronger multipath fading introduced by metallic objects in the factory, and is affected by interference from coexisting wireless technologies and machines. Furthermore, the mobility of devices and objects in the factory introduces time-varying effects in the communication PER, which have not been analyzed in existing works.

This chapter provides a detailed characterization of the PER achieved in an industrial scenario via model- and measurement-based approaches. A description of the IWSN physical-layer characteristics is given in Sec. 2.1. Sec. 2.2 presents an analytical PER model of IWSN transmissions and evaluates it for different propagation parameters of the industrial environment and in the presence of interference. Sec. 2.3 describes different methods to estimate the PER in real IWSN deployments. In particular, Sec. 2.3.1 presents PER estimation from measurements of received power that have been performed in the Master's Theses of

Mr. Dauti [35] and Mr. Rammurthy [111] under my supervision. Furthermore, Sec. 2.3.2 presents software-based PER estimators that rely on the observation of received packets in the presence of time-varying interference. Results in Sec. 2.3.2 provide a performance overview of the software-based estimators that are used by RRM schemes for QoS provisioning in Ch. 3. Finally, Sec. 2.4 concludes the chapter summarizing the main results and discussing them in the context of this work.

2.1 Physical Layer

WSN are designed to be deployed in an ad-hoc fashion in order to serve a wide spectrum of applications. While several wireless communication technologies are available for WSN, such as magnetic, optical, and acoustic, Radio Frequency (RF) remains the most commonly adopted thanks to its ease of deployment and pervasiveness. The other technologies are generally deployed when RF communication is impractical, i.e., when RF interference is too high. As described in [12, p. 54], WSN RF communication can be performed using three possible frequency and bandwidth allocations in the Industrial Scientific and Medical (ISM) unlicensed bands:

- Narrow-band: 1 channel at 433 MHz, 1 channel at 868 MHz, and 10 channels at 915 MHz with a data rate of 20-50 kbps.
- Spread-spectrum: 16 channels at 2.4 GHz with a data rate of 250 kbps.
- Ultra-wide-band: 3.1-10.6 GHz with a data rate of 48-1024 Mbps.

While traditional WSN are deployed over large distances (several kilometers) and transport infrequent messages, IWSN must cover smaller areas (10-100 m) and require higher data rates. For these reasons, WSN are usually deployed in the narrow-band, while IWSN operate in the 2.4 GHz band. Higher frequencies additionally allow for frequency reuse in the same area, improving coverage in dense industrial environments. Although ultra-wide-band provides attractive data-rates, its short communication range (<10 m) limits the range of applications and is not suitable for mobility.

The industrial propagation environment is particularly harsh for wireless communication. It is characterized by metallic objects, e.g., machines and structures, that introduce, at 2.4 GHz, a high attenuation and a rich scattering [28, 71]. Moreover, it incorporates interference from the transmission of coexisting wireless technologies, such as Wireless Local-Area Network (WLAN) and Bluetooth, and electromagnetic emissions of industrial devices and machines. To reduce the effect of multipath and interference, spread spectrum techniques such as Direct

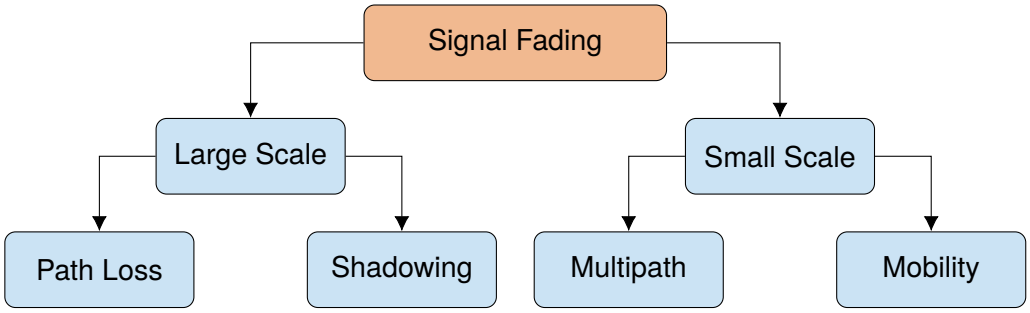

Figure 2.1: Wireless channel effects on the transmitted signal.

Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) are used. Between them, DSSS is preferred as it achieves better performances in presence of multipath [12]. DSSS spreads the narrow-band signal over a large bandwidth modulating the bits with a pseudo-random sequence called chip. The resulting transmitted signal lies below the noise floor and is less sensitive to frequency-selective interference and multipath.

To comply with these communication requirements, the Institute of Electrical and Electronics Engineers (IEEE) Standard (Std.) 802.15.4 [32] defines for the 2.4 GHz band DSSS processing and Offset Quadrature Phase Shift Keying (OQPSK) modulation. The resulting transmitted signal has 2 MHz bandwidth and achieves peak data rates of 250 kbps. The standard defines a minimum receiver sensitivity of -85 dBm.

2.2 Channel Model and Communication PER

The performance of IWSN communication is strongly characterized by the propagation environment and interference from other technologies. In order to evaluate it, an IWSN channel model and its analytical PER characterization are presented in this section. The wireless channel alters each transmission introducing fading, i.e., stochastic variations in the received power. As presented in [49, p. 27] and shown in Fig. 2.1, large- and small-scale fading are possible, which are caused by effects that are visible over large and small changes in the transmitter-receiver distance.

On the one hand, *large-scale* fading is introduced by signal attenuation during the propagation in the wireless medium and through obstacles, causing the effects of path loss and shadowing. These effects are commonly modelled using the simplified path-loss model in Eq. (2.1). Let $\{\ell, m\}$ be the communication link identified by the ℓ-th transmitter and the m-th receiver of an IWSN transmission, the path-loss component is

$$\zeta^{\ell,m}\ \text{dB} = 20\log_{10}\frac{c_0}{4\pi f d_0} - 10\eta\log_{10}\frac{d^{\ell,m}}{d_0}, \tag{2.1}$$

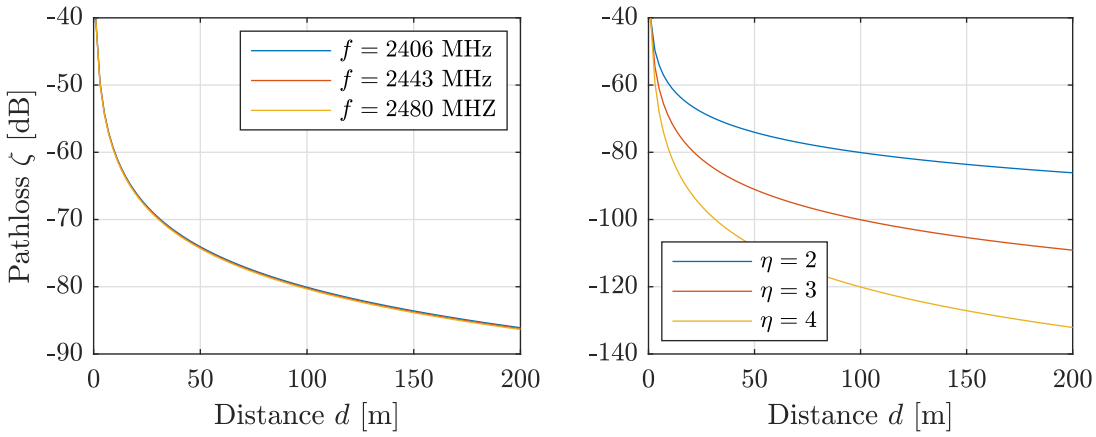

Figure 2.2: Evaluation of the simplified path loss model for typical IWSN propagation values.

where c_0 is the speed of light, f the carrier frequency, η the path-loss decay, and $d^{\ell,m}$ the relative transmitter-receiver distance normalized by a reference distance d_0. The path-loss decay significantly varies according to the propagation environment and additionally captures the effect of shadowing. Its values range between 2, which represents free-space propagation, and 4, which models the obstructed communication caused by obstacles [28]. Typical values for IWSN are $f \in [2405, 2480]$ MHz, $\eta \in [2, 3.8]$, and $d_0 = 1$ m [28, 85]. The impact of different propagation parameters on the path loss is shown in Fig. 2.2 for varying distances. The figure shows that different frequencies of the ISM band have a negligible effect, while the path-loss decay η considerably affects the path-loss components, thus the coverage of the network.

On the other hand, *small-scale* fading arises from multipath and the Doppler effect caused by the relative movement between transmitter and receiver. In IWSN, the movement of devices is generally slow and characterized by the local dynamics of industrial machines, which are limited to a few m^2, or the motion of Automatic Guided Vehicles (AGV), which can reach a maximum speed of few km/h. For this reason, the impact of mobility on IWSN communication is secondary compared to multipath. Multipath represents the propagation of multiple copies of the transmitted signal over different physical paths caused by reflections and scattering in the environment. Each copy is a fraction of the original signal and reaches the receiver with a path-dependent delay. This causes the superposition of multiple delayed copies on the received signal, which is called inter-symbol interference. This effect is particularly strong in indoor industrial environments, where metallic structures and objects generate a highly reflective environment [71]. For short-range indoor IWSN communication, the value of multipath at transmission at time k is modelled in dB by ξ_k, a normal distribution with zero mean and variance σ^2_{dB} [155]. Typical values for σ^2_{dB} in IWSN are between 2 and 4 dB [28].

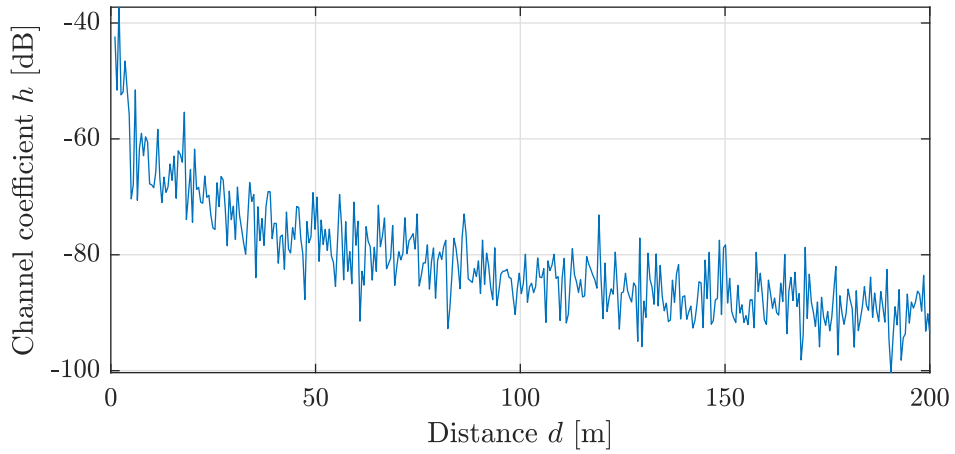

Figure 2.3: Exemplary realization of the channel coefficient h as a function of distance for $f = 2405$ MHz, $\eta = 2.2$, and $\sigma^2_{\text{dB}} = 6.2$ dB.

For a transmitted signal at time k with transmission power t_k^ℓ on link $\{\ell, m\}$, the value of received power $r_k^{\ell,m}$ is calculated using the channel coefficient $h_k^{\ell,m}$, which combines both large- and small-scale fading effects

$$r_k^{\ell,m}\ \text{dB} = t_k^\ell\ \text{dB} + h_k^{\ell,m}\ \text{dB}, \tag{2.2}$$

$$h_k^{\ell,m}\ \text{dB} = \zeta^{\ell,m}\ \text{dB} - \xi_k\ \text{dB}. \tag{2.3}$$

Fig. 2.3 shows the effect of multipath on the channel coefficient for $f = 2405$ MHz, $\eta = 2.2$, and $\sigma^2_{\text{dB}} = 6.2$ dB over different distances. Although path loss is the main cause of signal attenuation, multipath introduces large oscillations in the received powers. In some cases this can lead to a stronger power, in others it severely attenuates the received signal.

From a given received signal, the communication quality can be calculated via the Signal-to-Interference-and-Noise-Ratio (SINR), which incorporates the propagation characteristics of the wireless medium and the impact of interference and noise on the desired signal

$$\gamma_k^{\ell,m} = \frac{r_k^{\ell,m}}{r_k^{\text{IF}} + r^{\text{N}}} = \frac{t_k^\ell\, h_k^{\ell,m}}{\sum_{n\neq\ell} t_k^n\, h_k^{n,m} + r^{\text{N}}}. \tag{2.4}$$

The SINR is the ratio between the power of the received signal $r_k^{\ell,m}$ and the power of interference r_k^{IF} plus noise r^{N}. In Eq. (2.4) t_k^n is the transmission power of the n-th transmitter and r^{N} is the power of the additive white Gaussian noise. The noise power can be calculated using the Boltzmann's equation and the signal bandwidth B as follows

$$N_0 = k_b \tau, \tag{2.5}$$

$$r^{\text{N}} = N_0 B, \tag{2.6}$$

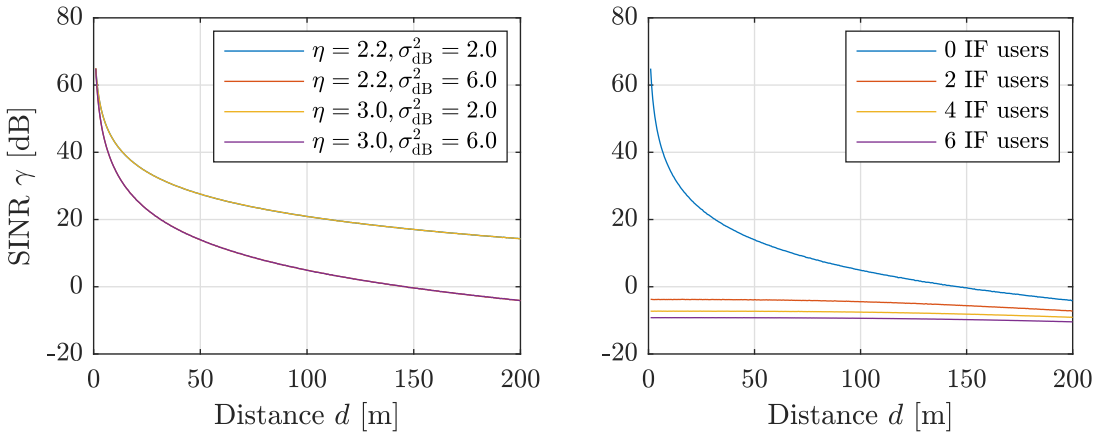

Figure 2.4: Average SINR values for increasing communication distances with $f = 2405$ MHz and transmission power equal to -20 dBm in different interference-free propagation environments (left) and multiple interferers with $\eta = 3$ and $\sigma^2_{\text{dB}} = 6$ (right).

where k_b is the Boltzmann's constant and τ is the environment temperature. Alternatively, it can be determined experimentally by averaging measurements of received power in the absence of data transmission and interference. Fig. 2.4 visualizes the average SINR for different propagation scenarios (left) and number of interferers (right) with $f = 2405$ MHz, transmission power equal to -20 dBm and increasing communication distances. In the left figure, the impact of low and high values of attenuation and multipath is shown for an interference-free scenario and $r^{\text{N}} = -105$ dBm, value obtained from experimental measurements [35, 111]. As the path-loss exponent η increases, the average SINR severely degrades. On the contrary, the effect of multipath on the average SINR is negligible as it only affects individual transmissions. On the right, the impact of different number of interferers is shown for $\eta = 3$ and $\sigma^2_{\text{dB}} = 6$ dB. When operating in presence of interference, the SINR is severely reduced and the impact of distance is negligible. In this condition, the network is operating in the interference-limited scenario. On the contrary, when no interference is present, the network operates in the noise-limited scenario.

From the SINR value it is possible to calculate different communication quality metrics. In fact, each SINR corresponds to unique Bit Error Rate (BER) and PER values. As shown in [12, p. 64], the BER for the OQPSK modulation with DSSS is given by

$$b_k^{\ell,m} = Q\left(\frac{2N_c E_k^{\ell,m}}{N_c + 4E_k^{\ell,m}(K-1)/3}\right), \tag{2.7}$$

$$E_k^{\ell,m} = \frac{B}{R}\gamma_k^{\ell,m}. \tag{2.8}$$

In Eq. (2.7), $Q(\cdot)$ denotes the standard Gaussian error function, $N_c = 8$ is the number of chips per bit, $K = 1$ is the number of simultaneous transmitting users, and $R = 250$ kbps

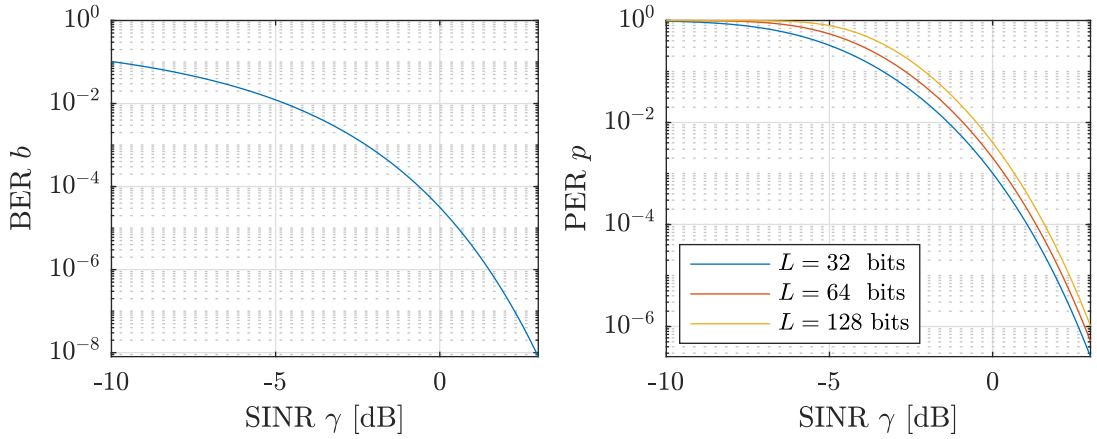

Figure 2.5: BER (left) and PER (right) of different SINR values for varying packet size L.

is the transmission data rate. As IWSN only use the Cyclic Redundancy Check (CRC) error detection mechanism that is not able to recover erroneous bits, a packet is correctly received only if all the bits are correctly received. Thus, for a packet of L bits, the PER value is given by

$$p_k^{\ell,m} = 1 - \left(1 - b_k^{\ell,m}\right)^L. \tag{2.9}$$

Fig. 2.5 visualizes Eq. (2.7) (left) and Eq. (2.9) (right) for different values of SINR and packet size. As the SINR increases, both BER and PER values increase. Different packet sizes have a small impact on the PER, however, transmitting small or large packets can impact the PER up to half order of magnitude. Thanks to the equations presented in this section, it is possible to quantify the reliability of an individual IWSN transmission given its propagation and communication parameters. As presented in Ch. 3 and 4, characterizing the communication PER is essential to model the behavior of the wireless links and derive RRM schemes to provide QoS.

2.3 PER Estimation in Real Deployments

The analytical channel presented in the previous section can be used to quantify the communication quality in IWSN, however, it shows limitations when it is required to reflect the characteristics of a real propagation environment. While this is a natural limitation of analytical models, it is particularly harmful in IWSN, where the communication reliability is affected by unpredictable effects, such as the movement of machines and interference, and by device-specific hardware imperfections [155]. Therefore, it is a common practice for experimental IWSN deployments to characterize the communication quality of a particular environment via measurement campaigns [38, 57, 132, 139, 150].

Differently than the existing works that evaluate the communication PER in interference-free scenario, in this section, the impact of interference on PER is evaluated in details. In Sec. 2.3.1, the PER is estimated via measurements of received power in an IWSN subject to interference. In contrast with the own preliminary work that considers an IWSN deployed in an aircraft interference scenario [57], measurements of received power are exploited to differentiate the impact of propagation environment and external interference on the resulting PER. Furthermore, in Sec. 2.3.2 the accuracy of different software-based PER estimators, which estimate PER observing received packets, is characterized in a time-varying interference scenario.

2.3.1 Measurement-Based Estimation

To investigate the reliability of IWSN communication in an experimental environment, measurements of a real IWSN device, the Zolertia RE-Mote™ platform, were performed. The RE-Mote™ is based on the Texas Instruments CC2538 ARM Cortex-M3 system on chip, which runs at a frequency of 32 MHz and is equipped with 32 KB of RAM and 512 KB of programmable flash. Its RF and processing components are able to provide very low energy consumption, maximum 20 mA for the transceiver operation and 0.6 mA for the processor.

Similarly to most of WSN RF chips, the CC2538 provides measurements of received power at the transceiver via the Received Signal Strength Indicator (RSSI). The RSSI is calculated averaging the received power over 8 symbol periods for a total of 128 µs [32]. An analytical model of the RSSI at time k for link $\{\ell, m\}$ is given by [35, 161]

$$\mathrm{RSSI}_k^{\ell,m}\,\mathrm{dBm} = r_k^{\ell,m}\,\mathrm{dBm} + r_k^{\mathrm{IF}}\,\mathrm{dBm} + r^{\mathrm{N}}\,\mathrm{dBm}. \tag{2.10}$$

Eq. (2.10) shows that three factors constitute the RSSI, the power of the received signal $r_k^{\ell,m}$, interference $r_k^{\ell,m}$, and noise $r^{\ell,m}$.

In order to estimate the communication PER from RSSI measurements, it is first necessary to relate RSSI to SINR. An SINR estimation is obtained by measuring RSSI values in the presence and absence of signal transmission. In fact, via Noise Plus Interference (NPI) measurements, i.e., RSSI measurements in the absence of signal transmission, it is possible to quantify the amount of noise and interference in a given environment. Once RSSI and NPI values are measured, it is possible to estimate the instantaneous SINR as follows [35, 107, 111]

$$\gamma_k^{\ell,m} = \frac{10^{\frac{\mathrm{RSSI}_k^{\ell,m}}{10}} - 10^{\frac{\mathrm{NPI}_k^{\ell,m}}{10}}}{10^{\frac{\mathrm{NPI}_k^{\ell,m}}{10}}}. \tag{2.11}$$

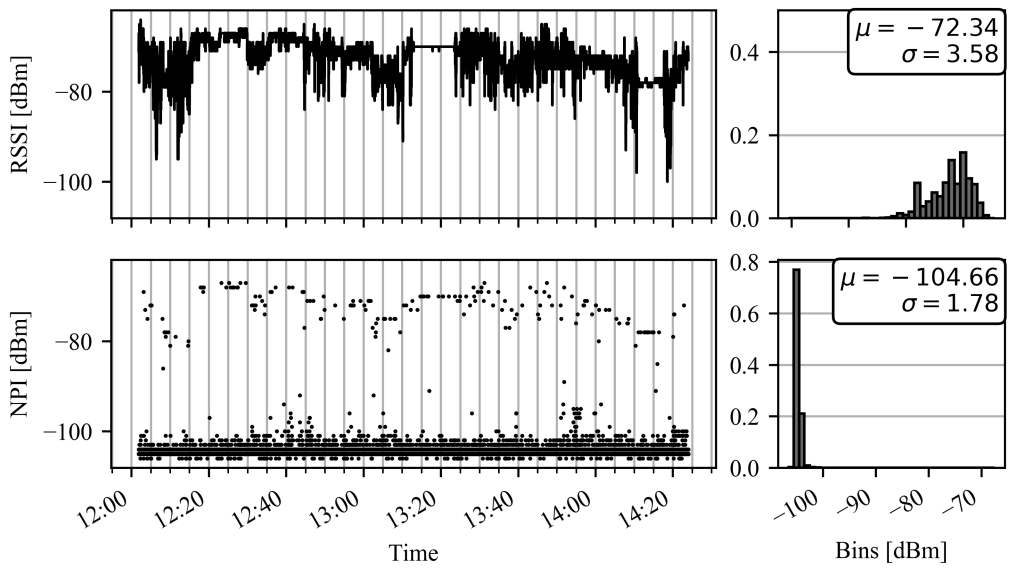

Figure 2.6: Time evolution (left) and empirical distribution (right) of RSSI and NPI measurements in an indoor office environment in the absence of interference [35].

The instantaneous values of RSSI and NPI are affected by the random fading introduced by the environment and by the measurement uncertainty of the RF transceiver. Therefore, as shown in the literature [107] and in our work [35], a precise estimation of SINR via Eq. (2.11) can be obtained using a Kalman Filter. The filter adopts a Linear Time-Invariant (LTI) model for the evolution of the received powers and their measurement, which are filtered, respectively, from fast fading and measurement noise.

Exemplary measurements of RSSI and NPI are shown in Figs. 2.6 and 2.7 for an interference-free and interfered scenarios. Both measurements were performed by Mr. Dauti in a common office room by periodically transmitting packets every 100 ms using a transmission power of 0 dBm and a distance of 3 m between transmitter and receiver. The interference-free measurement was performed in a room without WLAN and Bluetooth interference and captures the effect of moving objects, such as moving employees or lab equipment. The interference measurement was performed in the same propagation environment by adding the WLAN communication between an Access Point (AP) and a client running a video streaming application. The IWSN and WLAN transmissions were performed on channels 23 and 11 defined by the respective communication standards. RSSI measurements show a similar trend in both figures and have similar values of average and standard deviation. Differently, NPI measurements highly reflect the amount of noise in the environment. In Fig. 2.6, NPI measurements evaluate the power of the noise floor, while, in Fig. 2.7, they reflect the presence of interference, showing a higher means and standard deviation. The results show that, via measurements of RSSI and NPI, it is possible to independently characterize the impact of the

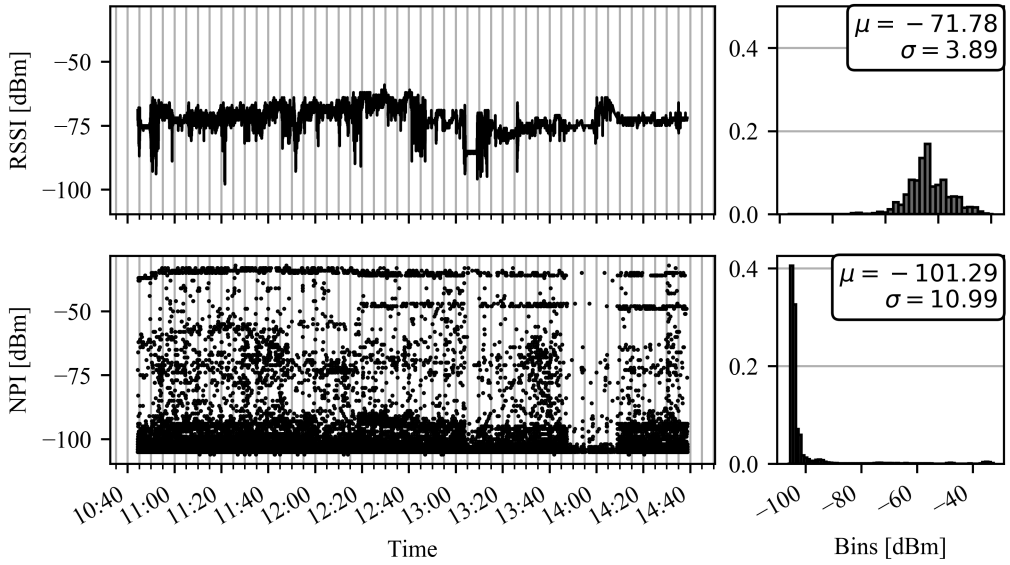

Figure 2.7: Time evolution (left) and empirical distribution (right) of RSSI and NPI measurements in an indoor office environment in the presence of interference [35].

propagation environment, interference, and noise on the communication SINR, thereby on the communication performance of the link.

In experimental deployments an additional step is required to characterize the communication PER from SINR values. In fact, due to device-specific hardware imperfections [155], the analytical model described by Eqs. (2.7)–(2.9) cannot be used and the SINR-PER relationship must be characterized via measurements. This is caused by low-cost IWSN transceivers that introduce device-specific effects that are not captured by the analytical model [155], such as differences in the output power of the transceivers and losses in the RF chain. Therefore, in order to reliably compute PER values from SINR estimations, the SINR-PER relationship must be obtained experimentally. The relationship is obtained by relating the output of the SINR estimator with observed BER and PER values independently. An example of the experimental SINR-BER relationship is shown in Fig. 2.8. The results are based on the measurement campaign performed by Mr. Rammurthy in his Master's Thesis [111] in an underground storage room in the absence of interference. IWSN transmit messages every 30 ms at a distance of 2.5 m with transmission power ranging between 0 and 10 dBm on channel 26. Although the measured values show a similar trend with respect to the IEEE Std. 802.15.4 BER and PER curves, as expected, a noticeable difference can be observed. The results reflect the impact of device-specific hardware imperfections on the Signal-to-Noise-Ratio (SNR)-PER relationship of real IWSN devices [155].

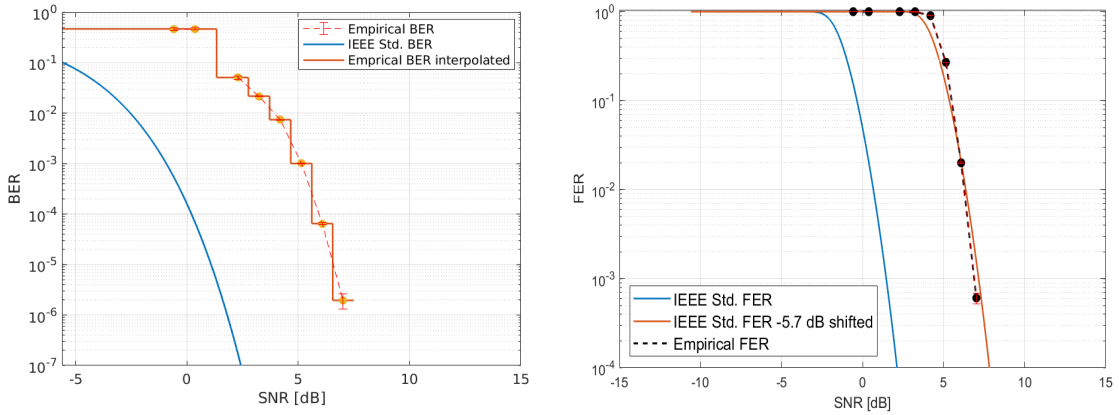

Figure 2.8: Measurement-based SINR-BER (left) and SINR-PER (right) curves compared to the IEEE Std. 802.15.4 [111].

2.3.2 Software-Based Estimation

Software-based link quality estimators compute a PER estimation solely relying on the observation of received packets, thus they do not require measurements from the RF hardware. They are simpler to implement and are used to estimate the channel quality when RSSI measurements are not available. Thanks to these advantages, they received considerable attention in the literature [16]. In general, software-based link quality estimators can be used to calculate different metrics, such as ETX and Four-Bit [16]. These metrics, however, include additional effects in the estimation, such as routing performance and symmetry of links, and are more suitable for routing or application-specific scenarios. In this work, the average link PER is used by RRM schemes for QoS provisioning at the link and networking layers. Therefore, the description of software-based link quality estimators is restricted to PER estimation.

A direct estimation of average PER from received packets can be obtained in a simple way when the wireless environment is static. This applies to scenarios where the devices are stationary and external interference is absent or controlled. In this case, for each time instant k, the average link PER can be calculated as the ratio between the number of received packets $R_k^{\ell,m}$ and the total number of transmitted packets $T_k^{\ell,m}$

$$p_k^{\ell,m} = \frac{T_k^{\ell,m} - R_k^{\ell,m}}{T_k^{\ell,m}}. \tag{2.12}$$

Although it is a simple method, it can be used to effectively characterize the communication quality of static environments if the number of observations $T_k^{\ell,m}$ is large enough. The amount of transmissions to be performed depends on the amount of variability in the wireless medium that is determined by the multipath and interference characteristics. This method was used in Sec. 2.3.1 to obtain a PER estimation to calibrate the SINR-PER relationship.

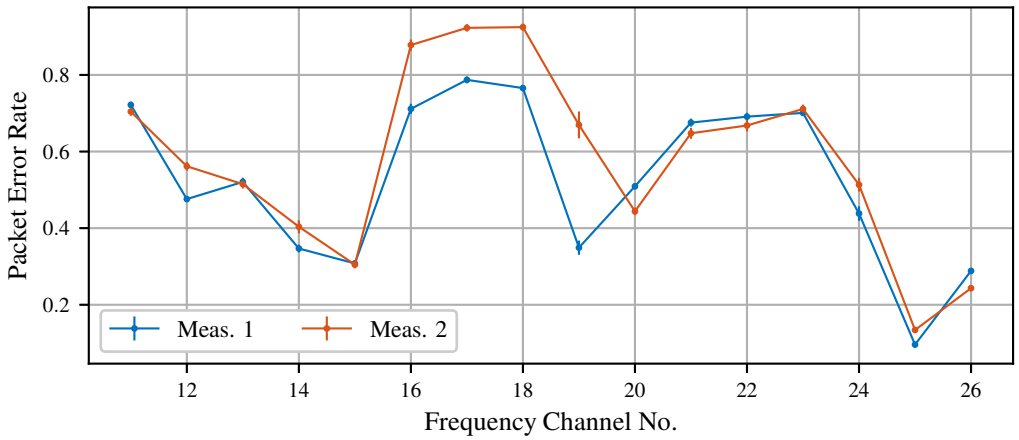

Figure 2.9: Experimental PER measurements of the 2.4 GHz channels of the IEEE Std. 802.15.4 [6].

PER measurements of an exemplary industrial environment were performed in two own works [57, 6], where a mock-up airplane cabin was used for transmission of an IWSN coexisting with a WLAN entertainment system. WLAN networks are represented as several clients connected to multiple different APs operating simultaneously over three non-overlapping channels in the 2.4 GHz ISM band. Communication between WLAN clients and APs follows the full buffer assumption, thus occurs without interruptions. Although this interference scenario is related to a specific application, it is representative of a general type of interference and is used in this work to model the *worst-case industrial interference*. In fact, this arises from WLAN networks coexisting in the factory and, as communication occurs on all WLAN channels without interruption, it represents the worst-case WLAN interference. Fig. 2.9 presents two instances from the measurement campaign in [6], where the average PER and 95% Confidence Interval (C.I.) is shown for all the 16 IWSN channels. For every channel, between 6000 and 9000 packets are sent. Measurement results are grouped in batches of 100 packets that are used to calculate PER values and determine the C.I., which is visualized as a vertical line centered at the average value. The channels are characterized by different PER values caused by the interference of three coexisting WLAN AP that are active close to the WSN channels 12, 17, and 22. The results demonstrate the effectiveness of Eq. (2.12) in characterizing the average PER despite the randomness introduced by multipath and the WLAN interference.

In IWSN deployments, however, the presence of moving objects and the mobility of communication devices introduces time-varying effects on the link quality. In this case, the estimation of average PER must be performed using *moving average* filters. In the following, different moving average filters are presented to estimate the link Packet Delivery Ratio (PDR), i.e., $1 - \text{PER}$. Given a sequence of transmission outcomes $o_i^{\ell,m} \in \{0,1\}$ over link $\{\ell, m\}$,

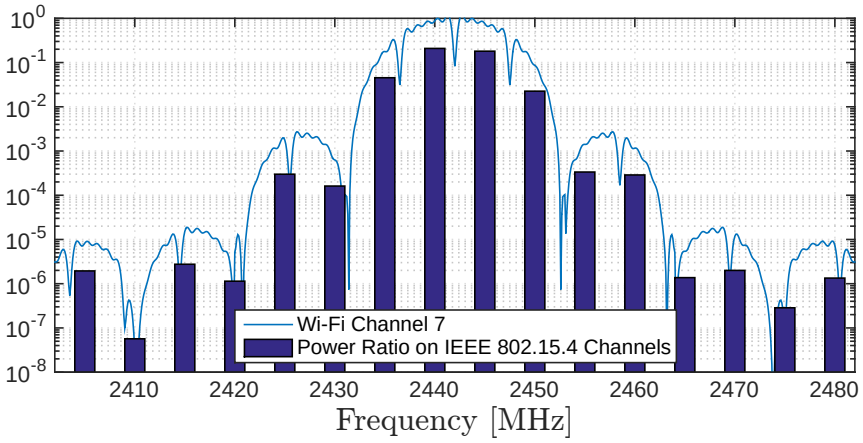

Figure 2.10: Normalized Power Spectral Density (PSD) of the WLAN channel 7. The amount of power received by IWSN devices is represented as histograms on the IEEE 802.15.4 channels.

where 0 indicates failure and 1 indicates success, the k-th PDR estimate of the Cumulative Moving Average (CMA) filter corresponds to the cumulative average of the transmission outcomes until k

$$q^{\ell,m}_{\mathrm{CMA},k} = \frac{1}{k}\sum_{i=1}^{k} o_i. \tag{2.13}$$

Differently, at each transmission, the Simple Moving Average (SMA) estimator computes an average based on the last W samples, where W the window of the SMA estimator. Thus, at the k-th iteration, the SMA filter estimates the PDR

$$q^{\ell,m}_{\mathrm{SMA},k} = \begin{cases} \frac{1}{k}\sum_{i=1}^{k} o_i & 1 \leq k \leq W, \\ \frac{1}{W}\sum_{i=k-W}^{k} o_i & W < k \leq N. \end{cases} \tag{2.14}$$

The Exponentially Weighted Moving Average (EWMA) filter estimates the average PDR by weighting the history of transmission outcomes with exponentially decreasing weights based on the parameter α. At the k-th transmission, the EWMA recursively calculates

$$q^{\ell,m}_{\mathrm{EWMA},k} = \alpha\, o_k + (1-\alpha)\, q^{\ell,m}_{\mathrm{EWMA},k-1}, \tag{2.15}$$

where the weight of the i-th sample at time k is $\alpha(1-\alpha)^i$. The state-of-the-art approach in estimating the average PDR from received packets is the Window Mean with Exponentially Weighted Moving Average (WMEWMA) estimator [146]. The WMEWMA filter combines the strengths of both SMA and EWMA filters. At each transmission, it computes an intermediate estimation with the SMA filter, which is subsequently used by the EWMA filter to compute the final estimation. In this way, the WMEWMA filter benefits from a stable estimation of the SMA filter and a fast reactivity of the EWMA filter.

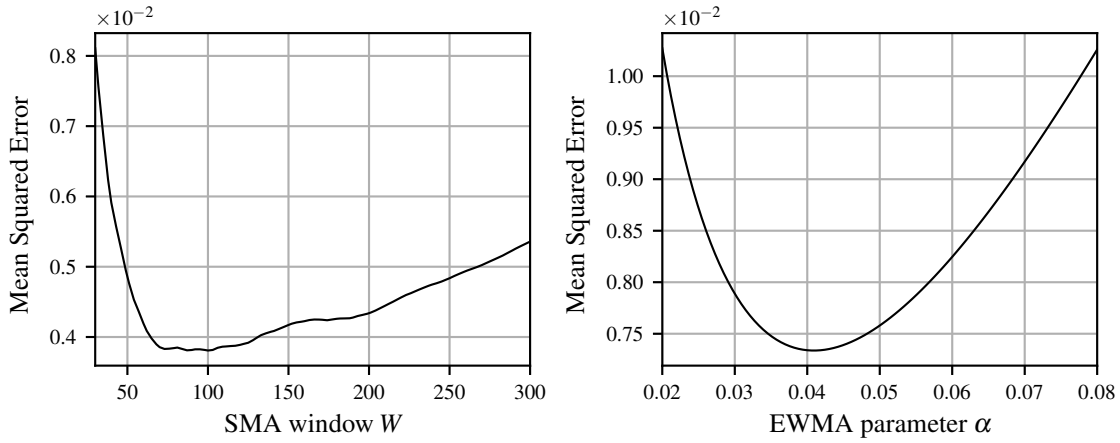

Figure 2.11: Mean Squared Error (MSE) of the SMA and EWMA filters for different values of their parameters W and α for an exemplary trace of the IWSN simulator.

.

To evaluate the performance of the different software-based estimators, simulations of an IWSN environment are performed. Simulations are necessary to obtain a precise performance comparison, as all estimators must experience the same time-varying interference. Via simulations, the performance of the estimators is compared using the exact same channel realizations, thus enabling their direct numerical comparison. The simulator implements the *worst-case industrial interference* shown in Fig. 2.9, where an IWSN coexists with an interfering WLAN entertainment system. The effect is a heavy usage of the ISM band with several highly interfered frequencies and few non interfered frequencies. The WLAN interference is generated with respect to the IEEE 802.11b DSSS modulation replicating the work in [33]. The impact of WLAN interference is evaluated calculating its received power, and by including it in Eqs. (2.4)–(2.9) together with the received power of the IWSN devices. In this way, the resulting PER is calculated. In Fig. 2.10 the normalized PSD of interference from WLAN channel 7 is shown. The amount of interference power received by IWSN devices is represented as histograms on the IWSN channels. The interference from WLAN channel 7 manly affects IWSN channels from 17 to 20 in the frequency range 2435 – 2450 MHz, as WLAN side lobes have a considerably lower power.

The performances of the estimators are evaluated in terms of MSE. Although several metrics can be selected to perform complete characterization of the estimators' performance, such as stability and reactivity [16], in this work, the MSE is selected. This choice is motivated by the fact that the RRM schemes presented in Ch. 3 and 4 rely on a precise estimation of average PER. Fig. 2.11 shows the MSE of the SMA and EWMA estimators for different values of their parameters W and α for an exemplary run of the IWSN simulator. During the experiment, in order to emulate the variation of channel quality in the industrial environment,

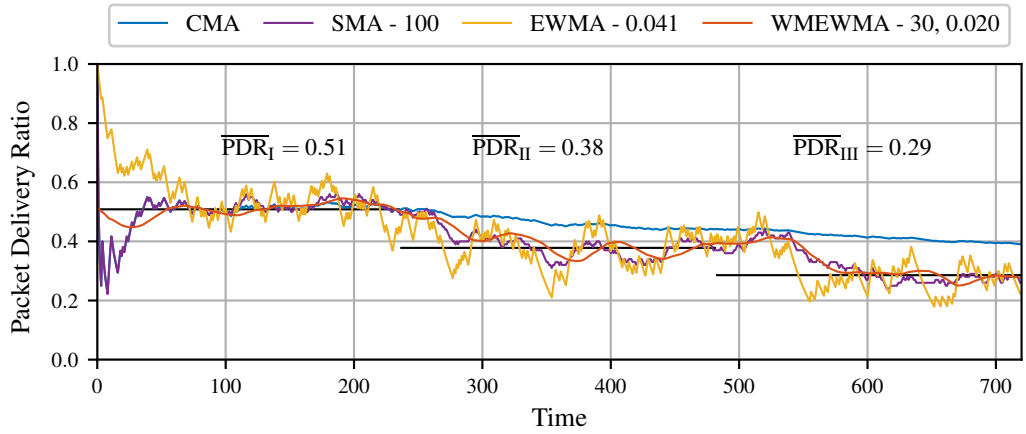

Figure 2.12: Time evolution of software-based link quality estimators of a simulated IWSN and time-varying interference. The filter parameters (W, α) are shown on the right of each filter's name and are the ones that achieve minimum MSE over the entire experiment.

Interference	CMA	SMA - 100	EWMA - 0.041	WMEWMA - 30, 0.020
I	0.0043	0.0045	0.0121	0.0006
II	0.0084	0.0027	0.0042	0.0025
III	0.0175	0.0042	0.0057	0.0042
Mean	0.0101	0.0038	0.0073	0.0025

Table 2.1: MSE achieved by the estimators at the end of each interference interval (I, II, III) for the experiment shown in Fig. 2.12. The filter parameters (W, α) are shown on the right of each filter's name and are the ones that achieve minimum MSE over the entire experiment.

the initial average PDR of 0.51 is changed to 0.38 and 0.28 every 240 transmissions. The results highlight the importance of parameter selection in the precision of the estimator, which regulates the reactivity and stability of the filters to changes in the reception of packets. The figure shows that, while for the EWMA filter the MSE is convex for its parameter α, the SMA filter shows small MSE oscillations varying W.

Fig. 2.12 shows the time evolution of the PDR estimation of all estimators for the experiment of Fig. 2.11. The parameters W and α of the filters are selected as the ones that achieve the minimum MSE. As expected, the CMA filter achieves the worst performance as it does not adapt to a time-varying interference. The estimation of the SMA filter is more stable than the one from the EWMA filter, but is less reactive to PER changes. Thanks to the combination of the SMA and EWMA filters, the WMEWMA filter is able to remove the fluctuations in the estimation while showing good reactivity. In this way, it achieves the best results. This effect is also shown in Tab. 2.1, which summarizes the performance of the estimators in each interference interval (I, II, III). CMA is outperformed by all moving average filters. Although

the SMA, EWMA and WMEWMA filters achieve MSE in the same order of magnitude, the WMEWMA filter is able to achieve performance improvements of 65% and 34% with respect to EWMA and SMA estimators.

2.4 Concluding Remarks

This chapter presented the physical-layer characteristics of IWSN communication and evaluated its physical-layer reliability thanks to an analytical model and measurement results of different estimators. In particular, measurement-based estimators can be used to compute a real-time PER estimation in the presence of interference from measurements of RSSI values. Measurement-based estimators perform RSSI and NPI measurements, which independently characterize the impact of the propagation environment, interference, and noise on the communication reliability. These are fundamental results as they extend existing works that only investigate the estimation of PER in the absence of interference and do not relate RSSI measurements to different propagation scenarios.

Furthermore, the performance of several software-based estimators have been evaluated in the presence of time-varying interference. Simulation results enable the accurate comparison of the estimators' performances and prove the suitability of moving average estimators in tracking the time-varying PER of the channels by solely observing received packets. This is a fundamental achievement as it proves the suitability of sofware-based estimators in the presence of time-varying interference, which was not investigated in the literature. The results of this chapter enable the Network Coordinator (NC) of IWSN to obtain a time-varying estimation of the physical-layer reliability of IWSN transmissions that can be used for RRM. This is exploited by the RRM schemes of Ch. 3 to provide communication QoS in the presence of static and dynamic interference. Furthermore, an accurate estimation of PER is used by the RRM schemes presented in Chapters 4 and 5 to model the communication QoS of feedback-based applications.

Chapter 3

Link-Layer Quality of Service Provisioning in Interfered IWSN

In the previous chapter, the physical-layer reliability of interfered Industrial Wireless Sensor Networks (IWSN) transmissions has been characterized employing measurement- and software-based estimators. This is an important achievement, as it provides the Network Coordinator (NC) with the necessary information to counteract packet loss. In fact, the NC can exploit the estimated Packet Error Rate (PER) values to allocate radio resources in order to construct reliable and timely communication links.

In this chapter, two link-layer Radio Resource Management (RRM) methods for Quality of Service (QoS) provisioning in interfered IWSN are presented. RRM is performed according to the Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) protocol [63], which is described in Sec. 3.2. Radio resources in TSCH are organized according to a Time Division Multiple Access (TDMA) scheme with frequency hopping. This combination provides the determinism and reliability necessary to support time-critical industrial applications and is widely adopted in many IWSN standards [15, 124, 158]. The link-layer QoS performances are quantified by the *reliability* r^{L2}, i.e., percentage of received messages, and *delay* d^{L2} of application packets. The NC is in charge of allocating radio resources to construct communication links that fulfill the QoS requirements. This is achieved by the RRM schemes presented in this chapter by exploiting the PER values computed by the estimators described in Ch. 2. In particular, PER information is used to derive a stochastic model of radio resources that is presented in Sec. 3.2.

Over the last decades, numerous works investigated the problem of QoS provisioning in IWSN. A comprehensive description of the related works is provided in Sec. 3.1 together with their limitations. An issue of existing works is the lack of efficient QoS provisioning methods that exploit unreliable channels. A common approach is to blacklist, i.e., discard, interfered

channels from the available resources to improve the reliability. This solution, however, is very inefficient, as entire channels are discarded even if some packets can be successfully transmitted. A solution to this problem is presented in Sec. 3.3, which is based on the own publication “Reliable Hopping Sequence Design for Highly Interfered Wireless Sensor Networks” [6] developed in collaboration with Mr. Gürsu and Mr. Vilgelm. In this work, a novel algorithm was developed by Mr. Gürsu to design a hopping sequence that combines transmission over interfered channels to provide reliability. I contributed to this work by realizing an IWSN testbed, implementing the RRM schemes in the firmware of the devices, and performing the measurements. Furthermore, I have developed a physical layer IWSN simulator, described in Sec. 3.2.3, that replicates the experimental scenario of the testbed.

As discussed in Sec. 3.1, an alternative approach for QoS provisioning is to schedule the TSCH radio resources. All existing methods following this approach, however, only focus on large multi-hop networks, where interference arises from internal scheduling conflicts. Therefore, QoS provisioning is performed via the deterministic allocation of resources and information about the time-varying PER is not exploited. A solution to this problem is presented in Sec. 3.4, which is based on the own publication “Achieving Hybrid Wired/Wireless Industrial Networks with WDetServ: Reliability-Based Scheduling for Delay Guarantees" [2]. In this publication, I have developed a novel real-time dynamic scheduling protocol, which, thanks to a scheduling algorithm developed in collaboration with the other co-authors, is able to guarantee QoS requirements in the presence of time-varying interference. To verify the results, as described in Sec. 3.2.3, I have extended the simulator presented in [6] to accommodate time-varying interference and allow the dynamic scheduling of the simulated network. In particular, I have developed a Transmission Control Protocol/Internet Protocol (TCP/IP) network interface that allows the NC to interface the real-time multi-hop scheduling framework developed by Mr. Guck [54]. As explained in Sec. 3.4.1, this is achieved by abstracting the TSCH radio resources as a multi-hop network. Finally, Sec. 3.5 summarizes the results presented in this chapter and their relevance for the network-layer RRM schemes presented in the next chapter.

3.1 Related Work

A traditional approach for QoS provisioning in wireless networks is the joint optimization of various reliability-enhancing techniques across different layers. For instance, in [154] different modulation and transmission techniques were analyzed for cluster-based Wireless Sensor Networks (WSN), while, in [148], link layer QoS in mobile networks was improved by modelling the effects of Multiple-Input Multiple-Output (MIMO) diversity schemes and

Adaptive Modulation and Coding (AMC). These methods, however, require highly capable hardware and are not covered by IWSN communication standards. Besides and complementary to MIMO and AMC techniques, QoS can be provisioned via appropriate management of wireless resources through scheduling.

In general, scheduling of wireless resources is an NP-hard combinatorial problem [55] of mapping the demands to the time-frequency resources. In contrast to IWSN, it is a mature research topic in cellular networks, such as 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE). The scheduling problem in LTE is typically considered as a trade-off between resource utilization and fairness. Schedulers in LTE could be classified into two types: best effort and QoS-aware. Maximum Throughput (MT) and Proportional Fair (PF) are classic examples of the first type. MT allocates the resources always to the users with the best channel quality, hence, maximizing the total resource utilization under the demand saturation assumption. On the other hand, PF weights the current channel quality and the transmission history of a user, achieving a balance between fairness and utilization. QoS-aware schedulers are typically based on the prioritization of packets with strict delay or data rate requirements [7, 23, 84]. However, in LTE, reliability is outside the scheduling problem's scope, as it is achieved by other techniques, such as hybrid automatic repeat request and AMC.

In the last years, the newly developed IWSN standard protocols for industrial communications have drawn a lot of attention to the research of QoS provisioning in IWSN. This aspect is further discussed in this section, which is organized as follows. Sec. 3.1.2 presents the relevant static RRM schemes and blacklisting strategies, while Sec. 3.1.2 the existing work in the area of dynamic scheduling of IWSN.

3.1.1 Static Radio Resource Management

The availability of multiple channels and a channel hopping strategy, which is enabled by the TSCH medium access, has shown superior reliability performance in comparison with adaptive single-channel routing techniques in sparse networks [52]. For this reason, a number of studies has been dedicated to channel hopping. Its impact on link connectivity and routing in a meshed network has been discussed in [142]. The authors show the benefits of deploying channel hopping and discuss whitelisting – reducing the hopping sequence to non-interfered channels. In [139], the authors have compared scheduled and random access-based MAC for multi-hop TSCH, and have shown that they deliver similar reliability for low-load networks. While these studies provide insights into the benefits of channel hopping, they do not address design of the hopping sequence.

The state-of-the-art technique of optimizing channel hopping is blacklisting – excluding highly interfered channels from the hopping sequence. The practical hardware aspects of

hopping sequence generation were addressed in [121]. Adaptive TSCH has been introduced in [40] and it extends TSCH with dynamic blacklisting. The idea was further developed in several works. LABEL [75] is a distributed, link-based blacklisting method, however, it introduces packet loss from other nodes using the same channel. Similarly, LOST [159] constructs a distributed blacklist and allocates time slots for QoS provisioning, however it suffers from internal interference and only static channel qualities are considered. MABO [50] solves the problem of internal interference centrally allocating the channels. In all these methods, however, the QoS requirements of the application are not considered by the RRM scheme. Differently, Jung et al. [67] proposed a distributed scheduler for the energy efficient QoS provisioning in TSCH. Their scheduler, however, only considers deterministic transmissions and packet loss that arises from internal interference.

Although several IWSN protocols propose the assignment of multiple channels for communication [128], none of the state-of-the-art solutions address the issue that blacklisting is shrinking the number of available resources for parallel communication. Thanks to the contribution presented in Sec. 3.3, which is based on the own publication "Reliable Hopping Sequence Design for Highly Interfered Wireless Sensor Networks" [6], the design of a reliable hopping sequence is performed *without sacrificing resources*. In the previous work with Mr. Gürsu and Mr. Vilgelm [57], we noticed from experimental results that interfered channels are not fully blocked, and non-interfered channels are affected by small, but not negligible, packet drops. Thanks to presented whitening algorithm [6] presented in Sec. 3.3, the capacity of all channels is used to improve the latency and reliability of the links in the IWSN at the expense of an increased number of transmission attempts.

3.1.2 Dynamic Radio Resource Management

Since the release of the TSCH Medium Access and the 6TiSCH architecture [41], a considerable amount of work was performed tackling the problem of scheduling in IWSN. Some methods implement dynamic scheduling in an *autonomous* fashion. IWSN devices in Orchestra [42], e-TSCH-Orch [112], and Alice [72] autonomously build their schedules without interacting with a central entity or the neighbours. In their schedules, however, only the amount and priority of traffic is taken into account and no QoS is considered.

Several *distributed* schedulers were proposed in TSCH. Accettura et al. [8, 9] proposed DeTAS, a traffic-aware scheduler designed for low latency, small queue utilization. Similarly, Soua et al. developed Wave [129] and DiSCA [126], traffic-aware schedulers for convergecast to reduce data gathering delays. Morell et al. [89] investigated a label-switching approach to distributed scheduling in TSCH. In all these cases, however, QoS provisioning is not considered. Palattella et al. [98] developed an On-the-Fly scheduler that adapts the schedule

based on the traffic load. Channels with different Packet Delivery Ratio (PDR) are considered, but their performance is not taken into account by the scheduler. In DeBraS [90], a broadcast-based scheduling algorithm was designed to improve scalability in large networks. Also in this case, however, QoS is not considered. DeAMON [10] is a traffic-aware dynamic scheduler able to adapt to topology changes. Reliable packet delivery, however, is only ensured via predefined over-provisioning of cells. Hannachi et al. [60] proposed a dynamic scheduler that employs combinatorial design to achieve high throughput and reduce internal collisions in tree network topologies. Similarly, LaDis [58] is a low-latency scheduling method developed for data aggregations in tree topologies. Furthermore, the ReSF [34] algorithm constructs minimal-latency schedules for periodic sensor data transmission. In all these cases, however, reliability is provided via re-transmissions without taking the physical-layer PER into account and considering static propagation environments. Most of the dynamic decentralized solutions provide traffic-based scheduling in large multi-hop networks, which is different from the considered IWSN network that is organized as a star topology. For this reason, all the existing methods only consider packet loss as a result of internal interference caused by frequency reuse and do not investigate external interference.

Furthermore, several *centralized* traffic-based schedulers have been studied for TSCH. The TASA [97] algorithm optimally allocates resources based on the traffic demand and network topology. Soua et al. developed the MODESA [127] and MUSIKA [125] scheduling algorithms to achieve optimal communication delay and traffic-aware interference coordination in multi-hop convergecast networks. Farias et al. [44] developed a centralized scheduler that allocates resources based on Routing Protocol for Low-Power and Lossy Networks (RPL) queue metric. In all these works, however, packet loss is not considered. Jin et al. developed AMUS [66], a reliable and low latency scheduler for multi-hop networks. Reliability is provided via the allocation of re-transmissions over a pre-computed tentative schedule that is used in case of packet loss. However, the schedules are constructed without taking the physical-layer reliability information into account. Authors in [29] proposed scheduling algorithms for data collection in IWSN, but no packet loss and QoS requirements were considered. PriMulA [103] is a scheduler that provides low latency, traffic prioritization, and reliability via dynamically blacklisting interfered channels. Also in this case, QoS requirements are not considered by the scheduling procedure. In [131], a throughput-optimal dynamic scheduling scheme was derived using information about the link quality of the channels. Their method, however, did not address the issue of QoS provisioning. Existing centralized schedulers aim at building conflict-free schedules in multi-hop networks in order to avoid internal interference. Therefore, the effect of external interference is not taken into account and schedules are not constructed based on the physical-layer reliability. Furthermore, most of the existing works

do not construct schedules taking into account the link-layer delay and reliability requirements of industrial applications.

Recent resource allocation mechanisms dynamically estimate the physical-layer performance of the channels, which is used, however, to blacklist unreliable channels. In LABEL [75], a Window Mean with Exponentially Weighted Moving Average (WMEWMA) filter is used to estimate the performance of the channels in order to construct the hopping sequence. In MABO [50], channel quality estimation is performed as a Multi-Armed Bandwidth problem and, in AMABO [74], it is used to dynamically allocate channels for transmission. In [135], centralised and distributed channel quality estimation is performed for the same purpose. All these methods, however, transmit over interference-free channels and assume that the reliability requirements are always fulfilled. Therefore, the link-layer delay and reliability requirements are not addressed by the RRM schemes. Differently than the available state-of-the-art, the scheduler presented in Sec. 3.4, which is based on the own publication “Achieving Hybrid Wired/Wireless Industrial Networks with WDetServ: Reliability-Based Scheduling for Delay Guarantees" [2], can adapt to different *interference* scenarios and provides the link-layer *delay and reliability requirements* of industrial applications.

3.2 Link-Layer Model

The most important task of the link-layer in IWSN is the MAC, which coordinates the transmissions among the devices in the network. By determining the time and the frequency resources used by the devices, the MAC prevents uncontrolled loss of data caused by collisions, i.e., the simultaneous transmission of multiple devices in the same radio range. Furthermore, it determines the number of transmission opportunities and the physical-layer reliability of the communication channel. Therefore, the choice of MAC scheme strongly impacts the performance of the network, affecting throughput, delay, reliability, and energy efficiency. For this reason, numerous MAC protocols has been investigated over the last decades to address the most diverse requirements. Sec. 3.2.1 provides an overview of the existing MAC protocols together with a discussion of their suitability for IWSN.

A fundamental requirement of IWSN MAC schemes is to support the stringent delay and reliability requirements of industrial feedback systems. A discussed in Sec. 3.2.1, this can be achieved combining a time slotted medium access with frequency hopping. The former enables the deterministic allocation of resources in the network, while the latter enhances reliability via frequency diversity. This MAC scheme is employed by all IWSN standards [15, 124, 158] and by the RRM algorithms presented in work. A detailed model of the link-layer radio resources following this scheme is provided in Sec. 3.2.2. In contrast with the existing

approaches, the radio resources follow a stochastic model that is based on the physical-layer PER provided by the estimators described in Ch. 2. The stochastic model is exploited by the RRM schemes of this chapter to achieve the required link-layer QoS performance. In order to evaluate the performance of the presented RRM schemes, I have implemented a physical- and link-layer simulator of the IWSN, which is presented in Sec. 3.2.3. The simulator implements the stochastic radio resource model described in Sec. 3.2.2, as well as the IWSN physical-layer model and PER estimation techniques presented in Ch. 2.

3.2.1 Overview of MAC Protocols for IWSN

MAC protocols belong to two major schemes: contention-based and contention-free protocols. Contention-free protocols regulate the access by dividing the available radio resources in chunks and organizing the transmissions via schedules. For example, TDMA protocols discretize time into slots and group them in TDMA frames that periodically succeed one after the other. By assigning different time slots to the transmission contenders, the access to the shared channel is fully regulated and collision is avoided. On the other hand, in contention-based protocols, transmissions do not follow a predefined schedule and the risk of collisions is deliberately taken. Whenever a collision occurs, the packet is re-transmitted following a random backoff mechanism. This approach is considered in typical contention-based algorithms such as Additive Links On-line Hawaii Area (ALOHA) and Carries Sense Multiple Access (CSMA). To improve the performances of these schemes, CSMA introduces Collision Detection (CD), the operation of sensing the medium before attempting a transmission. If the channel is busy, a preventive backoff is performed. Further improvements to CSMA are brought by the Collision Avoidance (CA) mechanism, which introduces additional messages to the protocol in order to reduce the effect of the well-known hidden-terminals problem [12, p. 81]. Finally, the class of hybrid MAC protocols combines the ideas of contention-based and contention-free protocols in order to achieve increased performance and flexibility.

The main difference behind contention-based and contention-free MAC schemes lies behind the information that the protocols use to organize the access to the channel. On the one hand, contention-based schemes require little or no central coordination between the nodes, which allows them to be very efficient in terms of protocol overhead and scalability. On the other hand, contention-free protocols require that all the nodes comply with a centralized schedule, which needs to computed, distributed, and maintained. Additionally, when a TDMA scheme is used, which happens in the majority of cases, accurate time synchronization is required in the entire network. Although the amount of overhead is considerably larger than contention-based schemes, contention-free protocols enable the predictable and efficient allocation of network resources. In fact, contention-free schemes perform very well when

the networks is highly loaded and can provide predefined and deterministic communication QoS. In contrast, the collisions introduced by contention-based schemes make these schemes suitable for low network utilization and introduce random fluctuations in the offered QoS.

The literature extensively investigated WSN MAC schemes tailored to the QoS requirements of diverse WSN applications [61]. The most popular contention-based MAC protocols are S-MAC [143] and B-MAC [105].S-MAC [143] introduces synchronized sleeping and wake-up cycles in the medium access in order to lower contention and improve energy efficiency. B-MAC [105] enables the asynchronous operation via Low Power Listening (LPL) and Clear Channel Assessment (CCA) aimed at improving energy efficiency and channel utilization. Popular contention-free MAC protocols are TRAMA [110], which enables adaptive scheduling in the network thanks to a Schedule Exchange Protocol (SEP) that allocates time slot based on traffic priority, and TSCH [63], which combines a centralized TDMA medium access with frequency hopping. An example of popular hybrid MAC protocol is the Institute of Electrical and Electronics Engineers (IEEE) Std. 802.15.4 [32], which combines a contention-free and a contention access period in a TDMA frame called superframe.

The application requirements guide the selection of the MAC protocol. While traditional WSN are mainly deployed for environment monitoring, which motivate the predilection of energy efficiency over QoS, IWSN must support the challenging QoS requirements of industrial applications. The capability of guaranteeing reliable packet delivery within strict latency bounds in a dense network is a key requirement of all industrial applications. For this reason, the vast majority of MAC schemes developed for IWSN focus on the provisioning of deterministic latencies and high reliability. Although several MAC protocols address timely and reliable communication, only few MAC schemes are applicable to the industrial scenario [130]. In fact, additionally to a TDMA scheme that achieves deterministic latencies, multi-channel capabilities and the provisioning of link quality information are required to enhance reliability.

Standard MAC protocols that fulfill these requirements are WirelessHART [124], ISA 100.11a [15], WIA-PA [158], and IEEE 802.15.4e TSCH [63]. All these protocols follow a similar time-frequency structure, c.f. Fig. 3.1, which is presented in Sec. 3.2.2. They adopt a synchronized TDMA medium access, multiple frequencies, and the allocation of radio resources is determined by a central entity, the NC. The allocation of the time slots and frequencies to the devices is performed via schedules, which are distributed by the NC via periodic beacons. Thanks to this radio resource model, bounded latencies can be achieved by scheduling the time slots and reliability by exploiting frequency diversity across the available channels. The standards foresee centralized RRM via scheduling and they only define a minimal scheduling algorithm. In this way, the design of appropriate RRM mechanisms can

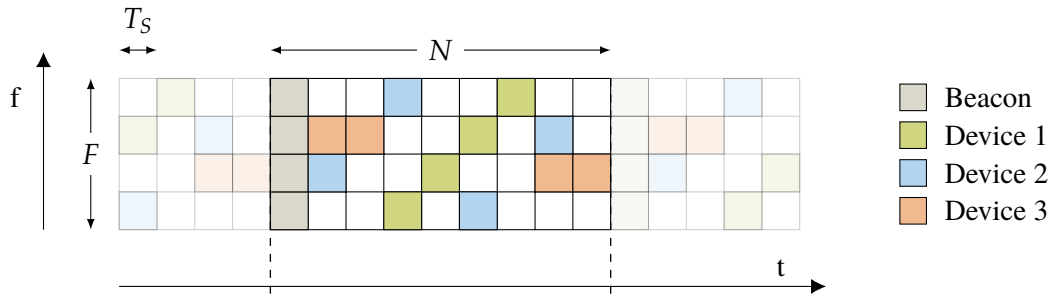

Figure 3.1: Model of the TDMA, multi-channel resource allocation in IWSN MAC protocols.

be performed based on the requirements of each industrial application. In this work, the definition of scheduling algorithms that fulfill the QoS requirements of industrial applications is discussed in Sec. 3.3, 3.4, and in Ch. 4.

3.2.2 Stochastic Radio Resource Model

The radio resources of the network are organized according to a time-frequency resource grid represented in Fig. 3.1. Time is discretized in slots of duration T_S and synchronized over the network. Within one time slot there is enough time for the transmission of one data packet and its Acknowledgment (ACK). Time slots are organized in cyclic frames consisting in N number of slots. In this way, time slots can be allocated to the devices in order to fulfil deterministic delay requirements. Synchronization in the network is achieved by means of the Absolute Sequence Number (ASN) counter, which indicates the global time and is propagated in the network by the NC via a broadcast beacon called Enhanced Beacon (EB). In the frequency domain, continuous frequency hopping exploits the frequency diversity of the F available channels, thereby accomplishing uncorrelated transmissions. Frequency hopping is achieved by means of the hopping sequence H, a lookup table containing a predefined sequence of channels selected for consecutive transmissions. To determine the frequency to be used at time k, the lookup table is indexed using the ASN and the offset o_j via the modulo operation

$$f_k = H\left[\left(\mathrm{ASN}_k + o_j\right) \mod F\right], \tag{3.1}$$

where F is the total number of available channels and $H \in \mathbb{N}^F$ is a given hopping sequence. The offset o_j is used to generate shifted versions of the hopping sequence and enables IWSN devices to transmit at the same time over different frequencies.

The time-frequency structure enables a flexible allocation of radio resources to the devices in the network. Each time-frequency combination from the grid, also referred to as Resource Block (RB), can be reserved or shared, thus used in a contention-free or contention-based access. The allocation of RBs to the devices is performed via *schedules*, depicted as squares

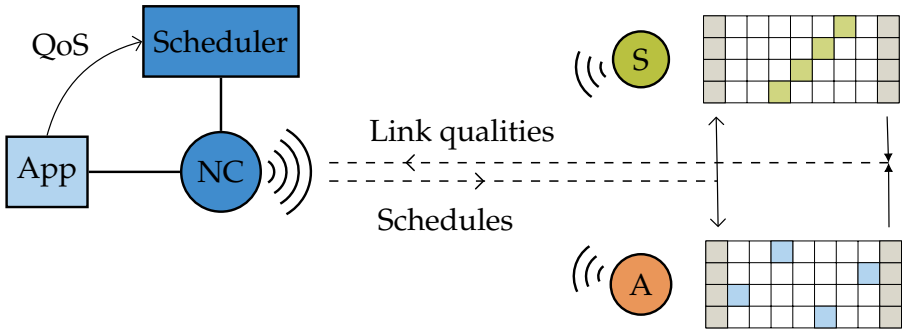

Figure 3.2: Dynamic scheduling procedure in IWSN. At the end of every frame, link qualities are sent to the NC, which replies with new schedules at beginning of next frame.

of the same color in Fig 3.1, which are distributed by the NC via EBs. The NC is in charge of defining and distributing schedules to ensure that the QoS requirements of the industrial applications are fulfilled. This procedure assumes a cross layer information exchange between the application and MAC layer and is depicted in Fig. 3.2. At the end of each TDMA frame, the current PER of all links in the network is transmitted to the NC, which, aware of the QoS requirements of the application, distributes new schedules at the beginning of the next frame via the EB.

Schedules that provide reliable and time-bounded communication require information about the physical-layer reliability of the communication channels. As discussed in Ch. 2, due to the unpredictable nature of the wireless medium, it is not possible to obtain a deterministic characterization of the communication quality even from experimental measurements. Therefore, RRM must be performed using a stochastic model of the link performance. The random outcome generated by a packet transmission over the communication link $\{\ell, m\}$ on RB $s_{i,j}^{\ell,m} = (t_i, f_j)$, where t_i is the slot number and f_j the frequency corresponding to the offset o_j from the hopping sequence, is modelled as a Bernoulli r.v. that follows the average PDR of the link $q_{i,j}^{\ell,m}$

$$s_{i,j}^{\ell,m} = \begin{cases} 1 & \text{with } q_{i,j}^{\ell,m}, \\ 0 & \text{with } p_{i,j}^{\ell,m}. \end{cases} \tag{3.2}$$

The RRM schemes presented in Sec. 3.3 and 3.4 make use of Eq. (3.1) and (3.2) to determine the aggregate performance of multiple RBs and compute the hopping sequence and schedules according to the QoS requirements. In this way, the PER information provided by link quality estimators described in Ch. 2 is used for RRM at the link layer.

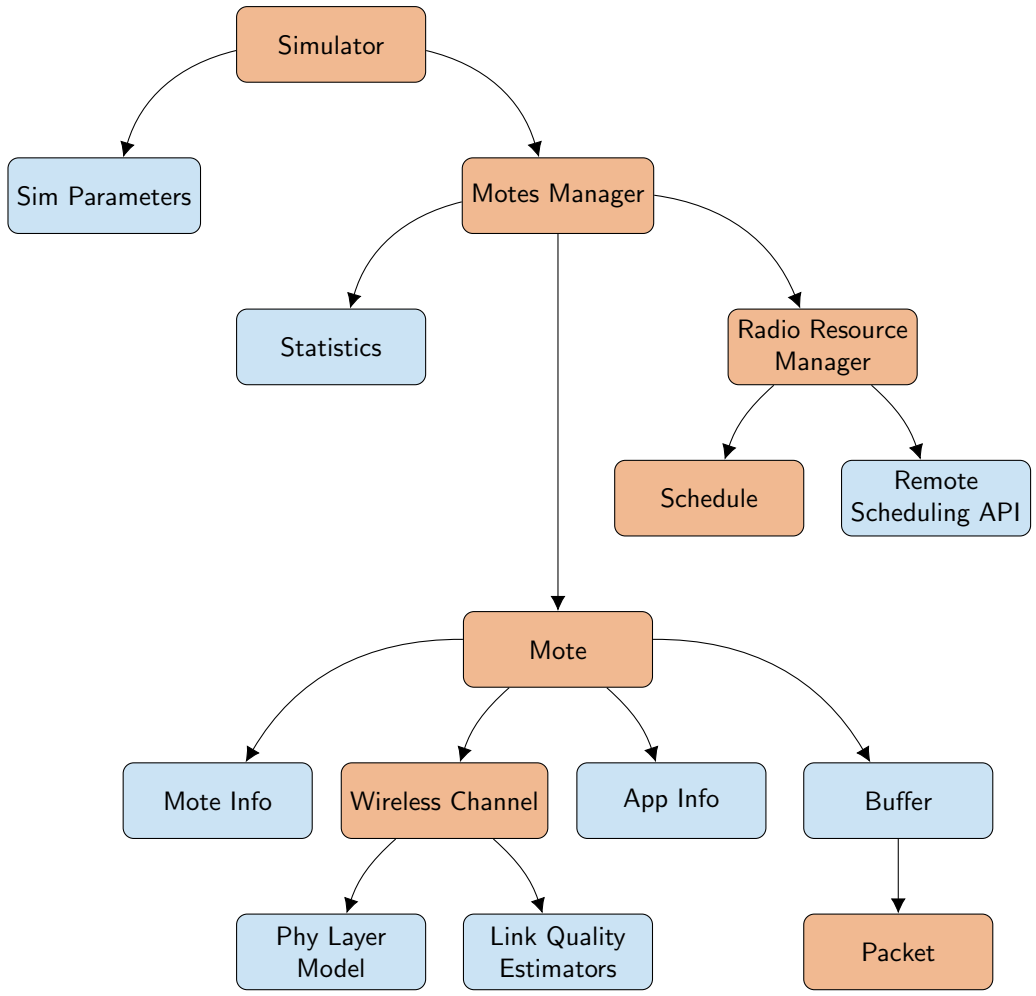

Figure 3.3: Software architecture of the Python IWSN Simulator.

3.2.3 Link-Layer Simulator

In order to implement and evaluate the RRM schemes described in this chapter, I have developed a physical- and link-layer Python simulator. The purpose of the simulator is to recreate the transmissions within an IWSN by accurately capturing the physical-layer characteristics and the interference scenario of the industrial environment described in Ch. 2. In this way, simulations do not only verify the correctness of the described RRM schemes, but provide a performance evaluation of IWSN communication in an industrial environment and of link quality estimation for RRM. The overall architecture of the simulator is depicted in Fig. 3.3.

The simulator relies on seven main classes, coloured in orange in Fig. 3.3. The *Simulator* class contains the overall simulation parameters, such as the experiment duration, number of devices and interference type. Furthermore, it executes the main simulation steps over time

and stores its results. For each TDMA frame, it triggers link quality information updates, the generation of new application packets, and the transmission of packets. The *Motes Manager* class coordinates all the devices in the IWSN and is in charge of interfacing the Radio Resource Manager to perform dynamic scheduling. In particular, it instantiates the devices and triggers schedule calculations and updates when new link quality information is available. For each device, it defines the position, address, schedule, and channel parameters. The *Radio Resource Manager* class implements the TCP/IP interface to communicate with the remote dynamic scheduler. The *Schedule* class contains the configuration parameters of the TSCH medium access, such as frame length, time slot duration and hopping sequence, and contains a list of RBs allocated by the schedule. The *Mote* class contains device-specific information, such as the position, address, wireless propagation parameters, and application characteristics, i.e., the traffic type, destination address, and QoS requirements. Additionally, it implements a First in, first out (FIFO) buffer for the transmission of data packets and performs the fundamental operations of traffic generation, buffer management, and transmission of packets. The information relative to each packet and its transmission attempts is contained in the *Packet* class. For each application packet, it stores timestamps for its generation and reception, the sequence number, and information about each transmission, such as the time slot, frequency, and Signal-to-Interference-and-Noise-Ratio (SINR) value. The *Wireless Channel* class contains all the physical-layer parameters of IWSN and Wireless Local-Area Network (WLAN) devices. For each transmission, it calculates the received power of IWSN and WLAN transmissions, and calculates the resulting SINR, Bit Error Rate (BER), and PER values as described in Ch. 2. In particular, the received power of IWSN transmission is calculated according to Eqs. (2.1) and (2.3) for a path loss exponent of $\eta = 2.2$, a multipath variance of $\sigma^2_{\mathrm{dB}} = 6.2$ dB, and noise power $r^{\mathrm{N}} = -90$ dBm. The calculation of received power of WLAN devices follows the same procedure described in Sec. 2.3.2. The calculation of SINR, BER, and PER values is performed according to Eqs. (2.4)–(2.9). Furthermore, the Wireless Channel class performs the operation of link quality estimation. It implements the Cumulative Moving Average (CMA), Simple Moving Average (SMA), and Exponentially Weighted Moving Average (EWMA) software-based filters described in Sec. 2.3.2 and stores their statistics.

3.3 QoS Provisioning Under Statically Interfered Channels

In an industrial environment, interference can arise from the transmission of coexisting wireless technologies, such as WLAN or Bluetooth [109], and from production machines that generate Radio Frequency (RF) emissions in the Industrial Scientific and Medical (ISM)

band. Although interference usually changes over time, in specific scenarios, such as isolated production areas or in controlled environments, it can have a constant impact on the communication quality. In such cases, information about interference can be obtained via experimental measurements and used for the allocation of radio resources in the network at design-time.

As discussed in Sec. 3.1, the state-of-the-art solution in providing reliability in the presence of static interference is blacklisting, i.e., excluding bad or interfered channels from the hopping sequence. This strategy, however, is very inefficient, as scarce transmission frequencies are discarded although not fully blocked. This is particularly problematic in IWSN, as several devices share the same resources to transmit the packets of industrial applications. To overcome this limitation, interfered channels can be used in combination with white, i.e., non-interfered, channels to increase the number of transmission opportunities of the devices in the network and improve the application reliability within a given deadline. To achieve this, my co-authors and I proposed the *whitening* hopping sequence design algorithm [6], which is described in Sec. 3.3.1.

3.3.1 Whitening Hopping Sequence Design

Whitening is the reliable hopping sequence design where *white*, i.e., non-interfered, channels are fairly provisioned over consecutive transmission opportunities within the deadline. The link-layer *reliability* is defined as the percentage of application packets delivered within the application's deadline. The number of *transmission opportunities* is equal to N_{D}, i.e., the amount of TSCH RBs available within the deadline. Thanks to whitening, the industrial QoS requirements of industrial applications can be guaranteed by allocating transmission opportunities over interfered and white channels fulfilling the target reliability within the deadline.

In the work with Gürsu et al. [57], we have shown that, in order to guarantee a fair allocation of white channels, each combination of transmission opportunities from the hopping sequence should have the same amount of white channels. This is the intuition behind the whitening hopping sequence design algorithm. The algorithm initiates by considering only one RB for every TDMA frame. In this way, there will be exactly F sets $\mathcal{R}^i$, $i = 1, 2, \ldots, F$ representing all the possible resource allocations in a TDMA frame. Furthermore, the indexing of the hopping sequence from Eq. (3.1) can be simplified substituting $ASN_j = jN$, which yields

$$f_k = H\left[(jN + o_i) \mod F\right]. \tag{3.3}$$

Thus, every new transmission opportunity j indexes a channel from an N-shifted version of the hopping sequence.

Algorithm 1 Whitening Algorithm.

$H = (1, 2, \ldots, F)$ ▷ Initialization
$\Delta = (\delta_1, \delta_2, \ldots, \delta_F), \delta_n \leftarrow 0, \forall n \in \{1, ..., F\}$
$\mathcal{W} = \left\{ \mathrm{ch}_1^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{W}}, \ldots, \mathrm{ch}_{N_{\mathrm{W}}}^{\mathrm{W}} \right\}$
$\mathcal{I} = \left\{ \mathrm{ch}_1^{\mathrm{I}}, \mathrm{ch}_2^{\mathrm{I}}, \ldots, \mathrm{ch}_{N_{\mathrm{I}}}^{\mathrm{I}} \right\}$
$\mathcal{R}^n \leftarrow \emptyset, \forall n \in \{1, ..., F\}$
$\mathcal{P} \leftarrow \emptyset$
for $i = 1$ **to** F **do**
    for $j = 0$ **to** $N_{\mathrm{D}} - 1$ **do**
        $\phi \leftarrow H\left[(j \cdot N + o_i) \mod F\right]$
        $\mathcal{R}^i \leftarrow \mathcal{R}^i \cup \{\phi\}$
for $k = 1$ **to** N_{W} **do** ▷ Weights
    $\phi \leftarrow H(\arg\min \Delta)$
    $\delta_\phi = \delta_{\max} + 1$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\phi\}$
    for $m = 1$ **to** F **do**
        if $\phi \in \mathcal{R}^m$ **then**
            $\mathcal{U} \leftarrow \mathcal{R}^m \setminus \mathcal{P}$
            for all $u \in \mathcal{U}$ **do**
                $\delta_u = \delta_u + \delta_{\min}$
for $n = 1$ **to** F **do** ▷ Finalization
    if $\delta_n > \delta_{\max}$ **then**
        $ch \leftarrow \mathrm{rand}(\mathcal{W})$
        $\mathcal{W} \leftarrow \mathcal{W} \setminus \{ch\}$
    else
        $ch \leftarrow \mathrm{rand}(\mathcal{I})$
        $\mathcal{I} \leftarrow \mathcal{I} \setminus \{ch\}$
    $H(n) \leftarrow ch$

The whitening algorithm, summarized in Alg. 1, defines the order of white channels in the hopping sequence in order to fairly distribute the channels with high physical-layer reliability to all the resource allocations of the devices in the network. A resource allocation $\mathcal{R}^i$ is a set of transmission opportunities towards the deadline where channel occurrences are mapped to placeholders drawn from the hopping sequence H. A placeholder is a number that corresponds to an available place for a channel. Whitening removes the effect of initial channel selection for a each packet as it guarantees the presence of a white channel over the different transmission opportunities of a resource allocation.

During the *Initialization* phase, the resource allocations $\mathcal{R}_i$ are generated. For each channel offset o_j in the hopping sequence, a set $\mathcal{R}^i$ containing N_{D} placeholders is drawn from the hopping sequence using Eq. (3.3). In the *Weights* phase, the whitening algorithm

	Initialization	Weights	Weights	Finalization
H	$(1,2,3,4)$	$(1,2,3,4)$	$(1,2,3,4)$	$(\mathrm{ch}_1^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{I}}, \mathrm{ch}_1^{\mathrm{I}})$
Δ	$(0,0,0,0)$	$(9,0,2,0)$	$(9,9,2,2)$	$(9,9,2,2)$
$\mathcal{R}^1$	$\{1,3\}$	$\{1,3\}$	$\{1,3\}$	$\{\mathrm{ch}_1^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{I}}\}$
$\mathcal{R}^2$	$\{2,4\}$	$\{2,4\}$	$\{2,4\}$	$\{\mathrm{ch}_2^{\mathrm{W}}, \mathrm{ch}_1^{\mathrm{I}}\}$
$\mathcal{R}^3$	$\{3,1\}$	$\{3,1\}$	$\{3,1\}$	$\{\mathrm{ch}_2^{\mathrm{I}}, \mathrm{ch}_1^{\mathrm{W}}\}$
$\mathcal{R}^4$	$\{4,2\}$	$\{4,2\}$	$\{4,2\}$	$\{\mathrm{ch}_1^{\mathrm{I}}, \mathrm{ch}_2^{\mathrm{W}}\}$

Figure 3.4: Examplary application of the Whitening Algorithm for $F = 4$, $\mathcal{W} = \{\mathrm{ch}_1^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{W}}\}$, $\mathcal{I} = \{\mathrm{ch}_1^{\mathrm{I}}, \mathrm{ch}_2^{\mathrm{I}}\}$, $N = 2$, $N_{\mathrm{D}} = 2$.

associates to every placeholder in H a weight δ_i from the set $\Delta = \{\delta_1, \ldots, \delta_F\}$ initially set to zero. The algorithm allocates white channels to placeholders selecting the placeholder ϕ from H with minimum corresponding weight. At every step, a white channel is allocated to ϕ, and ϕ is added to the set $\mathcal{P}$ containing allocated white channels. Then, for every set $\mathcal{R}^i$ containing ϕ, the weights of the non-white placeholders are incremented by $\delta_{\min}$. Finally, the weight of the placeholder δ_ϕ is set to $\delta_{\max} + 1$, where $\delta_{\max} = FN_{\mathrm{W}}\delta_{\min}$. N_{W} is the number of white channels and corresponds to the maximum number of increments of non-white placeholders during the Weights phase. The *Finalization* phase of the algorithm allocates white and interfered channels to the whitening hopping sequence. All channels are clustered into two disjoint sets of white $\mathcal{W}$ and interfered $\mathcal{I}$ channels using the following relationship,

$$1 - p_j^{\mathrm{I}} < \frac{1 - p_i^{\mathrm{W}}}{\alpha}, \quad \forall i, j \in 1, \ldots, F, \tag{3.4}$$

where α is the channel quality separation criterion, p is the PER defined in Eq. (2.9) and its index identifies a frequency channel, and W and I indicate white or interfered channels. It is assumed that the PER of the channels is almost static and that small variations are accommodated by the threshold α. For every placeholder in the hopping sequence, a channel is allocated based on its weight. If the weight is bigger than $\delta_{\max}$, the placeholder is substituted with a white channel randomly chosen from $\mathcal{W}$, otherwise, with an interfered channel randomly chosen from $\mathcal{I}$.

In Fig. 3.4, an exemplary run of the whitening algorithm is shown for the scenario with $F = 4$, $\mathcal{W} = \{\mathrm{ch}_1^{\mathrm{W}}, \mathrm{ch}_2^{\mathrm{W}}\}$, $\mathcal{I} = \{\mathrm{ch}_1^{\mathrm{I}}, \mathrm{ch}_2^{\mathrm{I}}\}$, $N = 2$, $N_{\mathrm{D}} = 2$. During the Initialization phase, the sets $\mathcal{R}^i$ are created from the hopping sequence with respect to Eq. (3.3) and all the placeholders' weights are set to zero. For every white channel, the Weight phase associates a white channel to the placeholder with minimum weight, in this example to the placeholder 1. Then, it sets its weight to $\delta_1 = \delta_{\max} + 1 = 9$. Afterwards, it increments δ_3 to 2, as placeholder 3 is included in two sets containing the white placeholder 1. Finally, the placeholders in H

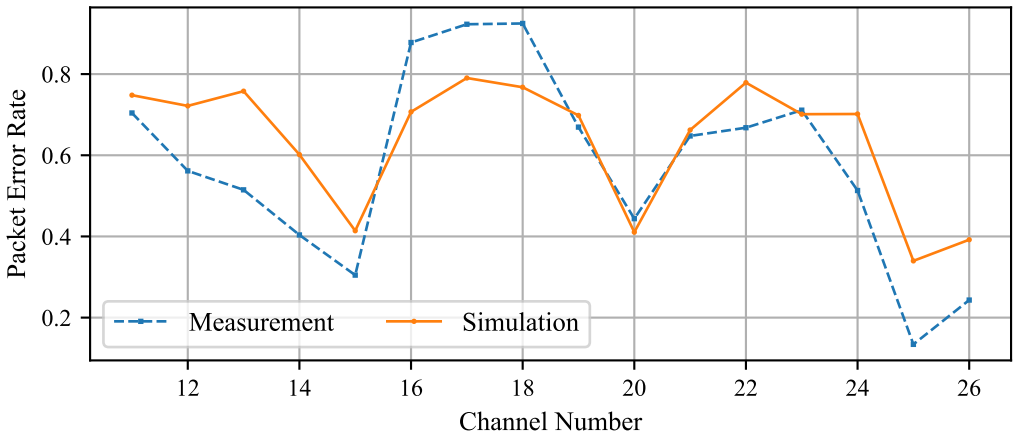

Figure 3.5: Per-channel link layer reliability of the 16 IWSN channels under WLAN interference for the simulated and measured evaluation scenarios.

with corresponding weight bigger than $\delta_{\max}$, namely $\phi = 1, 2$, are replaced with a random channel from $\mathcal{W}$, and the others ($\phi = 3, 4$) with a random channel from $\mathcal{I}$.

3.3.2 Performance Evaluation

Both simulations and measurements were performed to verify the whitening approach. On the one hand, measurements were carried out to obtain an estimation of PER values in a real environment and to verify that interfered channels can be used to improve reliability in a dense network. On the other hand, simulations were performed to compare the performances of different resource allocation methods using the same realizations of interference and data signals.

The measurements were performed in a testbed with 32 devices, 16 senders and 16 receivers, each one using one channel of the 2.4 GHz ISM band at a time, in an isolated room where only the interference of 3 WLAN Access Points (APs) was present. Each AP served one client and received uplink User Datagram Protocol (UDP) packets at maximum possible data rate. This interference corresponds to the worst-case interference scenario described in Sec. 2.3.2 and the estimation of PER of the different channels is performed using the CMA filter of Eq. (2.13). The simulations reflect the measurement scenario and their implementation follows the IWSN simulator described in Sec. 3.2.3. In Fig. 3.5, the simulated PER values are compared against the measured ones. The PER values of simulation and measurement follow similar dynamics, with channels 15, 20, 25, 26 being white, i.e., little affected by interference, and the remaining channels highly affected by it. The amplitudes of the PER values, however, deviate from the measurement for certain channels. This can be explained by multipath propagation parameters of the measurement environment being different that the

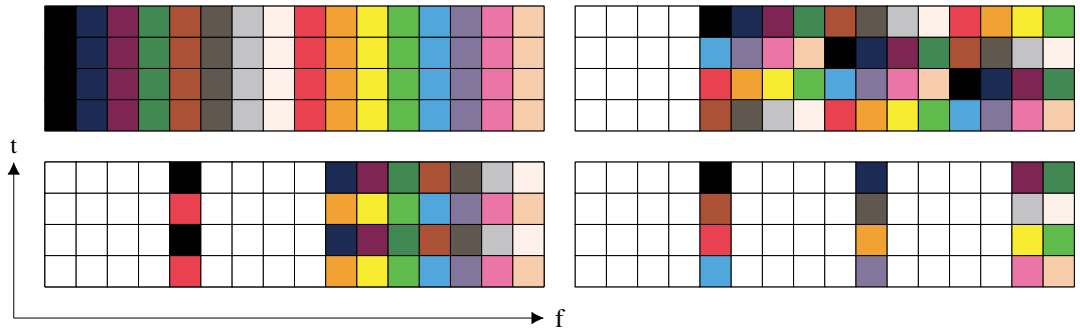

Figure 3.6: Exemplary resource allocations of the Random hopping sequence (upper-left) and the blacklist hopping sequences in the presence of 1 (upper-right), 2 (lower-left), and 3 (lower-right) interfering WLAN APs.

ones used in the model. From the simulated channel drop probabilities, we calculate α to fulfill Eq. (3.4). In particular, we select $\alpha = 1.47$ to separate the 4 white channels from the 12 interfered channels.

Different hopping sequences are used to allocate resources to the IWSN devices as represented by the scheduling grids of Fig. 3.6. Every proposed hopping sequence allocates transmissions for 16 devices, each one represented by a different color in the figure, over the 16 available frequencies within a deadline of $N_{\mathrm{D}} = 4$ time slots. Scheduling grids representing blacklists do not make use of the channels occupied by the interference. With blacklist, the number of available transmission opportunities is inversely proportional to the amount of channels occupied by interference. The upper-right scheduling grid of Fig. 3.6 shows the different hopping sequences in the presence of one interfering AP on WLAN channel 1. In this scenario, 12 channels are available over 4 slots and every device has 3 transmission opportunities to deliver an application packet. Similarly, the lower-left and lower-right scheduling grids in Fig. 3.6 show hopping sequences when 2 and 3 APs are active on WLAN channels 1,6 and 1, 6, 11. Here, due to the reduction in number of non-interfered channels, each device only has, respectively, 2 and 1 transmission opportunities to meet the application deadline. In contrast to blacklist, we define the Random hopping sequence, which is described by the upper-left scheduling grid of Fig. 3.6. This hopping sequence accommodates 4 transmissions opportunities for each device using all channels regardless of interference. The increased amount of transmission opportunities increases the reliability of the devices. However, as white channels are not fairly provisioned, the devices achieve different values of link-layer reliability. Thanks to the whitening algorithm, white channels are fairly distributed to all the devices in the network in order to maximize the reliability of all the application packets.

The measurement results in Fig. 3.7 prove that allocating transmission opportunities over interfered channels is beneficial for the total link-layer reliability. Every sender generates a

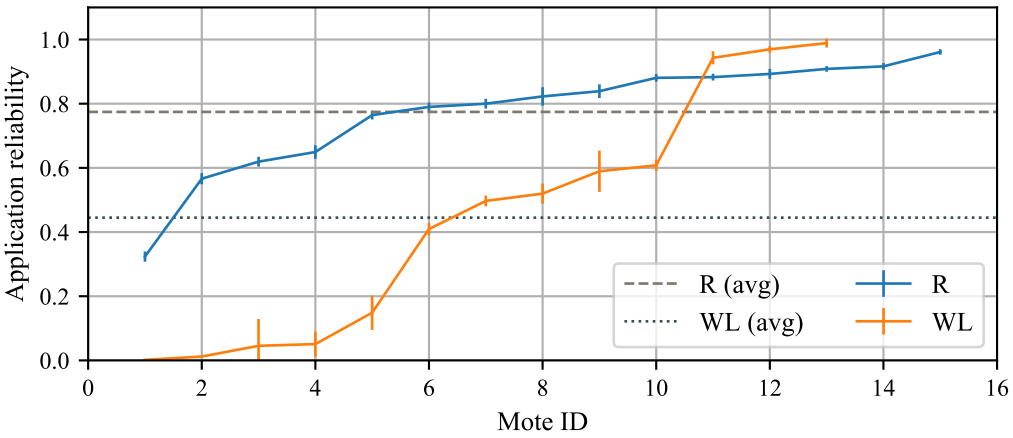

Figure 3.7: Measurements results. Sorted per-device application layer reliability, comparison of Random and blacklist hopping sequences.

burst of 11 packets with 0.5 s inter-packet interval, and 5 s interval between the bursts. In Fig. 3.7, the link-layer reliability for each device in the network is shown for two hopping sequences: blacklist with 3 APs, denoted by WL, and Random, denoted by R. The Random hopping sequence outperforms blacklist in terms of average network link-layer reliability. The result is explained by the fact that, in this scenario, blacklist allows only one transmission opportunity, whereas Random is allowing four transmission opportunities.

The evaluation of the whitening hopping sequence is performed via simulations. At the same time, the blacklist, Random, and whitening hopping sequences are simulated in the same wireless environment. For every ASN, wireless realizations of the transmission of WSN and WLAN are generated and used to compute instantaneous PER values for all the three hopping sequences. This guarantees the same performances across different hopping sequences for every transmission, thus enabling comparison. Simulations were performed for different values of TDMA frame length N, number of WLAN APs, and interference scenarios. The distance between the transmitters and receivers is fixed to 20 m, while the transmission power of the IWSN devices is set to 1 mW. The application running on each transmitter generates packets periodically, with a period that guarantees buffer stability in presence of re-transmissions. The blacklist and Random hopping sequences do not change for different N. However, whitening hopping sequences are calculated for every TDMA frame size in order to allocate the channels of the hopping sequence according to its periodicity.

Fig. 3.8 shows the behavior of the application drop-rate for different TDMA frame sizes in the presence of three WLAN APs on channels 1, 6, 11, generating interference with transmission powers equal to 10 mW (L), 100 mW (M), 1 W (H), and 10 W (U). Results show that, although the interference level is very high, accommodating re-transmissions on

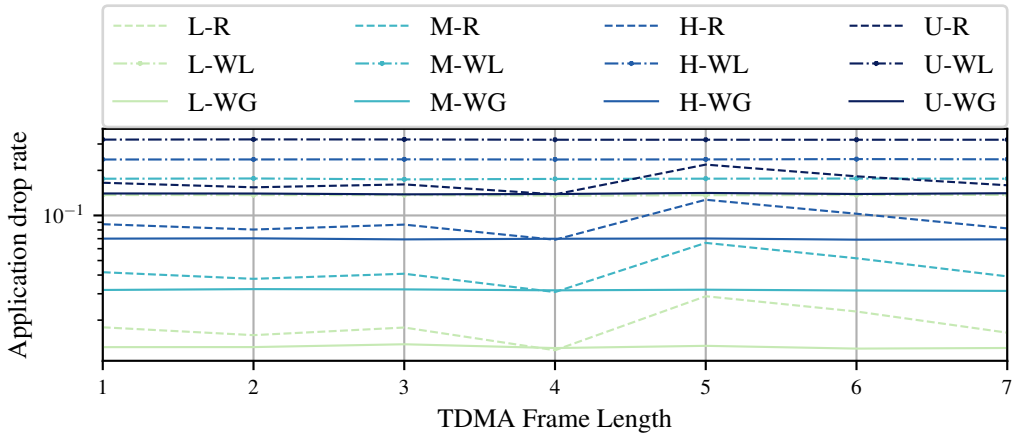

Figure 3.8: Link-layer reliability for different TDMA frame lengths N for Random (R), blacklist (WL) and whitening (WG) techniques with different levels of interference (L, M, H, U).

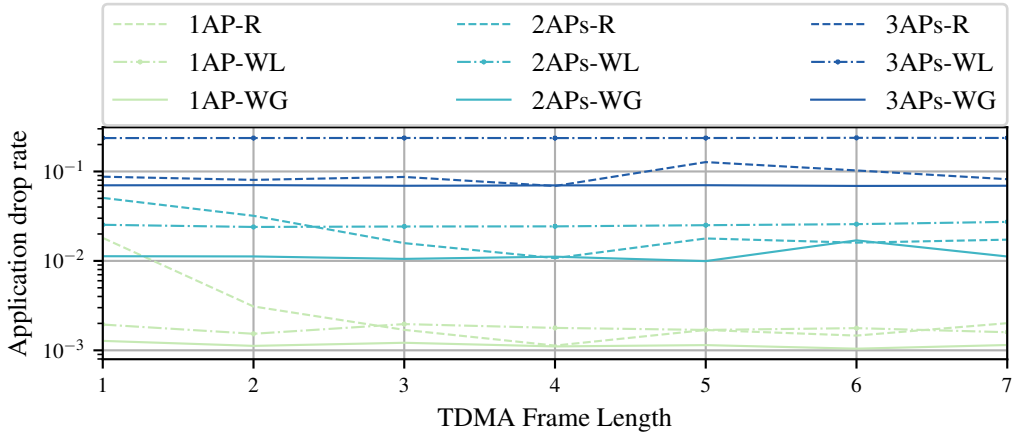

Figure 3.9: Link-layer reliability for different TDMA frame lengths N for Random (R), blacklist (WL) and whitening (WG) techniques with 1, 2, and 3 interfering WLAN APs.

interfered channels always provides a considerable reliability improvement with respect to blacklist for both Random and whitening methods. The lower the interference, the higher the performance improvement. Fig. 3.9 shows the same plots when one, two, and three APs are active on WLAN channels 1, 6, 11, with transmission power equal to 1 W. It shows how different amounts of available channels used for transmission opportunities impact the application reliability. Whitening outperforms blacklist in every scenario, while Random performs better than blacklist only with three APs. In both Fig. 3.9 and 3.8, the oscillating behaviour of the Random hopping sequence is explained by the synchronization of white channels in the transmission opportunities for different N. For instance, for $N = 4$ the Random hopping sequence is a whitening hopping sequence, and whitening and Random achieve the same results. However, as shown in Fig. 3.9 for $N = 5$, an initial channel

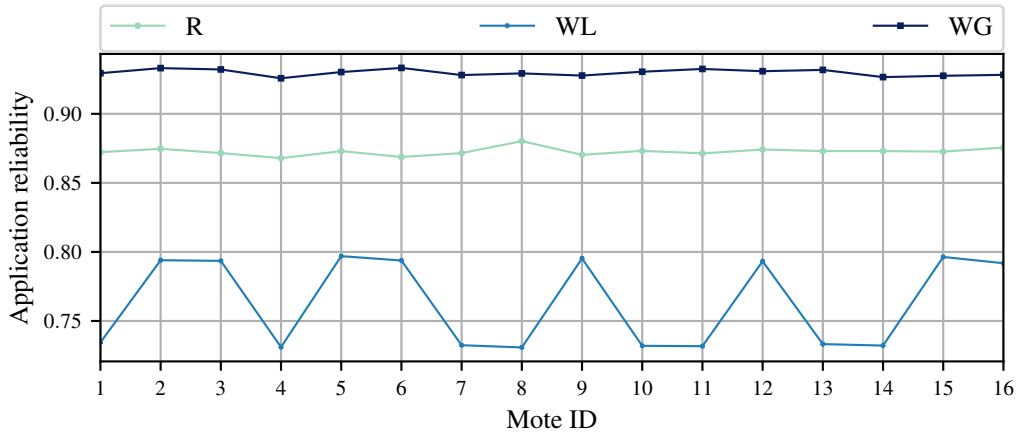

Figure 3.10: Per-device application reliability for $N = 5$.

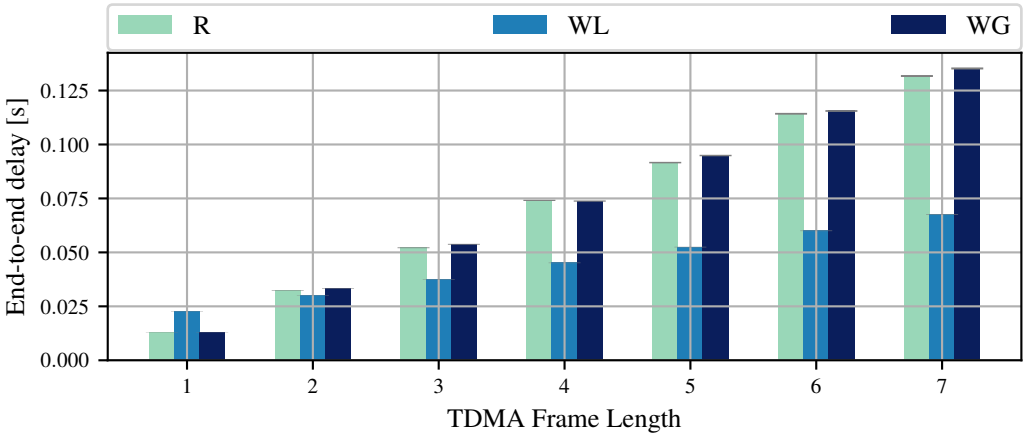

Figure 3.11: Latency achieved by the network for different TDMA frame lengths.

selection can allocate three consecutive white channels to some devices and consecutive interfered channels to others. Both blacklist and Random are penalized by the way that the hopping sequence is indexed according to the TDMA frame size. However, this effect is not present in whitening that optimized the hopping sequence designed for every TDMA frame length N. This confirms the importance of computing a whitening hopping sequence based on the network configuration, interference scenario, and application requirements.

Fig. 3.10 shows the link-layer reliability achieved by different devices with $N = 5$. The interference power is equal to 1 W and is generated by 3 APs. Blacklist is outperformed by Random and whitening, and, due to the reduced number of channels and transmission opportunities, different link-layer reliabilities are achieved by the devices. Furthermore, Whitening additionally outperforms Random, which has reduced performances due to the periodicity of the TDMA frame as shown in Fig. 3.9.

Fig. 3.11 shows the link-layer latency achieved by all the devices for odd values of N under same interference conditions. Blacklist has the latency advantage of using a single reliable transmission to achieve the application reliability, while, for Random and whitening, re-transmissions are needed and increase the total latency. However, for $N = 1$, the latency achieved by blacklist is higher than Random and whitening due to the longer waiting times between transmission opportunities. The results shown in Fig. 3.11 confirm the delay-reliability trade-off shown in [57], that is, increased reliability can be achieved by re-transmissions at the price of increased latency.

3.4 QoS Provisioning Under Dynamically Interfered Channels

The previous section shows that link-layer QoS can be provisioned in the presence of static interference. While this approach is valid for some scenarios, it does not cover the general problem of dealing with time-varying, unpredictable interference. In fact, in an industrial environment, the link budget of wireless devices can vary over time. This can be caused, for instance, by the movement of machines and robots or by interference from coexisting neighbouring nodes. To deal with this unpredictable scenario, information about the physical-layer reliability does not only have to be collected at design-time, but has to be continuously updated during the operation of the network. Furthermore, upon critical changes in the channels' reliability, new resource allocations must be computed and communicated to the devices in order to fulfil the industrial QoS requirements. This can be efficiently performed in IWSN thanks to a centralized star topology that allows the NC to react to changes in the channels' reliability via dynamic scheduling. *Dynamic scheduling*, represented in Fig. 3.2 and described in Sec. 3.2.2, is a periodic operation that consists in *(i)* sending new link quality information to the scheduler, and *(ii)* sending up-to-date schedules from the scheduler to the devices.

In this section, a dynamic IWSN scheduling procedure is described in Sec. 3.4.1 that enables link-layer QoS provisioning in the presence of time-varying interference. Similarly to the RRM algorithm described in Sec. 3.3, the goal of the scheduler is to fulfil a desired link-layer reliability within a given deadline. Furthermore, in Sec. 3.4.2, the performance of the dynamic scheduler is evaluated via simulations of an IWSN in the presence of dynamic interference. The simulation scenario replicates the IWSN deployed for measurements and simulations of an industrial environment used in Ch. 2 and Sec. 3.3, and extends the worst-case WLAN interference scenario introducing time-varying effects in the number of WLAN APs and in their transmission power.

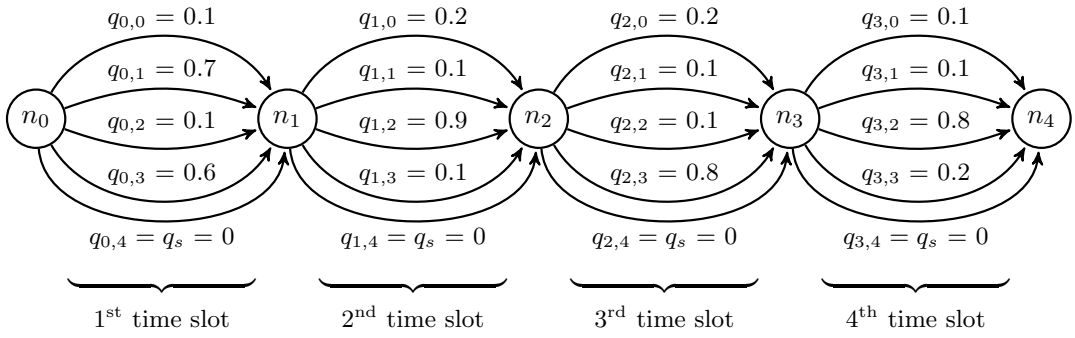

Figure 3.12: Example of a *scheduling graph* used by the reliability-based dynamic scheduler in order to allocate RBs to the IWSN devices.

3.4.1 Reliability-Based Dynamic Scheduler

The proposed scheduler is based on a graph model, the *scheduling graph* depicted in Fig. 3.12. The scheduling graph and the scheduling procedure are unique for each transmitter-receiver link $\{\ell, m\}$. For this reason, in this section, the link indexing is omitted to improve readability. Nodes in the scheduling graph represent time instants between the different time slots. Edges represent the transmission at different frequencies and in different time slots. For each time slot, one additional edge, the *silent edge*, represents no transmission. In the example of Fig. 3.12, for each time slot, the four upper edges represent transmission at four different frequencies and the lower edge corresponds to the silent edge. That is, every possible RB is represented by an edge in the scheduling graph. The PDR of the RBs is obtained from the link quality statistics gathered by the NC and represent the weights of the different edges. As silent edges never lead to a successful transmission, their PDR is defined as $q_s = 0$. As a result, path finding on this graph from the first to the last node defines a schedule.

The reliability-based scheduler operates by finding a path in the scheduling graph such that the reliability of the found path, i.e., the product of the PERs of the chosen edges, satisfies the target reliability requirement of the application r_0, that is

$$\prod_{(t,f)\in\mathcal{S}} 1 - q_{t,f} < 1 - r_0, \tag{3.5}$$

where $\mathcal{S}$ is the set of edges constituting the found path, i.e., the schedule. The QoS requirements of industrial applications are defined in terms of link-layer delay and reliability. Therefore, in order for the scheduler to additionally take the delay bound into account, the number of time slots in the scheduling graph cannot exceed the deadline. In this way, each feasible schedule found by the scheduler fulfils both delay and reliability requirements.

The PER metric is multiplicative, i.e., the metric values of the individual edges of a path have to be multiplied in order to obtain the end-to-end metric value of the path. However,

most state-of-the-art routing algorithms can only deal with additive metrics [54], i.e., with metrics whose end-to-end value corresponds to the sum of their values on the individual edges. As a result, transforming the unavailability metric into an additive metric allows using state-of-the-art algorithms. To do so, the logarithm operator is applied to Eq. (3.5). This successively yields[1]

$$\log\left(\prod_{(t,f)\in\mathcal{S}} 1-q_{t,f}\right) < \log(1-r_0),$$
$$\sum_{(t,f)\in\mathcal{S}} \log\left(1-q_{t,f}\right) < \log(1-r_0). \tag{3.6}$$

There can be more than one path in the scheduling graph satisfying Eq. (3.6), i.e., there are potentially more than one schedule that allocate enough resources for communication in order to satisfy its target reliability value. In order to define a preference order among the schedules, we define a cost $c_{t,f}$ for each edge in the scheduling graph. The cost function $c_{t,f}$ can be defined arbitrarily and, in general, should be defined in a way that leads to the largest amount of schedulable applications. In our case, we aim at minimizing the amount of allocated RBs and define $c_s = 0$ and $c_{t,f} = 1, \forall (t,f) \in \mathcal{S}$.

The routing problem for finding a schedule then consists in solving

$$\min_{\mathcal{S}\in\mathcal{P}} \sum_{(t,f)\in\mathcal{S}} c_{t,f} \tag{3.7}$$
$$\text{s.t.} \sum_{(t,f)\in\mathcal{S}} \log\left(1-q_{t,f}\right) < \log(1-r_0), \tag{3.8}$$

where $\mathcal{P}$ is the set of all the paths from the first node to the last node of the scheduling graph. The problem described by Eqs. (3.7) and (3.8) corresponds to a Constrained Shortest Path (CSP) problem.

A plethora of algorithms have been developed to solve CSP problems [54]. However, most of these algorithms assume that the constrained metric is positive and construct the result path hop-by-hop, thereby starting with feasible paths, i.e., paths satisfying Eq. (3.7), and by avoiding reaching infeasible paths, i.e., paths not satisfying Eq. (3.8). However, as $q_{t,f} < 1$, our constrained metric is negative. As such, path construction starts from infeasible paths and has to end in the feasible zone. Hop-by-hop algorithms are not straightforward to adapt to this situation. The *LARAC* algorithm [14, 20, 59, 68], one of the best performing CSP algorithms [54], does not construct its result path hop-by-hop. Instead, it computes subsequent paths using Dijkstra with an aggregate cost metric and labels the intermediate

[1] It is assumed that $\gamma_{t,f} < 1$, which allows the usage of the log operator.

paths found as either infeasible or feasible. Thanks to this property and to Dijkstra's ability to handle loop-less graphs with negative metrics, the LARAC algorithm can be directly used to solve problem (3.7)-(3.8) and is employed by the dynamic scheduler.

The dynamic scheduling procedure, represented in Fig. 3.2, operates as follows. At the beginning of each TDMA frame, the NC computes schedules based on the current PER estimations and the link-layer QoS requirements. The schedules are distributed to the devices using the EB, cf. Sec. 3.2.1. It can happen, however, that the EB is not correctly received by all the devices. In this case, the devices temporarily silence their radios, thus avoiding potential interference in the network, and wait for the next EB. As control messages can be transmitted using robust radio resources, for instance using non-interfered frequencies, it is assumed that no EB is lost. For every RB allocated by the schedule, a device transmits either a data packet, if available, or a substitute probing packet. In this way, information about the physical-layer reliability of the scheduled communication channels can be inferred by the NC at each time slot. At the end of every frame, the NC uses the newly estimated PERs to compute new schedules that fulfilthe QoS requirements.

The dynamic estimation of PER is performed using the EWMA software-based estimator described in Sec. 2.3.2. For each RB in schedule $\mathcal{S}$, the NC estimates its PDR $q_{i,j}$ using Eq. (2.15). Furthermore, in order to avoid that unscheduled frequencies are accidentally discarded by the scheduler due to temporarily bad performance, an ageing procedure is performed. Ageing consists in artificially improving the PDR of unscheduled frequencies such that they can be explored again by the scheduler. The ageing procedure is implemented using another EWMA filter as follows. At each time slot, the PDR of the unscheduled RBs is artificially increased by

$$q_{i+1,j} = \beta + (1-\beta)\, q_{i,j}\,, \forall\,(i,j) \notin \mathcal{S}. \tag{3.9}$$

The value of β must be chosen carefully, as it introduces a reliability penalty that occurs if channels with low PDR are used. In general, it should be set according to the interference behaviour and the scheduling procedure.

3.4.2 Performance Evaluation

The reliability-based dynamic scheduler is evaluated via simulations in different interference scenarios. The IWSN consists of 16 devices periodically transmitting sensor readings to a remote NC, thereby generating uplink traffic. The NC acts as a sink. It receives the whole traffic and forwards it to a real-time monitoring application, which is employed, for instance, for the timely analysis of one or multiple industrial processes. To fulfil this role, the NC is directly connected to the industrial wired network and is equipped with improved

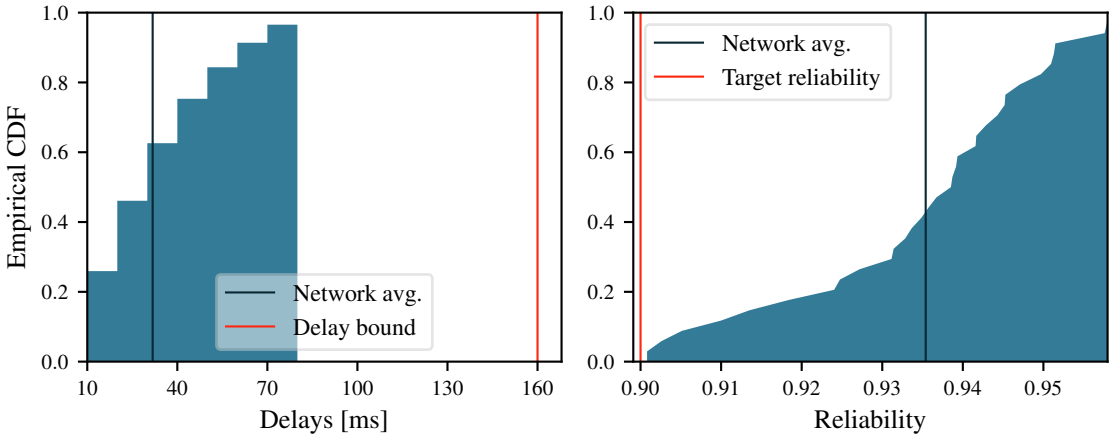

Figure 3.13: Empirical CDF of the link-layer delay and reliability achieved by the devices with dynamic scheduling and perfect Link Quality Information (LQI), 0.90 target reliability, 34 supported devices.

hardware, which enables it to simultaneously receive or transmit over all the 16 IEEE 802.15.4 frequencies of the 2.4 GHz ISM band without processing and energy limitations. The IWSN devices deployed in the factory are equipped with constrained hardware and transmit over a single frequency at each time slot.

Simulations with the availability of perfect PER estimation are performed to verify the correctness of the reliability-based scheduler and test its performance independently from the EWMA filter. Fig. 3.13 and 3.14 show the empirical Cumulative Distribution Function (CDF) of the link-layer delay and reliability achieved by the industrial messages of all the devices supported by the scheduler during the entire simulation. The results show that the reliability-based scheduler is able to fulfil target link-layer reliabilities of 0.9 and 0.99, and a latency bound of 160 ms by achieving a maximum delay of 80 ms when the channel quality information is known. The exemplary schedule $\{(0,9),(2,11),(6,11)\}$ supports the target reliability of 0.9. For the same application packet, it allocates 3 transmissions opportunities in the TDMA frame at time instants $0,2,6$ over the frequencies $9,11$ with corresponding PDR equal to $0.68,0.9$. The allocation provides an application reliability of 0.987, which fulfils the requirement of 0.9.

A static interference scenario is used to evaluate the impact of different EWMA parameters α on the performance of the dynamic scheduler. Tab. 3.1 summarizes the results of different values of α for an experiment duration of 16.7 minutes. The static interference consists in three WLAN APs on WLAN channels 1, 6, 11 using a transmission power of -10 dBm. On the one hand, for low values of α, the filter includes more history in the estimation, slowly adapting to the changes. On the other hand, for high values of α, the filter is more reactive.

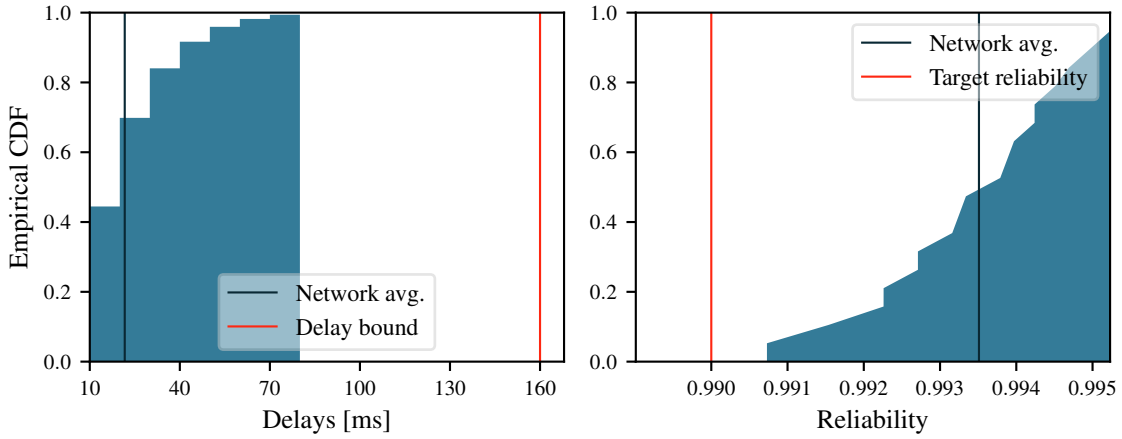

Figure 3.14: Empirical CDF of the link-layer delay and reliability achieved by the devices with dynamic scheduling and perfect LQI, 0.99 target reliability, 19 supported devices.

α	Network reliability: avg. / target (% devices)	Network delay: avg.	N. supported devices
0.001	0.872 / 0.9 (12.5)	35.38	40
0.01	0.919 / 0.9 (97)	32.78	34
0.02	0.920 / 0.9 (94)	32.89	35
0.03	**0.924 / 0.9 (100)**	**32.70**	**35**
0.04	0.921 / 0.9 (97)	32.75	34
0.05	0.920 / 0.9 (94)	32.80	35
0.06	0.915 / 0.9 (89)	33.24	35
0.1	0.891 / 0.9 (74)	33.77	34

Table 3.1: Evaluation of the EWMA parameters α for dynamic scheduling under static interference and aging factor $\beta = 10^{-6}$. The avg. link-layer network reliability is compared to the target reliability of 0.9. "%" indicates the percentage of devices achieving the target reliability.

The extremes of this dynamic can be observed in Tab. 3.1. When α is equal to 0.001 and 0.1, the impact of erroneous estimation is crucial and the scheduler cannot meet the average target reliability. Moreover, for the remaining values of α, although the average target reliability is achieved by the network, only some devices (%) achieve the target reliability. For the specific value $\alpha = 0.03$, both the network and per-device average reliability targets are achieved.

Fig. 3.15 and 3.16 show the detailed network performance for $\alpha = 0.03$ in the static interference scenario. Fig. 3.15 shows the time evolution of the average network PDR and the number of supported devices. After an initial transient, the EWMA filters of the network channels converge to the real PER values and the scheduler adjusts the number of admitted devices. At the end of the simulation, the average network PDR is equal to 0.56 and the

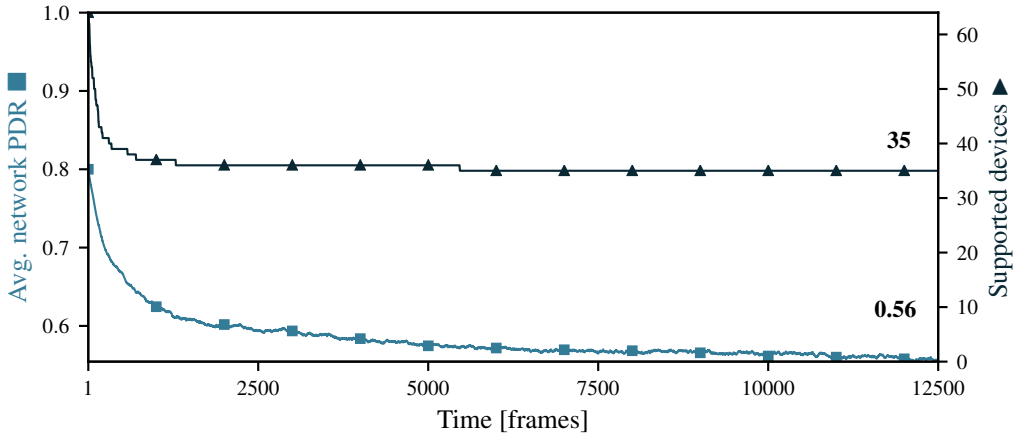

Figure 3.15: Time evolution of the number of supported devices and avg. physical-layer network reliability in the presence of static interference, $\alpha = 0.03$, $\beta = 10^{-6}$, 35 supported devices.

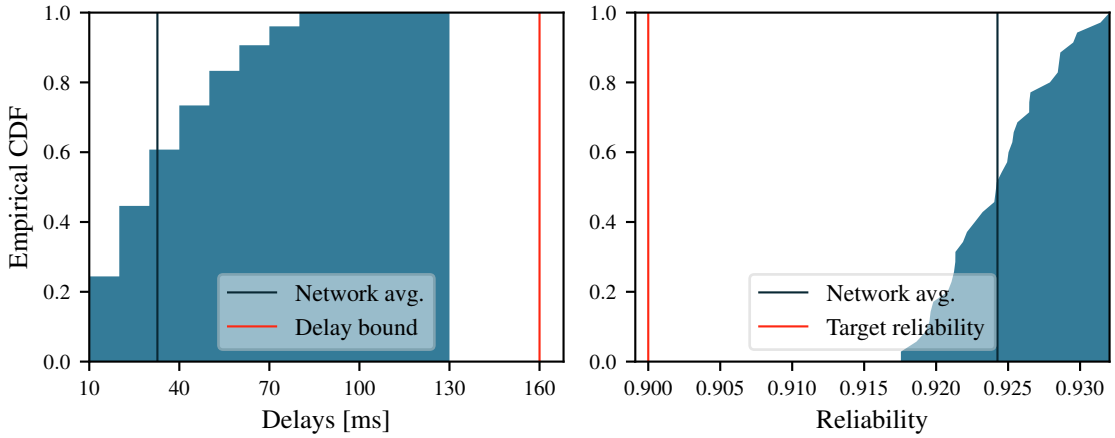

Figure 3.16: Empirical CDF of the link-layer delay and reliability achieved by the devices with dynamic scheduling and static interference, $\alpha = 0.03$, $\beta = 10^{-6}$, 35 supported devices.

scheduler is able to support 35 devices. Fig. 3.16 shows the empirical CDF of the delay and reliability performances of all the devices. The results confirm that the link-layer QoS requirements are met by all communication links. In this case, the combination between α and the interference scenario yields to a physical-layer estimation that allows the scheduler to fulfil the QoS requirements.

Finally, the dynamic scheduler is evaluated with simulations in the presence of dynamic interference. Fig. 3.17 and 3.18 show the network performance for an interference arising from three WLAN APs on WLAN channels 1, 6, 11 that increase the transmission power in three steps from -20 dBm to 10 dBm, which are represented in Fig. 3.17 by vertical dashed lines. Results are shown for $\alpha = 0.03$, a delay bound of 160 ms, and a target reliability of 0.9.

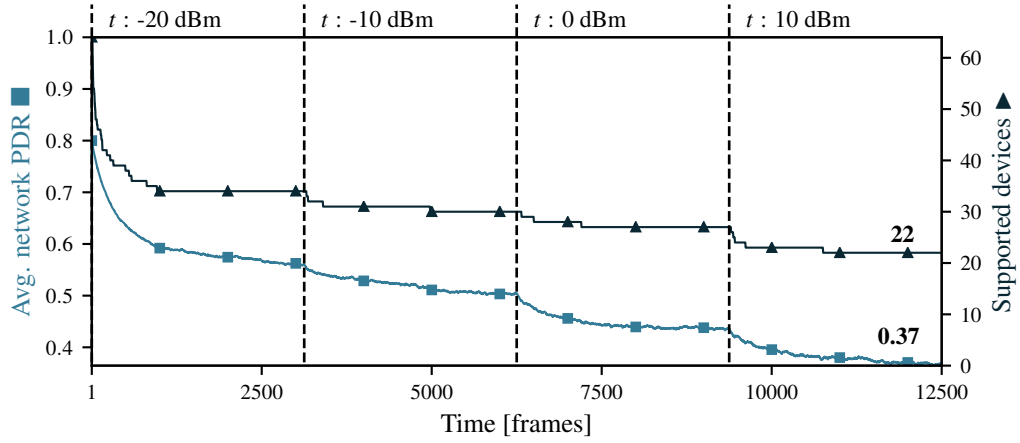

Figure 3.17: Time evolution of the number of supported devices and avg. physical-layer network reliability in the presence of increasing WLAN transmission power t, $\alpha = 0.03$, $\beta = 10^{-6}$, 22 supported devices.

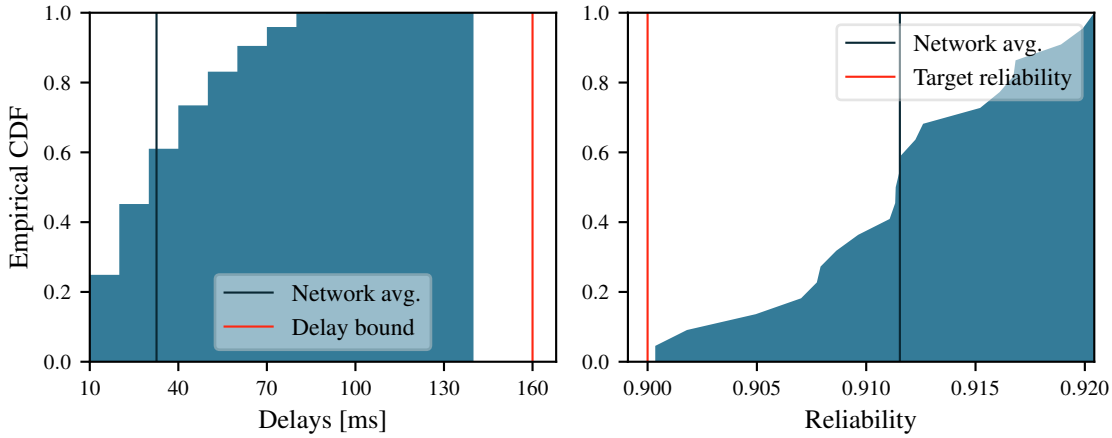

Figure 3.18: Empirical CDF of the link-layer delay and reliability achieved by the devices with dynamic scheduling and increasing WLAN transmission power t, $\alpha = 0.03$, $\beta = 10^{-6}$, 22 supported devices.

Results prove that the dynamic scheduler is able to detect interference changes and adapt the schedules in order to fulfil the required target reliability and delay bound. Due to the higher interference, at the end of the simulation, the average network PDR is equal to 0.37 and the scheduler supports 22 devices.

Fig. 3.19 and 3.20 show the network performances for a second dynamic interference scenario where the number of WLAN APs increases from 1 to 3 over the channels 1, 6, 11 in two steps, which are represented in Fig. 3.19 by vertical dashed lines. Results are shown for $\alpha = 0.02$, a delay bound of 160 ms, and a target reliability of 0.99. Also in this case the QoS

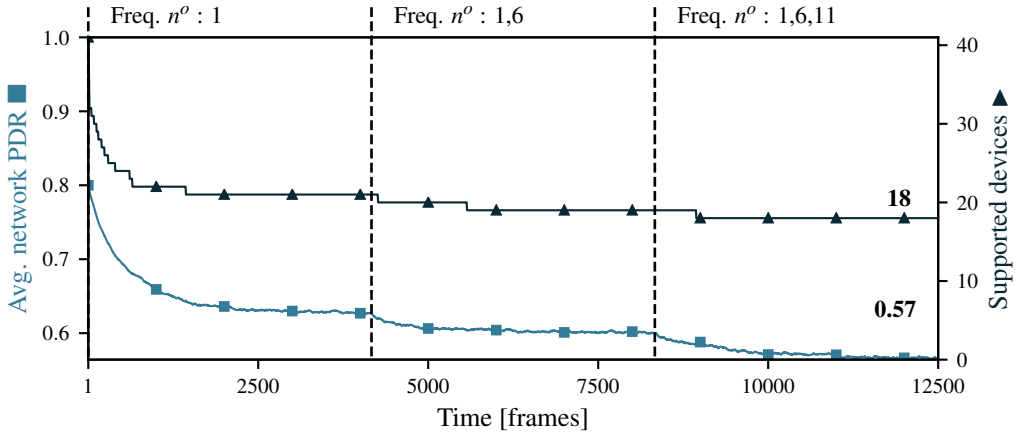

Figure 3.19: Time evolution of the number of supported devices and avg. physical-layer network reliability in the presence of increasing number of interfering WLAN APs, $\alpha = 0.02$, $\beta = 10^{-6}$, 18 supported devices.

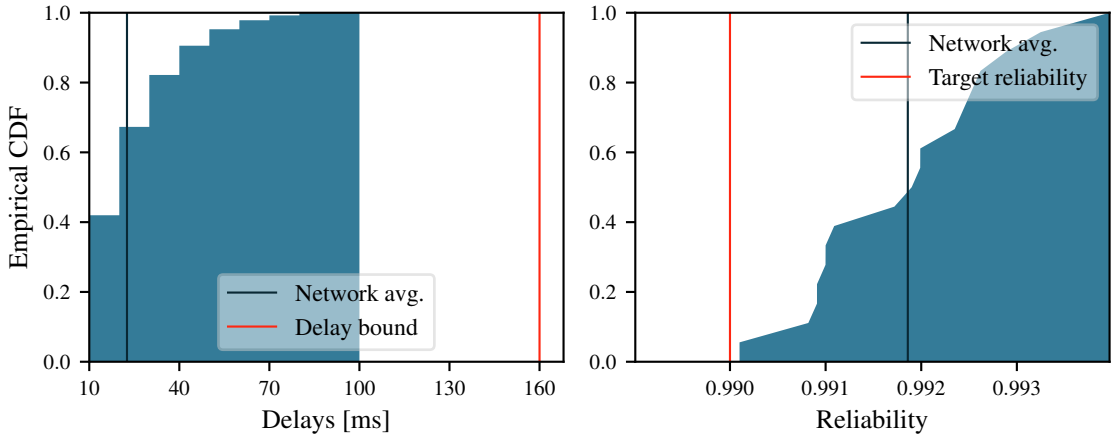

Figure 3.20: Empirical CDF of the link-layer delay and reliability achieved by the devices with dynamic scheduling and increasing number of WLAN APs, $\alpha = 0.02$, $\beta = 10^{-6}$, 18 supported devices.

requirements are fulfilled for the entire IWSN. However, due to more stringent application requirements, the number of supported devices at the end of the simulation is 18.

3.5 Concluding Remarks

This chapter presented link-layer RRM schemes to provide link-layer QoS in the presence of static and dynamic interference. The RRM schemes rely on a stochastic model of the link-layer resources of IWSN, which is used to determine the QoS performance of schedules. To perform RRM in a static interference scenario, the whitening hopping sequence design was presented. Whitening uses the physical-layer reliability of all IWSN channels to allocate

channels in order to provision reliable consecutive re-transmissions to each device. In this way, it exploits interfered channels to improve the link-layer reliability compared to existing state-of-the-art blacklisting methods that discard them.

Furthermore, a reliability-based dynamic scheduler was presented. The scheduler makes use of the PER estimation of an EWMA filter to compute schedules that fulfil link-layer QoS requirements of industrial applications. In this way, the reliability-based scheduler is able to provide link-layer QoS in the presence of dynamic interference. This is a fundamental achievement, as existing RRM schemes for QoS provisioning in IWSN discard interfered channels or do not consider the effect of external interference. Thanks to these results, the RRM schemes presented in this chapter allow lossy IWSN links to counteract the effect of packet loss and achieve desired link-layer QoS performances. As presented in Chapters 4 and 5, this is particularly important, as it allows deriving network-layer RRM schemes that support the multi-hop communication requirements of Networked Control Systems (NCS) and to characterize their performance.

Chapter 4

Network-layer QoS Provisioning in a Two-Hop IWSN

The Radio Resource Management (RRM) schemes presented in the previous chapter enable the devices in the network to achieve desired link-layer Quality of Service (QoS) performance in the presence of time-varying interference. In this way, the problem of packet loss arising from unreliable links is overcome and the transmissions of sensors, controllers, and actuators can achieve desired link-layer reliability and delay performances. This achievement, however, is not sufficient to ensure the correct operation of an entire feedback system. In fact, differently from traditional industrial applications, feedback systems do not only require timely and reliable communication links between sensor, controller, and actuator, but strict QoS performances for the entire operation of their control loop.

When steering a physical system, the delay between the generation of sensor measurements and the reception of actuation commands is vital for the application. Depending on its duration, an actuator can successfully react to a sudden event captured by the sensor or fail its task, thus impacting the performance of the entire system. Therefore, for each control loop, it is important that *each packet* generated by the sensor, processed by the controller, and received by the actuator experiences a maximum delay that is smaller than a given *deadline*. Due to the random nature of the wireless transmissions, a deterministic characterization of the network delays is not possible. For this reason, the network-layer QoS metrics considered in this work quantify the delay d^{L3} and the Delay Violation Probability (DVP) r^{L3} of the packets of each control loop.

Given a specific deadline from the control system, it is the responsibility of the Network Coordinator (NC) to ensure that the available network resources are optimally coordinated to minimize the DVP of packets in each control loop. This is a challenging task as the DVP depends on fluctuating network parameters, such as backlogs in the queues at the beginning

of the loop and the temporal variability achieved by the links with specific link-layer delay and reliability performances. Optimizing the network resources in this scenario corresponds to adapt the schedules of the devices to the transient behaviour of the network state, i.e., to the evolution of the backlogs at the transmitters of sensor and controller [27]. As described in details in Sec. 4.1, several works tackle the problem of resource allocation in a multi-hop wireless network to support time-critical applications. Existing methods, however, cannot be applied to feedback-based applications served by Industrial Wireless Sensor Networks (IWSN) as they either focus on the steady-state performance of the application packets [37, 46, 62, 122], allocate a maximum number of re-transmissions [22, 30, 51], do not optimize the network resources to minimize DVP, and do not model the correlation between resource allocations of subsequent transmitters [123].

This chapter presents a solution to these issues by deriving, in Sec. 4.2, the DVP of individual packets of a control loop and presenting, in Sec. 4.3, different scheduling policies to minimize their DVP. The results are based on the own publication "Dynamic Scheduling for Delay-Critical Packets in a Networked Control System Using WirelessHART" [5] and the own article "Scheduling of Two-Hop Lossy Wireless Networks for Time-Critical Feedback Systems" [1]. In particular, as described in Sec. 4.2, I contributed to the works by deriving the analytical expression of DVP for a two-hop IWSN and approximating it with two upper bounds. Furthermore, I have developed several heuristic static scheduling algorithms, which are presented in Sec. 4.3.1, and collaborated with Mr. Champati in the development of a heuristic dynamic scheduling algorithm presented in Sec. 4.3.2. Finally, I have developed an IWSN network-layer simulator and, as shown in Sec. 4.4, performed extensive simulations to evaluate the performance of the schedulers and compare them with state-of-the-art algorithms. This chapter is concluded by Sec. 4.5, which summarizes the contributions and discusses their implications within this thesis.

4.1 Related Work

Several existing works tackle the problem of resource allocation in a multi-hop wireless network to support time-critical applications. Methods that make use of the queue state to allocate network resources follow a *theoretical* approach [122, 133, 134]. In their pioneering work [133], Tassiulas et al. derived the Max Weight scheduling policy, which allocates resources based on the transmitters' backlogs and achieves maximum throughput and minimum delay. Their scenario, however, is different from the one in this work as they only considered a single-hop network. For a multi-hop network, maximum throughput was achieved by the backpressure algorithm [134], which allocates resources based on the backpressure of queues

in the network. Differently from this work that minimizes the DVP of time-critical packets, their scheduling policy focused on throughput optimality. Singh et al. [122] exploited information about the queue state to schedule transmissions in order to maximize throughput under delay constraints. Their approach, however, considers the steady-state performance of packets and, differently from the approach described in this chapter, does not allocate resources to optimize the network for a single time-critical arrival.

From a different perspective, many practical works investigate resource allocation methods for real-time flows in IWSN. Some of them compute schedules to allow several time-constrained applications to meet their deadlines assuming *deterministic transmission outcomes* [86, 115–118, 141]. Saifullah et al. [116, 118] investigated the problem of real-time scheduling subject to end-to-end deadlines between sensors and actuators. Differently from this work, however, communication between sensors and actuators is deterministic and packet loss is not considered. In their recent works [115, 117], communication failures due to packet loss are considered and retransmissions are used. In contrast to the approach of this chapter, however, they only provided a deterministic delay model for the communication between sensors, controller, and actuators. Similarly, Wang et al. [141] calculated schedules to ensure the worst-case delay of packets in a flow, however, without assuming random packet loss. Modekurthy et al. [86] derived a distributed deadline-based scheduling algorithm based on the Earliest Deadline First policy. Also in this case, differently from this chapter, packet loss is not considered and the random end-to-end delay of packets in the network is not characterized.

Other IWSN works tackle the problem of reliable communication in the presence of *random packet loss* [37, 46, 62]. Dobslaw et al. [37], allocated resources to each transmitter in a path based on the required number of retransmissions to fulfil a given reliability constraint. Differently from the proposed model, however, their work did not consider a deadline for the packets. Following a similar approach, Gaillard et al. [46] extended the pioneer traffic-aware centralized scheduler TASA [96] including re-transmissions to guarantee flow reliability requirements. Also in this case, however, traffic is not time-constrained. Yan et al. [62] developed a scheduling method that allocates time slots to the transmitters in order to maximize the network reliability under delay constraints. A major difference of their work, which is common to Dobslaw and Gaillard et al., is that reliability constraints are defined for all the flows in the network. Differently, the scheduling policies presented in this chapter optimize the network resources to maximize the application reliability of each time-critical packet.

Recent works tackle investigate resource allocation methods providing *per-packet* delay and reliability performance [22, 30, 51, 123]. Similarly to DVP, Chen et al. [30] computed, for each packet, the number of transmissions required to fulfil the application deadline with a given probability. Their work, however, only considered a single-hop scenario and cannot

be applied to the considered two-hop network path. Brummet et al. [22] developed a method to dynamically allocate retransmissions to each packet and at each network hop subject to delay and reliability requirements. They followed a different approach as their schedules are designed to fulfil the requirements and limit the maximum number of retransmissions. Therefore, their scenario did not optimize network resources to maximize the per-packet DVP. A similar approach was used by Gong et al. [51], which allocated time slots to transmitters fulfilling per-packet delay and reliability constraints while minimizing the number of resources. Also in this case, however, they considered a finite number of retransmissions and the network resources are not optimized to minimize DVP. On the contrary, Soldati et al. [123] allocated time slots over multiple hops maximizing the end-to-end reliability of each time-critical packet subject to a deadline. Differently from the approach of this chapter that characterizes the distribution of end-to-end delay considering the correlation of transmission outcomes over consecutive transmitters, their scenario assumes that resource allocations of consequent transmitters are independent.

In this chapter, different scheduling policies are presented in Sec. 4.3 based on the own works "Dynamic Scheduling for Delay-Critical Packets in a Networked Control System Using WirelessHART" [5] and "Scheduling of Two-Hop Lossy Wireless Networks for Time-Critical Feedback Systems" [1], which investigate the impact of network state information on the minimum achievable DVP. This is achieved thanks to a transient queuing model of the network and by modelling the end-to-end delay of each packet of feedback-based applications as described in Sec. 4.2. This is different from the available state-of-the-art for multiple reasons. Instead of considering network-layer QoS for a stationary flow, the stochastic delay incurred for each time-critical arrival is analysed. The considered queuing model allows deriving scheduling policies that consider the correlation between subsequent transmitters introduced by stochastic wireless transmissions. This is different from the related work as existing scheduling policies optimize network resources based on the interaction of multiple independent flows sharing the network. Furthermore, by allocating a finite amount of retransmissions, all the existing methods allow application packets to be dropped, which may result in critical failures in feedback systems. Finally, the impact of queue state information on static and dynamic scheduling policies is investigated, which was never tackled in the literature for a two-hop network path.

4.2 Transient Network Model and Delay Violation Probability

The main task of the network layer is to establish logical communication links between all the devices in the network. In this way, it enables communication between nodes located in separate radio ranges. Additionally, the network layer is responsible for the scalability of the network, dealing with problems such as the addressing of the devices and the definition of the network topology.

In traditional Wireless Sensor Networks (WSN) deployments, hundreds or thousands of nodes are deployed in a geographical area to measure physical parameters, such as temperature and brightness, and transmit them to a remote sink for data collection and monitoring. In this paradigm, the main task of the networking layer is to define routes, i.e., a sequence of links in the network, to enable the collection of data from the devices to the sink. The selection of a specific route between a given source and destination influences the energy consumption of the network and the QoS performance of the packets. For this reason, the design of routing algorithms for WSN applications has been extensively studied in the literature [11, 69] with the main goal of reducing the energy consumption.

IWSN represents a paradigm shift for WSN, where QoS performances, such as low latency and reliability, have more priority than energy efficiency [76, 109, 160]. These requirements pose additional constraints to the network topology and routing strategy that are reflected by all IWSN standards [15, 124, 158]. When low latency operation is required, the depth of the network topology and the amount devices is limited. Enhanced reliability is achieved via multipath diversity, that is, redundant routes are established to exploit the spatial diversity of the network. Furthermore, in order to fulfill the QoS requirements of industrial applications, the allocation of link- and network-layer resources is centrally supervised by the NC. Relevant state-of-the-art routing techniques that tackle QoS provisioning in IWSN are REALFLOW [153], MSPEED [45], and RPL [13]. In order to satisfy QoS requirements over multiple hops, a common strategy, which is adopted by the network layer considered in this chapter, is to allocate resources at the link-layer in order to fulfill the QoS requirements of multiple links. One example of such mechanism is present in ISA 100.11a [15], where the routing mechanism is performed at the data link layer in each sub-network.

In this chapter, the task of the network layer is to deliver the packets of a control loop over a two-hop sensor-to-controller and controller-to-actuator network path. The IWSN operates in an indoor industrial environment, where extensive network infrastructure provides ubiquitous connectivity to the NC. In this way, as represented in Fig. 4.1, sensors and actuators are in the same radio range of the NC, which is directly connected to the control logic. For this

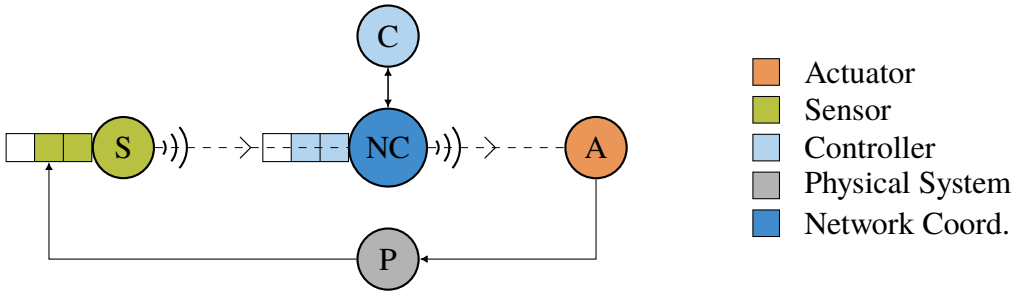

Figure 4.1: Network-layer architecture of the feedback-based communication of an Networked Control Systems (NCS).

reason, the network follows a star topology and a routing procedure is not necessary. The task of the NC is to allocate the available radio resources to the two-hop network path in order to minimize the DVP of each instance of the control loop. This is achieved by the scheduling policies described in Sec. 4.3 thanks to a transient model of the network queues that is presented in Sec. 4.2.1 The model describes the packet arrival of each control loop and the stochastic service provided by the IWSN devices. Finally, in Sec. 4.2.2, the model is used to derive the analytical DVP of packets traversing the two-hop network path.

4.2.1 Two-Hop Network Model

The sensor-to-controller and controller-to-actuator links are modelled using a packet-flow, discrete-time, two-queue lossy wireless network with first-come-first-serve discipline as shown in Fig. 4.2. The time is discretized into slots, which are grouped in frames. A sequence of y packets arrives at the first queue in frame 0[1]. These packets are time-critical with a requirement that they depart the second queue within next w frames, where w is finite. These packets, for instance, may belong to a time-critical sensor measurement whose latency could significantly impact the control performance, i.e., the stability/safety of the controlled system. It is thus interesting to analyse the two-queue network for the time frames $k \in \{0, 1, \ldots, w-1\}$. In this transient regime, the delay incurred by the time-critical packets depends on the initial backlogs in the queues at frame 0, and the temporal variations in the service received by the queues.

The queues are indexed with $i \in \{1, 2\}$. Let x_i denote the backlog in queue i in frame 0.

[1]Frame 0 is considered for notational simplicity; nevertheless, the analysis is equally valid starting the system with any other frame.

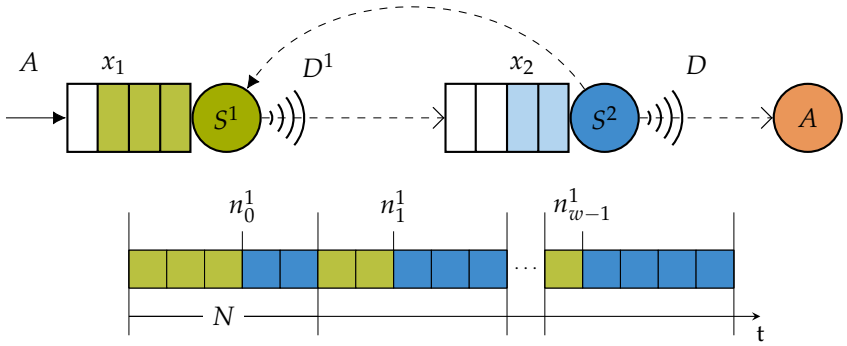

Figure 4.2: Network model of the two-hop feedback network path. The available time slots are entirely allocated to the two queues at each frame until the deadline.

Let $A^i(k)$ and $D^i(k)$ denote the cumulative arrivals and departures at queue i, in frame k. For $k = 0$, all the quantities are set to zero. For $k \geq 1$, the following quantities are defined

$$A^1(k) = y + x_1, \tag{4.1}$$

$$A^2(k) = D^1(k-1) + x_2, \tag{4.2}$$

$$D^i(k) = \sum_{j=0}^{k-1} d_j^i, \tag{4.3}$$

where d_j^i is the number of packets departed queue i in frame j. In Eq. (4.2) a one-step delay is introduced between the reception of a packet and its service at the second queue indicating that packets must be fully received before being relayed. In the following, $A(k) = A^1(k)$ and $D(k) = D^2(k)$. For analytical simplicity, it is assumed that a packet received by the controller is processed within the same frame of reception, i.e., processing latencies are negligible, and results in a new packet carrying the control command. Sensor and actuator messages can assume arbitrary size, however, we assume that their size is fixed to a maximum size of B bits.

The radio resources of the network layer are modelled following an error-prone time-slotted system. This model reflects the link-layer QoS performances achieved by the RRM schemes presented in Ch. 3 as follows. The link-layer delay and reliability achieved by a given schedule are abstracted as an individual transmission at the network-layer achieving a specific network-layer reliability. For analytical simplicity, it is assumed that both sensor and controller links are scheduled to achieve the same link-layer QoS performance. It is assumed that network-layer transmissions are independent, as their outcome depends on different frequencies and time slots allocated by their link-layer schedules, which can change between

consecutive network-layer transmissions. Therefore, the analysis in this chapter is restricted to the time domain.

The random service provided for a single network-layer transmission is modelled as a Bernoulli r.v. according to the link-layer reliability of the link. That is, a packet is lost with probability $p_e = r^{L2}$ and received with probability $1 - p_e$. Each network-layer frame, in this chapter referred to as frame, consists of N time slots to be shared between the transmissions of packets from the two transmitters. In frame k, let n_k^1 and $n_k^2 = N - n_k^1$ denote the slots used for transmitting the packets from the first transmitter and the second transmitter, respectively. Given this frame allocation, the service offered by the i-th transmitter at frame k is distributed as a Binomial r.v.

$$s_k^i(n_k^i) \sim \mathcal{B}\left(n_k^i, 1 - p_e\right). \tag{4.4}$$

The cumulative service provided by the same transmitter over k frames is the sum of Binomial random variables with parameters $1 - p_e$, which is also a Binomial r.v.

$$S^i(k) = \sum_{j=0}^{k-1} s_j^i(n_j^i) \sim \mathcal{B}\left(\sum_{j=0}^{k-1} n_j^i, 1 - p_e\right). \tag{4.5}$$

4.2.2 Derivation of Delay Violation Probability

Given the end-to-end deadline w of a control loop, the DVP of a sequence of time-critical packets generated by the sensor at frame 0 is defined as the probability that one or more packets of the sequence are not received at the actuator by the end of frame w. In this section, DVP is calculated applying Stochastic Network Calculus (SNC) [65].

The end-to-end virtual delay of the network, denoted by $W(k)$, is defined as

$$W(k) = \inf\{w \geq 1 : A(k) + x_2 \leq D(k + w - 1)\}. \tag{4.6}$$

The virtual delay quantifies the delay faced by the cumulative arrivals till frame $k - 1$. For initial backlogs x_1, x_2 and an arrival of y packets, DVP is obtained in terms of the virtual delay of the network

$$\text{DVP}(w, y, x_1, x_2) = \mathbb{P}\{W(1) > w\} = \mathbb{P}\{D(w) < y + x_1 + x_2\}. \tag{4.7}$$

The above equivalence is obtained using Eq. (4.6), where the event $\{W(1) > w\}$ implies that the cumulative departures by the end of frame w are smaller than the total number of packets in frame 0.

Proposition 1. *The DVP of a time critical arrival of y packets at $k = 0$, given initial queue backlogs x_1, x_2 is*

$$\begin{aligned}\mathrm{DVP}(w, y, x_1, x_2) =& \mathbb{P}\Big\{ \Big\{ S^2(w) < y + x_1 + x_2 \Big\} \cup \Big\{ S^2(w-1) < y + x_1 \Big\} \cup \\ & \bigcup_{u=2}^{w} \Big\{ S^2(w-u) + S^1(u-1) < y + x_1 \Big\} \Big\}. \end{aligned} \tag{4.8}$$

Proof. Applying to Eq. (4.7) the input-output relation of a queue with a dynamic server

$$D(k) \geq \min_{0 \leq u \leq k} \left[A(u) + S(k-u) \right], \tag{4.9}$$

the exact expression of DVP can be obtained as follows.

$$\begin{aligned}\mathrm{DVP}(w, y, x_1, x_2) &= \mathbb{P}\{ D(w) < y + x_1 + x_2 \} \\ &= \mathbb{P}\left\{ \min_{0 \leq u \leq w} \left[S^2(w-u) + A^2(u) < y + x_1 + x_2 \right] \right\} \\ &= \mathbb{P}\Big\{ \min_{0 \leq u \leq w} \left[S^2(w-u) + D^1(u-1) + x_2 \right] < y + x_1 + x_2 \Big\} \\ &= \mathbb{P}\Big\{ \Big\{ S^2(w) < y + x_1 + x_2 \Big\} \cup \Big\{ S^2(w-1) < y + x_1 \Big\} \cup \\ &\quad \bigcup_{u=2}^{w} \left\{ \min_{0 \leq v \leq u-1} [S^2(w-u) + S^1(u-1-v) + A(v) < y + x_1] \right\} \Big\} \\ &= \mathbb{P}\Big\{ \Big\{ S^2(w) < y + x_1 + x_2 \Big\} \cup \Big\{ S^2(w-1) < y + x_1 \Big\} \cup \\ &\quad \bigcup_{u=2}^{w} \Big\{ S^2(w-u) + S^1(u-1) < y + x_1 \Big\} \Big\}. \end{aligned} \tag{4.10}$$

□

From Eq. (4.8), we observe that the calculation of DVP is highly non-trivial. The DVP computation requires the knowledge of future, i.e., both the allocations n_k^1 and the resulting queue states, in order to calculate the cumulative services. Thus it is impossible to use DVP to obtain a dynamic scheduling policy that causally allocates the time slots using only the past information. Furthermore, computing the exact value of DVP is not tractable as it requires the calculation of the probability of union of w events that are not mutually disjoint.

4.3 Network-Layer RRM Schemes

The goal of the NC is to allocate network-layer resources to optimize the dynamic service offered by the wireless transmitters of sensor and controller to minimize the DVP of packets in a control loop while it traverses the network. As shown in Sec. 4.2, the impact of the network

queues, i.e., network state information, highly determines the value of DVP. Therefore, the impact of network state on the achievable performance of different scheduling methods is investigated.

A scheduling policy π defines the allocation of network-layer time slots to both queues in every frame until the deadline, i.e., $\pi \triangleq \{n_0^1, n_1^1, \ldots, n_{w-1}^1\}$. Different scheduling algorithms are computed based on the available network state information $\mathbf{q}_k = (q_k^1, q_k^2)$, where q_k^1 and q_k^2 denote the lengths of first and second queues in frame k, respectively. In this section, scheduling policies that minimize the DVP of y packets arrived in frame 0 are investigated in both static and dynamic scheduling scenarios.

On the one hand, *static* scheduling methods, denoted by π_S, compute a scheduling policy based on the initial state $\mathbf{q}_0$. On the other hand, *dynamic* scheduling methods, denoted by π_D, rely on the availability of the network state $\mathbf{q}_k$ at the NC, which is used to determine the allocation of slots for the next frame, i.e., at k-th frame $n_k^1 = \pi_D(\mathbf{q}_k)$. The NC can obtain the evolution of $\mathbf{q}_k$ observing the length of its transmission queue, and, knowing the initial state of the first queue, inferring its evolution observing the packet arrivals. The Enhanced Beacon (EB), already used to coordinate the resources at the link-layer, can also be exploited to deliver new slot allocations between network-layer frames.

In Sec. 4.3.1, a static policy π_S is obtained by formulating the following optimization problem

$$\underset{\pi_S \in \Pi_S}{\text{minimize}} \quad \text{DVP}\,(w, y, x_1, x_2)\,. \tag{4.11}$$

Consequently, in Sec. 4.3.2, a dynamic scheduling policy π_D is obtained solving

$$\underset{\pi_D \in \Pi_D}{\text{minimize}} \quad \text{DVP}\,(w, y, x_1, x_2)\,. \tag{4.12}$$

In Eq. (4.11) and Eq. (4.12), Π_S, Π_D denote the sets of all possible static and dynamic scheduling policies, and are non-empty as each resulting slot allocation is valid.

4.3.1 Static Scheduling

As discussed in Sec. 4.2.2, computing the exact value of DVP is not tractable as it requires the calculation of the probability of union of w events that are not mutually disjoint. Therefore, an upper bound on DVP, the Delay Violation Probability Upper Bound (DVPUB), is computed

using the *union bound* as follows

$$\begin{aligned}\text{DVPUB}(w,y,x_1,x_2) &\leq \mathbb{P}\left\{S^2(1+w)<y+x_1+x_2\right\}+\\ &\quad\sum_{u=1}^{1+w}\mathbb{P}\left\{S^2(1+w-u)+S^1(u-1)<y+x_1\right\}\\ &=\mathbb{P}\left\{S^2(1+w)\leq y+x_1+x_2-1\right\}+\\ &\quad\sum_{u=1}^{1+w}\mathbb{P}\left\{S^2(1+w-u)+S^1(u-1)\leq y+x_1-1\right\}. \quad (4.13)\end{aligned}$$

DVPUB is the sum of Cumulative Distribution Function (CDF) of the cumulative service of the second server S^2 and the CDF of the sum of the cumulative services of servers S^1, S^2.

Applying Eq. (4.5) to Eq. (4.13), the DVPUB resulting from the allocation of $\mathbf{n}^2 = \{n_1^1,\ldots,n_w^1\}$ slots at the second transmitter and $\mathbf{n}^2 = N - \mathbf{n}^1$ slots at the first one is given by

$$\begin{aligned}\text{DVPUB}(w,y,x_1,x_2,\mathbf{n}^2) &= \sum_{x=0}^{y+x_1+x_2-1}\mathbb{P}\left\{\sum_{i=0}^{w}s_i^2\left(n_i^2\right)=x\right\}\\ &\quad+\sum_{u=1}^{1+w}\sum_{z=0}^{y+x_1-1}\mathbb{P}\left\{\sum_{j=0}^{w-u}s_j^2\left(n_j^2\right)+\sum_{k=0}^{u-2}s_k^1\left(n_k^1\right)=z\right\}\\ &\overset{(1)}{=}\sum_{x=0}^{y+x_1+x_2-1}\left(\frac{1}{p_e}-1\right)^x\binom{\sum_{i=0}^{w}n_i^2}{x}p_e^{\sum_{i=0}^{w}n_i^2}\\ &\quad+\sum_{u=1}^{1+w}\sum_{z=0}^{y+x_1-1}\left(\frac{1}{p_e}-1\right)^z p_e^{\sum_{j=0}^{w-u}n_j^2-\sum_{k=0}^{u-2}n_k^2+(u-1)N}\\ &\quad\binom{\sum_{j=0}^{w-u}n_i^2-\sum_{k=0}^{u-2}n_j^2+(u-1)N}{z}. \quad (4.14)\end{aligned}$$

Step (1) was computed knowing that $S^i(k)$ is distributed as a Binomial r.v. according to Eq. (4.5).

The goal of the NC is to minimize the DVPUB to find static scheduling policies. The DVPUB of Eq. (4.14), however, is not convex. Therefore, following a similar approach as in the state-of-the-art [27], a looser convex bound, the Wireless Transient Bound (WTB), can be obtained by applying the *Chernoff bound* to Eq. (4.13).

$$\begin{aligned}\text{WTB}(w,y,x_1,x_2) &= \min_{s>0}\mathbb{E}\left[e^{-s\,S^2(1+w)}\right]e^{s\,(y+x_1+x_2-1)}+\\ &\quad\sum_{u=1}^{1+w}\mathbb{E}\left[e^{-s\left[S^1(u-1)+S^2(1+w-u)\right]}\right]e^{s\,(y+x_1-1)}. \quad (4.15)\end{aligned}$$

The calculation of WTB for an IWSN link achieving a link-layer reliability p_e is obtained by computing the Mellin transform of the cumulative service of Eq. (4.5)

$$\begin{aligned}\mathbb{E}\left[e^{-s\,S(m,\mathbf{n})}\right] &= \mathbb{E}\left[e^{-s\sum_{i=0}^{-1+\sum_{j=0}^{m-1} n_j} b_i}\right] \\ &= \mathbb{E}\left[\left(e^{-s\,b_i}\right)^{\sum_{i=0}^{m-1} n_i}\right] \\ &= \left[(1-p_e)e^{-s}+p_e\right]^{\sum_{i=0}^{m-1} n_i},\end{aligned} \tag{4.16}$$

where b_i indicates the i.i.d. Bernoulli random variables of each transmission outcome.

Finally, combining Eq. (4.15) and (4.16)

$$\begin{aligned}\mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2) = \min_{s>0} &\left[(1-p_e)e^{-s}+p_e\right]^{\sum_{i=0}^{w} n_i^2} e^{s\,(y+x_1+x_2-1)} + \\ &\sum_{u=1}^{1+w}\left[(1-p_e)e^{-s}+p_e\right]^{(u-1)N-\sum_{j=0}^{u-2} n_j^2+\sum_{i=0}^{w-u} n_i^2} e^{s\,(y+x_1-1)}.\end{aligned} \tag{4.17}$$

Given the initial queue backlogs $\mathbf{q}_0 = (x_1, x_2)$, we aim to solve the upper bound minimization problem below

$$\pi_S^\star(\mathbf{q}_0) = \underset{\mathbf{n}^2\in\{1,\ldots,N-1\}}{\arg\min}\ \mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2). \tag{4.18}$$

As shown by Theorem 1, a solution to the problem of Eq. (4.18) can be efficiently derived when the domain of the optimization variable $\mathbf{n}^2$ is continuous. This can be achieved by solving

$$\widetilde{\pi}_S^\star(\mathbf{q_0}) = \underset{\mathbf{n}^2\in[1,N-1]^w}{\arg\min}\ \mathrm{WTB}(w,y,x_1,x_2,\mathbf{n}^2), \tag{4.19}$$

then, by converting $\widetilde{\pi}_S^\star$ to the integer domain to find a feasible solution to Eq. (4.11), thus obtaining the scheduling policy $\hat{\pi}_S^\star$.

Theorem 1. *The optimization problem in* (4.19) *is convex.*

Proof. To simplify the analysis of Eq. (4.17), the auxiliary terms $\alpha = (1-p_e)e^{-s}+p_e$ and $\beta = k_a + x_1 - 1$ are used

$$\begin{aligned}\mathrm{WTB}(\mathbf{n}^2) &= \min_{s>0} \alpha^{\sum_{i=0}^{w} n_i^2} e^{s\,(\beta+x_2)} + \sum_{u=1}^{1+w} \alpha^{(u-1)N-\sum_{j=0}^{u-2} n_j^2+\sum_{k=0}^{w-u} n_k^2} e^{s\,\beta} \\ &= \min_{s>0} e^{s\,\beta}\left[\alpha^{\sum_{i=0}^{w} n_i^2} e^{s x_2} + \sum_{u=1}^{1+w} \alpha^{(u-1)N-\sum_{j=0}^{u-2} n_j^2+\sum_{k=0}^{w-u} n_k^2}\right].\end{aligned} \tag{4.20}$$

Applying the definition of convexity to Eq. (4.20), $\forall \mathbf{n}, \mathbf{m} \in \mathbb{N}^{1\times w}, \lambda \in [0,1]$ yields

$$
\begin{aligned}
&\mathrm{WTB}\left(\lambda\mathbf{n}+(1-\lambda)\mathbf{m}\right)\\
&= \min_{s>0} e^{s\beta}\left[\alpha^{\sum_{i=0}^{w}\lambda n_i^1+(1-\lambda)n_i^2}e^{sx_2}+\sum_{u=1}^{1+w}\alpha^{(u-1)N-\sum_{j=0}^{u-2}\lambda n_j^1+(1-\lambda)n_j^2+\sum_{k=0}^{w-u}\lambda n_k^1+(1-\lambda)n_k^2}\right]\\
&\overset{(1)}{\leq} \min_{s>0} e^{s\beta}\left[e^{sx_2}\left(\lambda\alpha^{\sum_{i=0}^{w}n_i^1}+(1-\lambda)\alpha^{\sum_{i=0}^{w}n_i^2}\right)+\right.\\
&\quad\left.\sum_{u=1}^{1+w}\alpha^{(u-1)N}\left(\lambda\alpha^{-\sum_{j=0}^{u-2}n_j^1}+(1-\lambda)\alpha^{-\sum_{j=0}^{u-2}n_j^2}+\lambda\alpha^{\sum_{k=0}^{w-u}n_k^1}+(1-\lambda)\alpha^{\sum_{k=0}^{w-u}n_k^2}\right)\right]\\
&= \min_{s>0} e^{s\beta}\left[\lambda\left(\alpha^{\sum_{i=0}^{w}n_i^1}e^{sx_2}+\sum_{u=1}^{1+w}\alpha^{(u-1)N}\alpha^{-\sum_{j=0}^{u-2}n_j^1}\alpha^{-\sum_{j=0}^{u-2}n_j^1}\right)+\right.\\
&\quad\left.(1-\lambda)\left(\alpha^{\sum_{i=0}^{w}n_i^2}e^{sx_2}+\sum_{u=1}^{1+w}\alpha^{(u-1)N}\alpha^{-\sum_{j=0}^{u-2}n_j^2}\alpha^{-\sum_{j=0}^{u-2}n_j^2}\right)\right]\\
&\overset{(2)}{\leq} \lambda\min_{s>0} e^{s\beta}\left[\alpha^{\sum_{i=0}^{w}n_i^1}e^{sx_2}+\sum_{u=1}^{1+w}\alpha^{(u-1)N}\alpha^{-\sum_{j=0}^{u-2}n_j^1}\alpha^{-\sum_{j=0}^{u-2}n_j^1}\right]+\\
&\quad(1-\lambda)\min_{s>0} e^{s\beta}\left[\alpha^{\sum_{i=0}^{w}n_i^2}e^{sx_2}+\sum_{u=1}^{1+w}\alpha^{(u-1)N}\alpha^{-\sum_{j=0}^{u-2}n_j^2}\alpha^{-\sum_{j=0}^{u-2}n_j^2}\right]\\
&= \lambda\mathrm{WTB}\left(\mathbf{n}\right)+(1-\lambda)\mathrm{WTB}\left(\mathbf{m}\right). \qquad (4.21)
\end{aligned}
$$

In Eq. (4.21), step (1) is obtained applying the definition of convexity to $\alpha^{f(\lambda\mathbf{n^1}+(1-\lambda)\mathbf{n^2})}$, which is an exponential, therefore convex. Step (2) is obtained observing that the minimum of the sum is smaller or equal than the sum of the minima. □

Thanks to Theorem 1, well-known convex optimization algorithms can be used to solve the relaxed problem, such as the subgradient or interior-point algorithms, which provide scalability for an increasing number of slots N and frames within the deadline w. In this chapter, the relaxed problem is solved using the nonlinear programming solver *fmincon* available in Matlab™ employing the Sequential Quadratic Programming (SQP) algorithm. Once $\widetilde{\pi}_S^\star$ is found, a conversion to the integer domain is needed in order to determine $\hat{\pi}_S^\star$. Although the optimal selection of an integer solution would require the exploration of the entire problem's domain, heuristic methods can be applied to find a solution in the neighbourhood of $\widetilde{\pi}_S^\star$. To this end, different neighbour search methods are investigated in order to achieve near-optimal performances.

The simplest way to derive $\hat{\pi}_S^*$ is to round each frame allocation of $\widetilde{\pi}_S^\star$ to its closest integer value. We refer to this method as WTB-R. Alternatively, the heuristic policy can be found in the neighbourhood of the continuous solution. For each frame allocation $n_i^1 \in \widetilde{\pi}_S^\star$, two

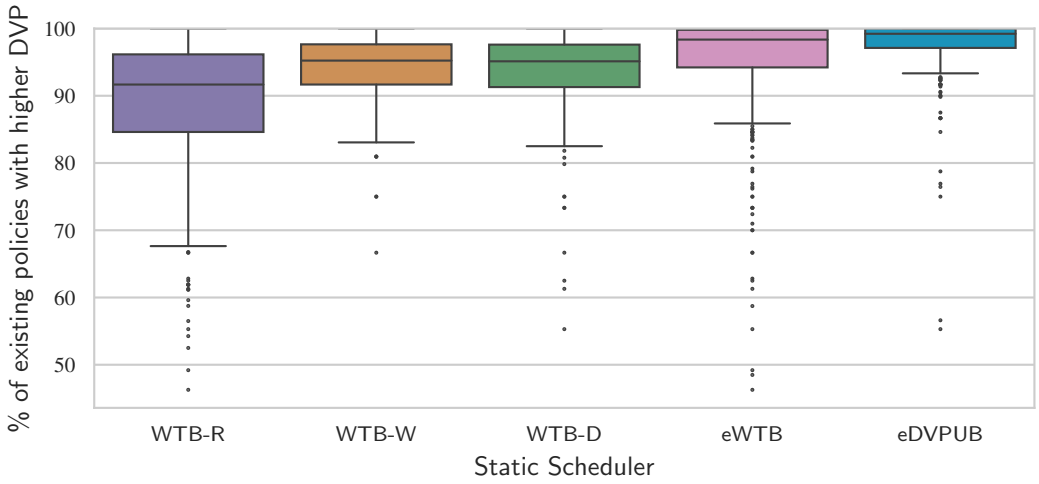

Figure 4.3: Performance of the proposed static schedulers for different system parameters. For each scheduler, the percentage of existing policies that achieve higher DVP is evaluated.

integer values are computed applying the floor and ceiling functions. A reduced search space is constructed computing all the combinations of the integer values for each frame until the deadline, which leads to a total of 2^w combinations. As final step, Eq. (4.14) and (4.17) are used to evaluate each combination and identify the best one. The static policies corresponding to the evaluation of Eq. (4.14) and (4.17) are referred to as WTB-D and WTB-W, respectively.

The performances of the different heuristics have been evaluated via extensive simulations of a broad range of values for each parameter of the WTB, i.e., for $x_1, x_2 \in \{0,1,2\}$, $w \in \{2,3,4,5,6\}$, $N \in \{2,3,4,5\}$, and $p_e \in \{0.2, 0.33, 0.4, 0.5\}$. For performance comparison, two additional schedulers have been evaluated, eDVPUB and eWTB, which exhaustively explore the problem's domain to find the policies that respectively achieve the minimum values of DVPUB, cf. Eq. (4.14), and WTB, cf. Eq. (4.17). Furthermore, the performance of each scheduler is compared with the performance of the policy that achieves the minimum DVP. This optimal policy is found by exploring the DVP of all possible policies via simulations.

Fig. 4.3 shows the performance of the proposed static schedulers by computing, for each scheduler and system configuration, the percentage of existing policies that achieve higher DVP. As expected, the exhaustive search methods eWTB and eDVPUB achieve the best results, with eDVPUB finding, in the large majority of cases, the top 10% static policies. The performance gap between eWTB and eDVPUB is introduced by the Chernoff bound in Eq. (4.16). These methods, however, are not efficient as they require the exhaustive exploration of the entire problem domain. The heuristic methods WTB-R/W/D can be used to efficiently find static policies for a broad range of parameters. WTB-R achieves the worst performance and is the heuristic with lowest complexity. Interestingly, WTB-W and WTB-D achieve similar performances of the exhaustive methods, with WTB-W achieving better performances. This

can be explained by the fact that, in WTB-D, Eq. (4.17) is used to solve the relaxed problem, while Eq. (4.14) to find the integer solution.

4.3.2 Dynamic Scheduling

As discussed in Sec. 4.2.2, in order to compute the DVP of Eq. (4.8), the knowledge of future slot allocations and queue states is needed. Thus, it is impossible to use DVP to obtain a dynamic scheduling policy which causally allocates the time slots using only the past information. To address this problem, the following upper bound for DVP is obtained using Markov's inequality, which can be used to obtain dynamic scheduling policies.

$$\begin{aligned}\mathrm{DVP}(w, y, x_1, x_2) &= \mathbb{P}\{D(w) \leq y + x_1 + x_2 - 1\} \\ &= \mathbb{P}\left\{y + x_1 + x_2 - D(w) \geq 1\right\} \leq \quad y + x_1 + x_2 - \mathbb{E}[D(w)], \quad (4.22)\end{aligned}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. From Eq. (4.22) it is possible to infer that maximizing the expected cumulative departures (throughput) of the network minimizes the upper bound of the DVP and thus potentially reduces DVP. Therefore, in the following a heuristic schedule is computed by solving the expected throughput maximization problem stated below.

$$\underset{\pi_D \in \Pi_D}{\text{maximize}} \quad \mathbb{E}[D(w)] = \sum_{k=0}^{w-1} \mathbb{E}\left[d_k^i\right]. \tag{4.23}$$

In order to solve the optimization problem in Eq. (4.23), a discrete-time, finite-horizon Markov Decision Process (MDP) is formulated. The term $\mathbf{q}_k$ denotes the state of the system and n_k^1 denotes the action in frame k. The maximum number of slots in a frame is N and therefore $n_k^1 \in \{0, 1, \ldots, N\}$. Given n_k^1, Eq. (4.4) yields

$$\begin{aligned}\mathbb{P}\{s_k^1 = r\} &= \binom{n_k^1}{r}(1 - p_e)^r p_e^{n_k^1 - r}, \\ \mathbb{P}\{s_k^2 = r\} &= \binom{N - n_k^1}{r}(1 - p_e)^r p_e^{N - n_k^1 - r}.\end{aligned}$$

The queues evolve as below:

$$q_{k+1}^1 = \max(q_k^1 - s_k^1, 0), \tag{4.24}$$

$$q_{k+1}^2 = \max(q_k^2 - s_k^2, 0) + \min(q_k^1, s_k^1). \tag{4.25}$$

Note that the number of departures from the first queue in frame k equals $\min(q_k^1, s_k^1)$, which are added to the second queue to be served in frame $k + 1$.

In the following, the transition probabilities for the states are formulated. Note that the initial backlogs in the queues are $(y + x_1, x_2)$, where y is due to the message of interest. Also, $q_0^1 = y + x_1$ and $q_0^2 = x_2$. The set of possible states in the system system is computed as follows. In any frame k, a feasible state (q_k^1, q_k^2) should satisfy the following conditions:

$$q_k^1 \leq q_{k-1}^1, \tag{4.26}$$

$$q_k^1 + q_k^2 \leq q_{k-1}^1 + q_{k-1}^2. \tag{4.27}$$

Conditions Eq. (4.26) and Eq. (4.27) follow from the fact that arrivals after the message of interest are ignored and in every frame each queue will receive certain service. Note that while the length of the first queue can only decrease as the packets are served, the length of the second queue may increase up to $y + x_1 + x_2$ as departures from first queue are added to the second queue. Therefore, for every state $\mathbf{q}_k$ in the state space, say $\mathcal{Q}$, $q_k^1 \in \{0, 1, \ldots, y + x_1\}$ and $q_k^2 \in \{0, 1, \ldots, y + x_1 + x_2\}$. This implies that $\mathcal{Q}$ can contain at most $S = (y + x_1 + 1)(y + x_1 + x_2 + 1)$ possible states.

Consider that in frame k, $q_k^1 = l^1$ and $q_k^2 = l^2$. The transition probabilities to the states are represented by $q_{k+1}^1 = l_+^1$ and $q_{k+1}^2 = l_+^2$. The following cases arise.

Case 1: $l_+^1 > l^1$ or $l_+^1 + l_+^2 > l^1 + l^2$. From Eq. (4.26) and Eq. (4.27), is it inferred that

$$\mathbb{P}\{q_{k+1}^1 = l_+^1, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\} = 0.$$

Case 2: $0 < l_+^1 \leq l^1$, $0 < l_+^2$, and $l_+^1 + l_+^2 \leq l^1 + l^2$. In this case $s_k^1 < q_k^1 = l^1$ and $s_k^2 < q_k^2 = l^2$. Eq. (4.24) yields

$$q_{k+1}^1 = q_k^1 - s_k^1 \quad \Rightarrow \quad s_k^1 = l^1 - l_+^1.$$

The number packets served from the second queue are computed from Eq. (4.25).

$$q_{k+1}^2 = q_k^2 - s_k^2 + s_k^1 \quad \Rightarrow s_k^2 = l^2 - l_+^2 + l^1 - l_+^1.$$

Therefore,

$$\begin{aligned}\mathbb{P}\{q_{k+1}^1 = l_+^1, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\} \\ = \mathbb{P}\{s_k^1 = l^1 - l_+^1, s_k^2 = l^2 - l_+^2 + l^1 - l_+^1\}.\end{aligned}$$

Case 3: $l_+^1 = 0$, $0 < l_+^2$, and $l_+^2 \leq l^1 + l^2$. In this case all l^1 packets from the first queue are served. This implies $s_k^1 \geq q^1 = l^1$. Using similar analysis as above, the following relationship is obtained

$$\begin{aligned}\mathbb{P}\{q_{k+1}^1 = 0, q_{k+1}^2 = l_+^2 | q_k^1 = l^1, q_k^2 = l^2\} \\ = \mathbb{P}\{s_k^1 \geq l^1, s_k^2 = l^2 - l_+^2 + l^1\}.\end{aligned}$$

Case 4: $l_+^1 = l^1, l_+^2 = 0$. In this case $s_k^1 = 0$, and all l^2 packets from the second queue are served, i.e., $s_k^2 \geq q^2 = l^2$. Eq. (4.25) yields

$$\begin{aligned}\mathbb{P}\{q_{k+1}^1 = l^1, q_{k+1}^2 = 0 | q_k^1 = l^1, q_k^2 = l^2\} \\ = \mathbb{P}\{s_k^1 = 0, s_k^2 \geq l^2\}.\end{aligned}$$

Note that the case $0 \leq l_+^1 < l^1$ and $l_+^2 = 0$ cannot happen as $l^1 - l_+^1$ packets will be added to the second queue in the current slot. All the above cases are written assuming that $l^1 > 0$ and $l^2 > 0$. If either of them is zero, then the transition probability only involves the probability for service received by the non-empty queue.

Given the initial state $\mathbf{q}_0 = (y + x_1, x_2)$, the NC computes a scheduling policy $\pi_D^\star$ that solves the maximization problem of Eq. (4.23). For this, the reward r_k of a policy π_D for a given state $\mathbf{q}_k$ is defined as the expected number of departures from the system, i.e., the expected number of packets that are served at the second queue under this policy, and is given by

$$r_k\left(\mathbf{q}_k, \pi_D(\mathbf{q}_k)\right) = \mathbb{E}\left[d_k^i | \, \pi_D(\mathbf{q}_k)\right] = \mathbb{E}\left[\min\left(q_k^2, s_k^2\right) | \, \pi_D(\mathbf{q}_k)\right]. \tag{4.28}$$

The total reward, obtained evaluating Eq. (4.28) over a horizon of w frames, is equal to Eq. (4.23). Therefore, the objective of the MDP is equal to the objective of Eq. (4.23).

Value iteration algorithms solve the MDP optimization recursively computing a value function J based on the Bellman's equation [106]. The optimal value function $J(\mathbf{q}_k)$ given a state $\mathbf{q}_k$ is

$$\begin{aligned}J_k(\mathbf{q}_k) = \underset{\pi_D \in \Pi_D}{\text{maximize}} \;\; & r_k(\mathbf{q}_k, \pi_D(\mathbf{q}_k)) + \\ & \sum_{\mathbf{q}_{k+1} \in \mathcal{Q}_{k+1}} \mathbb{P}\{\mathbf{q}_{k+1} | \mathbf{q}_k, \pi_D(\mathbf{q}_k)\} J_{k+1}(\mathbf{q}_{k+1}),\end{aligned} \tag{4.29}$$

where $\mathbb{P}\{\mathbf{q}_{k+1} | \mathbf{q}_k \pi_D(\mathbf{q}_k)\}$ is the transition probability from state $\mathbf{q}_k$ to state $\mathbf{q}_{k+1}$ in one time step using $\pi_D(\mathbf{q}_k)$, $\mathcal{Q}_{k+1}$ denotes the set of all states reachable from $\mathbf{q}_k$ with one time step transition.

By the construct of the MDP, it is easy to see that $\pi_D^\star$ is optimal for Eq. (4.23), which is stated in the following theorem.

Theorem 2. *$\pi_D^\star$ is throughput optimal, i.e.*

$$\pi_D^\star = \underset{\pi_D \in \Pi_D}{\arg\max} \;\; \mathbb{E}\left[D(w)\right].$$

For a finite number of states and actions, the optimal policy $\pi_D^\star$ for the MDP can be found by computing the optimal value function in Eq. (4.29) by backward recursion [106]. The calculation of the optimal dynamic scheduling policy satisfying Theorem 2 can be performed using the finite-horizon value iteration algorithm. Value iteration can compute the optimal policy efficiently, particularly for finite-horizon problems. For each epoch k until the deadline w, for each state $\mathbf{q}_k$, the value function in Eq. (4.29) is computed for all possible actions. Therefore, computing the optimal dynamic policy requires $\mathcal{O}\left((N+1)\,Sw\right)$ operations.

4.4 Performance Evaluation

In this section, the performance of static and dynamic network-layer scheduling policies are evaluated via extensive simulations. The network simulator is implemented in C, while Matlab™ and python are used to compute, respectively, static and dynamic policies. The evaluation is performed taking into account all system parameters - different initial queue backlogs x_1, x_2, application deadlines w, frame lengths N, and service Packet Error Rates (PERs) p_e. The performance evaluation of this section is organized as follows. Sec. 4.4.1 and Sec. 4.4.2 present, respectively, the performance of static and dynamic scheduling policies. Sec. 4.4.3 evaluates the performance gap between the proposed static and dynamic methods.

4.4.1 Static Scheduling Policies

Static scheduling policies are constructed based on the information of the network's queues at the moment of the arrival. The policies are calculated prior to the transmission of the packet and statically define the allocation of slots until the deadline. The DVP achieved by the WTB-W and eDVPUB schedulers is compared with the optimal static allocation obtained using exhaustive search using simulation. Due to its high complexity, the performances of the optimal policy is shown for smaller parameter sets. Exhaustive schemes are shown to evaluate the performance of WTB-B and do not represent a feasible way of computing semi-static scheduling policies. The last comparison scheme is a agnostic allocation of half of the slots for uplink and half of the slots for downlink transmission. In the case of an odd number of slots, one extra slot is allocated to the first link. We refer to this scheme as 50/50. In all the figures, the value of p_e is set to 0.2.

Fig. 4.4 shows the DVP achieved by the static schedulers for different application deadlines, backlogs and $N = 4$. The proposed static schedulers WTB-W and eDVPUB achieve close-to-optimal DVP for all backlog sizes and deadlines. The figure shows that taking into account the initial queue states is always beneficial and provides up to one order of magnitude DVP improvement with increasing w and different backlogs. Furthermore, as the deadline w

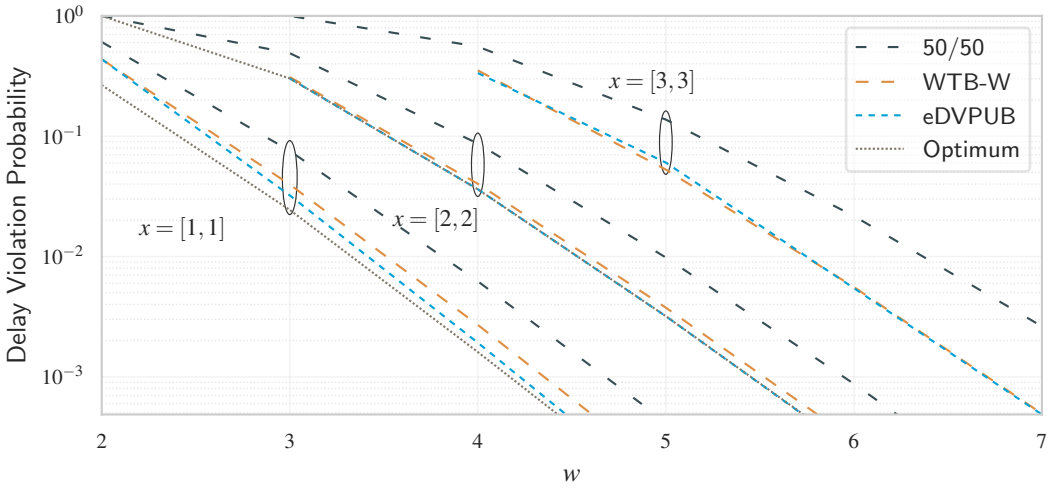

Figure 4.4: DVP achieved by static schedulers for different deadlines w, increasing backlogs x_1, x_2, $N = 4$, $p_e = 0.2$.

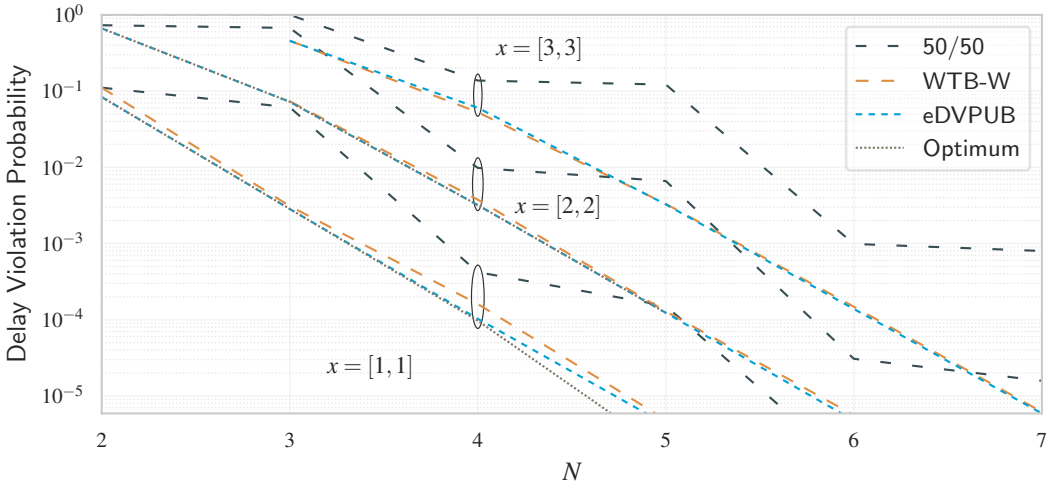

Figure 4.5: DVP achieved by static schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w = 5$, $p_e = 0.2$.

increases, the performance gap of the proposed method with respect to 50/50 increases. For x_1, x_2 equal to 1, eDVPUB provides a minor DVP improvement with respect to WTB-W, at the expense of higher computational complexity, while, for higher backlogs, its performance difference is negligible.

Fig. 4.5 presents DVP values for different frame lengths, backlogs, and $w = 5$. Also in this case, the DVP improvement of the proposed static schedulers with respect to 50/50 is confirmed and increases increasing the frame length. The oscillating behaviour of 50/50 is caused by the different ratio of allocated slots to the two transmitters for even and odd frame lengths. Results show a minor difference between WTB-W and eDVPUB, and an increasing difference with respect to optimal for increasing N when $x1, x2$ are equal to 1.

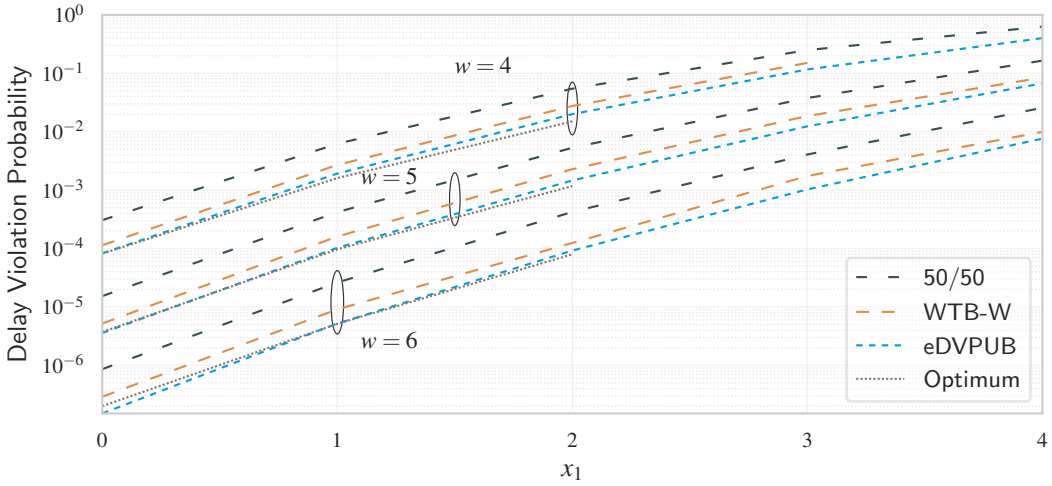

Figure 4.6: DVP achieved by static schedulers for different backlogs x_1 and deadlines w, $x_2 = 1$, $N = 4$, $p_e = 0.2$.

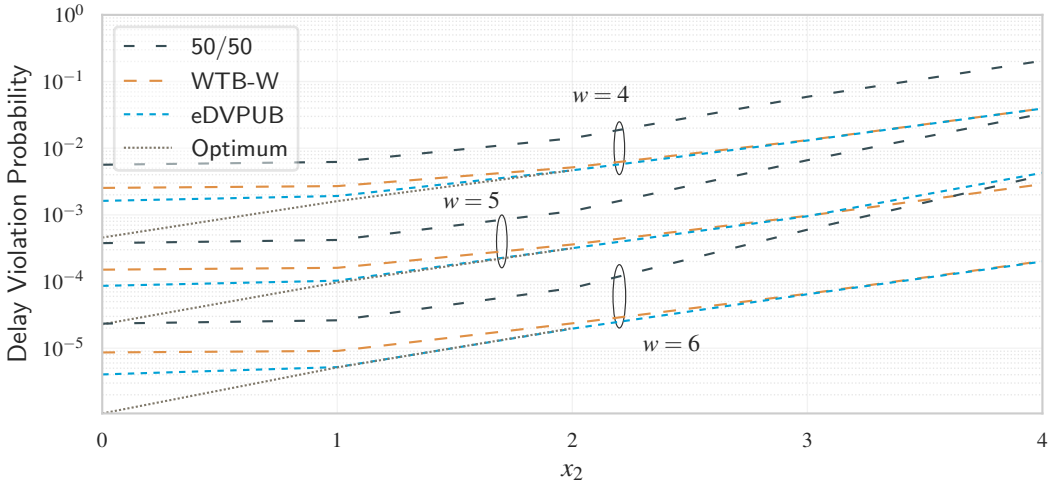

Figure 4.7: DVP achieved by static schedulers for different backlogs x_2 and deadlines w, $x_1 = 1$, $N = 4$, $p_e = 0.2$.

Fig. 4.6 and 4.7 show the impact of initial backlog on the DVP of the time-critical arrival for different deadlines and $N = 4$. Increasing x_1 has a stronger impact than x_2 on the achievable DVP. This is intuitive as the packets must be sent by both transmitters. It is possible to notice that increasing x_2, the gap between the static schedulers and 50/50 increases. Thus, adopting the proposed scheduler is highly beneficial when the buffer of the second queue is larger. In both figures, exploiting the initial network information leads to a gain of approximately half order of magnitude with respect to 50/50 with increasing initial backlogs. As shown in Fig. 4.6, eDVPUB achieves near-optimal DVP and performs better than WTB-W, while the performance gap is constant with increasing x_1. Differently, in Fig. 4.7, the gap between the static schedulers and the optimal policy decreases with increasing x_2, achieving close-to-optimal DVP.

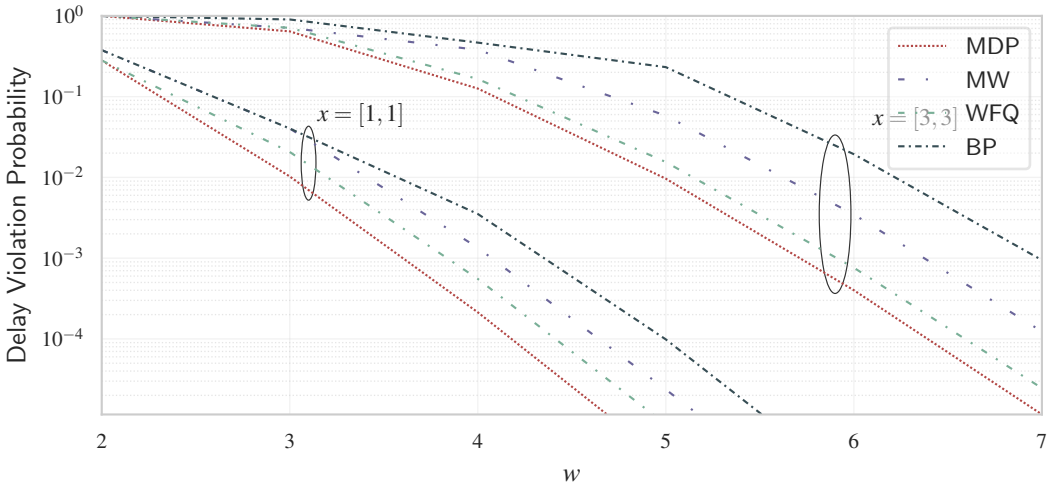

Figure 4.8: DVP achieved by dynamic schedulers for different deadlines w, increasing backlogs x_1, x_2, $N = 6$, $p_e = 0.4$.

The results of this section demonstrate the effectiveness of the using the Union and Chernoff bounds in order to minimize DVP. Results show that both bounds are able to characterize DVP with sufficient precision, and their application for static scheduling policies results in close-to-optimal DVP. On the one hand, thanks to its convexity, WTB can be efficiently optimized to compute policies that achieve the top 10% DVP for the majority of investigated system parameters. On the other hand, DVPUB achieves lower DVP, which, however, can only be exploited when the problem dimensions are small enough. It is possible to notice that the tightness of the bounds improves as x_2 increases. In fact, for large x_2, DVP is mainly determined by the packet transmission of the second transmitter. This is different than x_1, where backlogged packets are sent by both transmitters.

4.4.2 Dynamic Scheduling Policies

Dynamic schedulers benefit from up-to-date network state information and reallocate time slots to the transmitters at each frame until the deadline. The proposed MDP-based scheduling policy is compared to classical packet scheduling algorithms: Max Weight (MW) [94], Weighted-Fair Queuing (WFQ) [99], and Backpressure (BP) [134].

At each frame, MW allocates all slots to the transmitter with the maximum backlog, while WFQ allocates slots proportionally to the ratio between the queue sizes. BP allocates all slots to the transmitter with maximum backpressure. In our scenario, the backpressure at the first transmitter is $x_1 - x_2$ while at the second transmitter is equal to x_2. In all the figures, the value of p_e is set to 0.4.

Fig. 4.8 shows the DVP achieved by the static schedulers for different application deadlines, backlogs, and $N = 6$. The proposed MDP-based scheduler outperforms all the other methods.

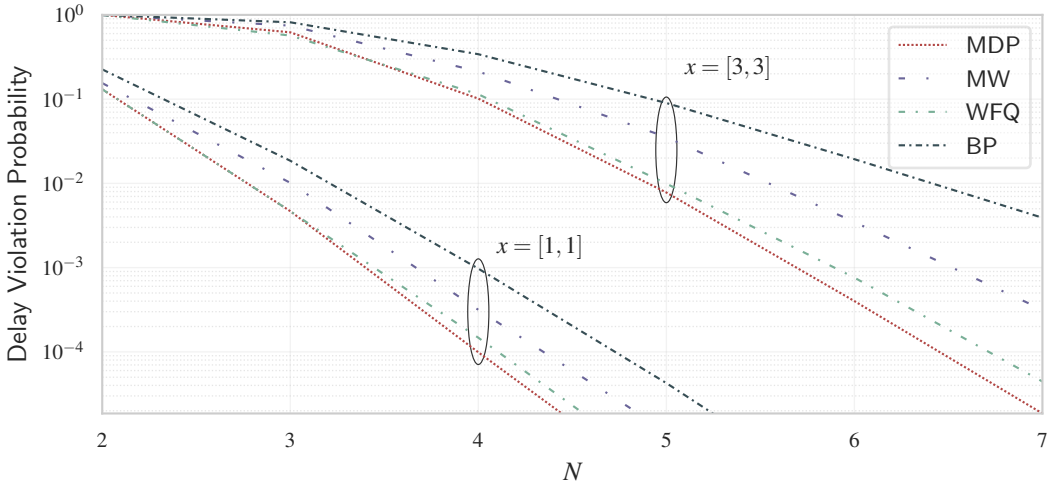

Figure 4.9: DVP achieved by dynamic schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w = 6$, $p_e = 0.4$.

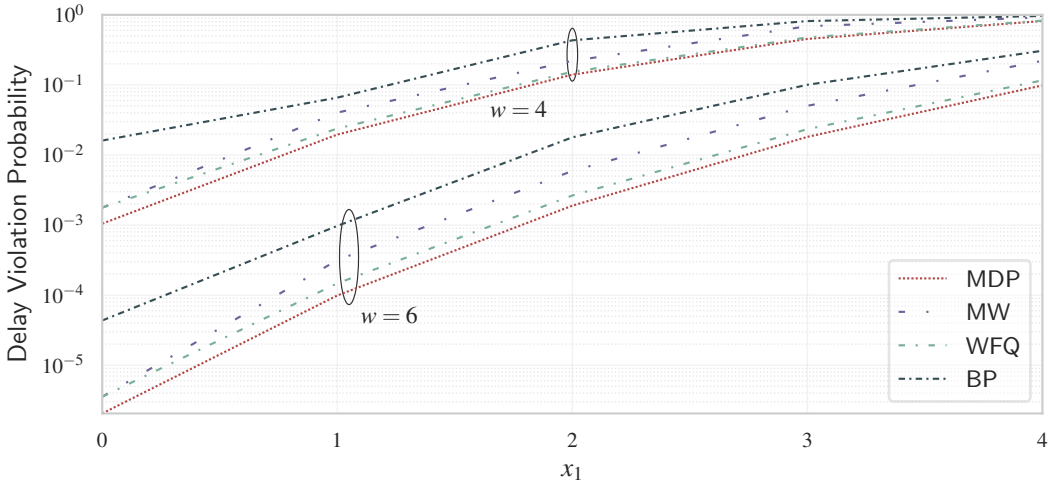

Figure 4.10: DVP achieved by dynamic schedulers for different backlogs x_1 and deadlines w, $x_2 = 1$, $N = 4$, $p_e = 0.4$.

BP and MW achieve the worst DVP as they allocate all the slots to a single transmitter in each frame. Similarly to MDP, WFQ allows a granular allocation of slots to the transmitters and achieves the smallest performance gap.

Fig. 4.9 presents DVP values for different frame lengths, backlogs, and $w = 6$. Also in this case MDP achieves lower DVP and the performances of the other methods are in line with the previous scenario. Furthermore, the performance of MDP with respect to all the schedulers improves with increasing the frame length. This advantage arises from the fact that, the action space of MDP is larger for large N, which results in more accurate slot allocations.

Fig. 4.10 and 4.11 show the impact of initial backlog on the DVP of the time-critical arrival for different deadlines and $N = 4$. Again, increasing x_1 has a higher impact on DVP compared

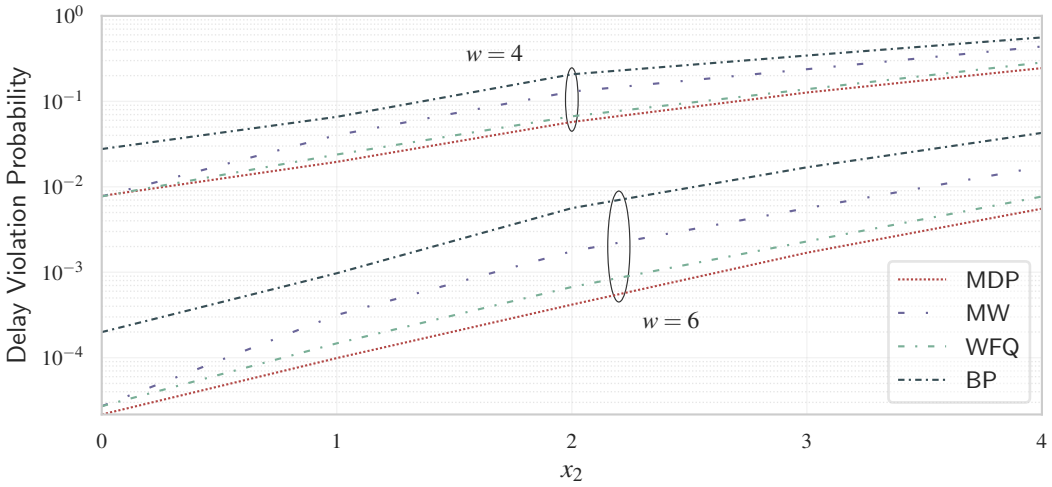

Figure 4.11: DVP achieved by dynamic schedulers for different backlogs x_2 and deadlines w, $x_1 = 1$, $N = 4$, $p_e = 0.4$.

to x_2. Due to the entire allocation of slots to a transmitter in a frame, the performance gap of MDP and MW increases as x_1 and x_2 increase. WFQ, however, maintains a constant gap, being able to adapt the allocations to different initial backlogs scenarios.

Simulations have revealed that throughput-optimal MDP can reduce the DVP on time-critical arrival subject to initial buffering in the network. In particular, the granular allocation of resources provided by WFQ and MDP provides lower DVP with respect to BP and MW. MDP outperforms all the other scheduling methods and its potential is fully exploited for large frame lengths. Thanks to this effect, given the same application deadline, MDP is able to achieve lower DVP with a higher frame length N and a smaller number of frames w.

4.4.3 Impact of Network State Information

A direct comparison of the proposed static and dynamic scheduling policies allows to quantify the performance improvement achieved by exploiting up-to-date queue states. In all the figures, the value of p_e is set to 0.4.

In Fig. 4.12 the DVP achieved by the proposed scheduling policies is presented for different deadlines, backlogs, and $N = 4$. The gap between dynamic and static scheduling policies increases with increasing deadlines, reaching multiple order of magnitudes. This effect is intuitive as, at each frame, dynamic scheduling benefits from up-to-date information about the network queues. A similar effect can be observed in Fig. 4.13, that shows the DVP values for different frame lengths, backlogs, and $w = 6$. Results confirm that additional queue states are exploited by dynamic policies to achieve lower DVP compared to static policies. Finally, Fig. 4.14 and 4.15 show the impact of initial backlog on the DVP for different deadlines and $N = 6$. For fixed system parameters, increasing the initial backlogs results in higher

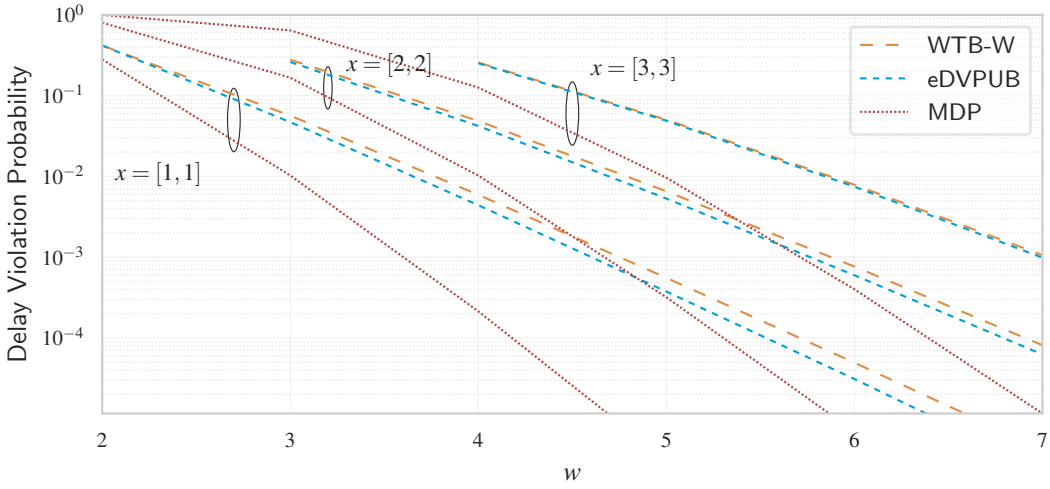

Figure 4.12: DVP achieved by static and dynamic schedulers for different deadlines w, increasing backlogs x_1, x_2, $N = 6$, $p_e = 0.4$.

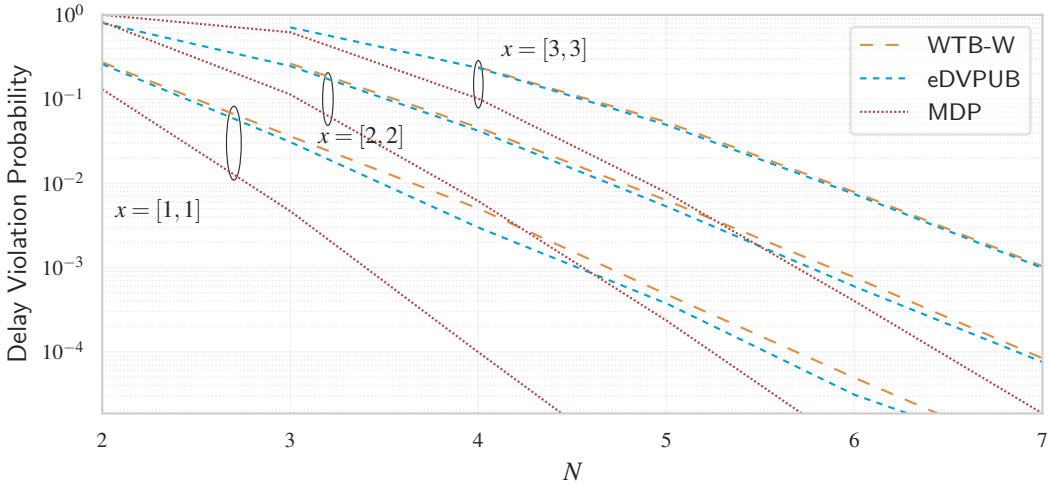

Figure 4.13: DVP achieved by static and dynamic schedulers for different frame sizes N, increasing backlogs x_1, x_2, $w = 6$, $p_e = 0.4$.

achievable DVP, which translates into smaller performance gaps between static and dynamic scheduling schemes.

Simulation results have shown that exploiting up-to-date queue states leads to considerable improvements in the DVP of time-critical packets. Initial queue states improve the achievable DVP by one order of magnitude with respect to a 50/50 allocation for smaller values of system parameters. Furthermore, the availability of per-frame queue states provides an improvement of several orders of magnitude. The improvement is particularly large for longer frames and higher deadlines, which shows that exploiting the queue states does not only improve the allocation of slots between frames but makes each allocation more efficient. A general trend describes the performance gap of dynamic and static schedulers – the lower the achieved DVP,

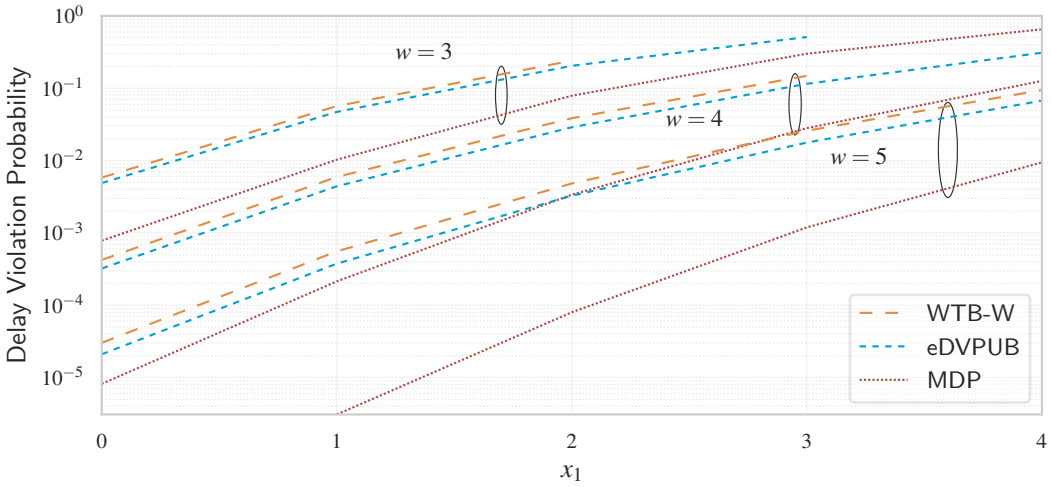

Figure 4.14: DVP achieved by static and dynamic schedulers for different backlogs x_1 and deadlines w, $x_2 = 1$, $N = 6$, $p_e = 0.4$.

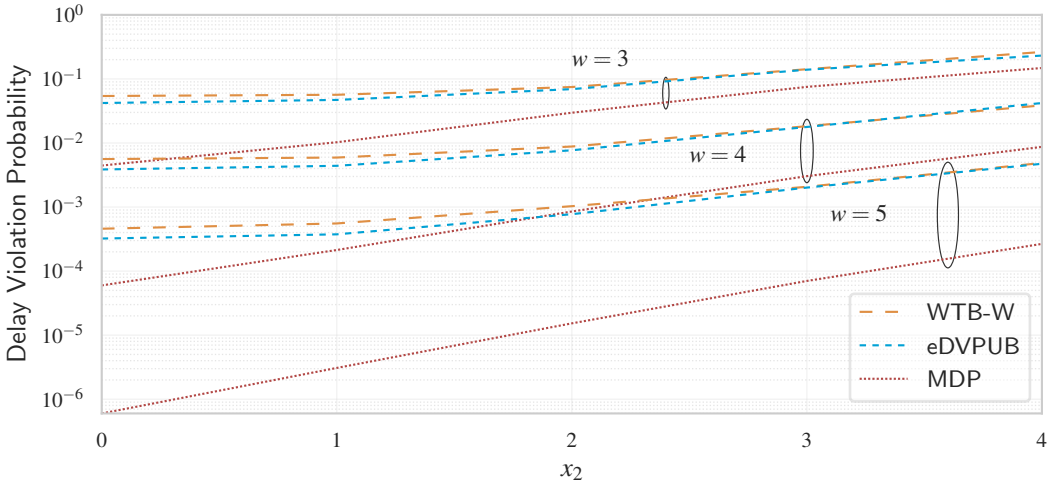

Figure 4.15: DVP achieved by static and dynamic schedulers for different backlogs x_2 and deadlines w, $x_1 = 1$, $N = 6$, $p_e = 0.4$.

the bigger the performance gap. Therefore, in order to fully benefit from up-to-date queue states, dynamic scheduling should be employed when the application requires very low values of DVP.

4.5 Concluding Remarks

In this chapter, several network-layer scheduling policies have been developed to support the transmission of individual messages of feedback-based applications. In contrast to the existing works that model the steady-state traffic of time-critical applications, this is achieved thanks to a transient model of the network, which is described in Sec. 4.2. The model characterizes the

random service offered by lossy transmitters over a two-hop network path in terms of delay and DVP. Thanks to the model, in Sec. 4.3, different scheduling policies have been developed to allocate network resources in order to minimize the DVP of packet in each feedback loop. This approach is different from existing methods as network resources are optimized to maximize the application performance instead of solely fulfilling its QoS requirements.

Furthermore, the impact of queue state information on the achievable DVP of static and dynamic scheduling policies has been investigated. Results have shown that exploiting information about the queue states leads to considerable improvements. On the one hand, initial queue states improve the achievable DVP by one order of magnitude with respect to an allocation that ignores them. On the other hand, the availability of per-frame queue states provides an improvement of several orders of magnitude. The improvement is particularly large for longer frames and higher deadlines, which shows that the presented dynamic scheduling methods do not only improve the allocation of slots between frames but make each allocation more efficient.

Thanks to these results, given the stochastic link-layer performances of IWSN, it is possible to allocate network resources in order to minimize the probability of exceeding the deadline imposed by the entire feedback control loop. In this way, IWSN transmissions can be fully optimized to support feedback-based applications despite random packet losses and delays caused by unreliable wireless links. The results of this chapter are further applied in Ch. 5 to investigate the performance of NCS in a real IWSN deployment.

Chapter 5

Impact of QoS Provisioning on Networked Control Systems

As shown in the previous chapters, by modelling the stochastic performance of wireless transmissions, it is possible to develop Radio Resource Management (RRM) schemes that fulfil desired link- and network-layer Quality of Service (QoS) requirements. In this way, the radio resources of Industrial Wireless Sensor Networks (IWSN) can be optimized to successfully convey the messages of Networked Control Systems (NCS). IWSN communication brings several advantages to NCS, such as ease of installation, maintenance, flexibility [101], mobility, and operation in hostile environments. In order to fully exploit these benefits at the application layer, it is necessary to characterize the impact of different network-layer QoS values on the Quality of Control (QoC), i.e., the performance of NCS. To achieve this, a thorough investigation of the QoC of feedback control loops subject to the constraints introduced by wireless transmissions is needed. This chapter provides a detailed investigation of the QoC of NCS in the presence of packet loss and delays caused by wireless transmissions.

Sec. 5.2 describes and evaluates the Linear Time-Invariant (LTI) model of a Networked Inverted Pendulum (NIP), an exemplary NCS that has a long tradition in the literature [73, 93] and industry [95]. Furthermore, it defines the QoC metrics used in this chapter to evaluate the performance of the NIP and evaluates them for different network-layer QoS performances. This is achieved by abstracting the network delays and packet loss in order to evaluate their effect on the NCS. While this is a popular choice in several works that investigated the impact of network imperfections on NCS [101, 108, 157], it does not provide insights in the performance of NCS with IWSN.

Sec. 5.1 provides a comprehensive overview of the works that investigated the impact of network imperfections on NCS, with a focus on Wireless Local-Area Network (WLAN) and IWSN technologies. Existing works that investigated WLAN and NCS [18, 136–138, 145],

however, cannot be applied to IWSN due to a different contention-based Medium Access Control (MAC) scheme and physical-layer performances. Furthermore, existing works that investigated the operation of NCS in IWSN [36, 100, 102, 104, 113, 114] did not investigate the network-layer distribution of delays in the IWSN nor the relationship between QoS and QoC. To solve these issues, in Sec. 5.3, based on the own publication "Delay-Reliability Model of Industrial WSN for Networked Control Systems" [3], the analytical QoS–QoC relationship of an IWSN and a simulated NCS is derived by means of a delay-reliability model. The model captures the impact of link- and network-layer QoS provided by the RRM schemes presented in this work on the operation of the NIP. In this publication, I have derived the mathematical delay-reliability model and characterized its relationship to QoC. In his Master's Thesis [120], Mr. Shantharam implemented the measurement of delays and QoC in a testbed and verified the model via experimental results.

Finally, Sec. 5.4 describes the evaluation of QoS and QoC for an experimental NCS platform based on the own work "NCSbench: Reproducible Benchmarking Platform for Networked Control Systems" [4]. Differently from state-of-the-art experimental works that focused on specific aspects of the NCS [17, 19, 26, 81], NCSbench provides a comprehensive performance characterization of all the components of an NCS in terms of delay, reliability, and QoC. NCSbench is an open-source wireless NCS platform that was developed in collaboration with Mr. Molinari and Mr. Music from the Control Systems Group, Technical University of Berlin, and, from the Technical University of Munich, Mr. Gallenmüller from the Chair of Network Architectures and Services and Mr. Ayan from the Chair of Communication Networks. Mr. Molinari and Mr. Music developed the control logic of the NCS, Mr. Gallenmüller formulated the NCS benchmarking methodology, while Mr. Ayan and I implemented the communication network and the measurement framework. In particular, I developed the time model of the delays in the control loop, defined the delay and reliability performance metrics, and implemented their measurement. Furthermore, I have analysed the measurements results and implemented their visualization in a real-time web-based Graphical User Interface (GUI), which was awarded with the "Best Demo Award" at the 17-th Institute of Electrical and Electronics Engineers (IEEE) Consumer Communications & Networking Conference. The main findings presented in this chapter are summarized and discussed in Sec. 5.5.

5.1 Related Work

5.1.1 Preliminary Research on NCS

Thanks to attractive application scenarios, wireless NCS have been largely investigated over the past decades [56, 101, 152, 156, 157]. Initial studies focused on intrinsic imperfections of control systems, such as plant delays arising from sensing and actuating the plant, and controller delays [152] arising from the computation of the control logic. When communication networks became more accessible, research works focused on the investigation of the network-induced effects on the control system [156]. The most critical problem introduced by the network is failure in delivering control messages caused by random delays and packet loss [101, 156]. Furthermore, the network introduces additional issues such as asynchronous operation of the nodes and time-varying sampling intervals [101, 156].

A large number of research works investigated network-induced effects abstracting them with arbitrary models of network delays and packet loss. In this way, the analysis, design, and optimization of NCS can be performed independently of the communication network. Existing models adopted Markov chains or Bernoulli random variables to model network delays and packet loss [56, 101, 156, 157], which were used to derive stochastic conditions on the stability of the NCS to design optimal controllers. The resulting control strategies can be classified as robust or adaptive [156]. Robust techniques counteract network effects, such as delay compensation and fault-tolerant techniques [56, 157], while adaptive controllers react to delays and packet loss that occur during the operation [156]. Exemplary NCS designs are given in [31, 77, 82, 87, 108, 140], where authors modelled the delays and packet loss of the network in order to control and evaluate NCSs. In all these works, however, a realistic model of the propagation environment and medium access of the network is not considered.

The nature of packet loss and delay is different in each communication network. Therefore, including a detailed model of the communication network is particularly important in the characterization of NCS performances. Initial studies of NCS and real networks were performed for *wired* communication technologies such as FieldBus, CAN, and Ethernet [152, 156]. Wired networks are not affected by packet loss and introduce quasi-deterministic network delays. Thus, research on wired NCS mainly developed real-time scheduling protocols to support multiple NCS over a shared, multi-hop network in the context of industrial applications [152, 156].

In the recent years, thanks to rapid advancements in wireless communication technologies, *wireless* NCS have been extensively investigated [101]. In particular, thanks to their popularity and availability, wireless NCS were investigated with WLAN and IWSN. WLAN adopts a contention-based MAC scheme. Therefore, research works that studied NCS and WLAN

focused on the evaluation of random delays caused by concurrent transmissions [24] and proposed techniques to achieve bounded communication delays. Authors in [136] proposed a deadline-aware backoff and re-transmission mechanism. In [149], a framework for the cross-layer design of contention-based NCS was presented. In [18], authors optimized the re-transmission limit of WLAN according to the NCS performance, while authors in [138] implemented a model-based predictive control over a relay node. A popular approach to support real-time communication in WLAN is to introduce a centralized contention-free medium access. In [138], a network coordinator centrally allocates transmission opportunities and redundancy mechanisms in order to fulfill the desired real-time QoS requirements. The combination of a Time Division Multiple Access (TDMA) scheme with the WLAN physical layer was explored in [21, 88, 137, 145]. Although several techniques were developed for NCS and WLAN, their results cannot be directly applied to IWSN for different reasons. The availability high transmission powers, multiple modulation orders, and coding schemes provide considerably better performances to WLAN in terms of coverage, throughput, and reliability. This is fundamentally different in IWSN, where lower transmission powers and data-rates limit the communication performance and increase the effect of packet loss. Existing works related to IWSN and NCS are discussed in details in the next section.

5.1.2 IWSN and NCS

Network delays and packet loss of IWSN have been investigated by several works in the literature. The available works are clustered into *network-related* and *control-related* approaches. *Network-related* approaches abstract NCS as real-time traffic. Therefore, they allocate network resources to fulfil the delay and reliability requirements of real-time flows in a multi-hop network and do not consider the interaction with the NCS. Authors in [48, 151, 6] investigated re-transmission strategies for IWSN to compensate packet loss. The methods are evaluated in terms of delay and reliability, but no analytical model is provided. Furthermore, several scheduling algorithms were proposed [37, 62, 79, 116, 2]. Liu et al. [79] constructed schedules for control loops and Saifullah et al. [116] solved an optimization problem to allocate resources fulfilling a desired delay-bound. In both cases, however, packet loss was not considered. Authors in [62] and in [37, 2] formulated scheduling problems to achieve maximum reliability within a delay bound and minimum latency within a reliability bound. These methods, however, did not characterize the distribution of delays and the reliability of the network. Authors in [91, 115] investigated resource allocation methods to provide bounded communication delays in the presence of packet loss. Munir et al. [91] characterized the network delays observing bursts in packet drops, while Saifullah [115] modelled channel contention and transmission conflicts between different flows in the network. Their methods,

however, only covered the uplink communication between sensor and controller and did not characterize the delay distribution of packets. In [147], authors characterized the worst-case delay achieved by multiple real-time flows under the Earliest Deadline First scheduling policy. Also in this case, however, the individual delay of each NCS message traversing the network was not characterized. A common drawback of existing network-based solutions is that network resources are allocated without evaluating the performance of the NCS.

To closely investigate the impact of IWSN on the NCS, *control-related* approaches study the joint design of IWSN and NCS. Park et al. [102] characterized the delay and reliability of an IWSN subject to packet loss and delays of contention-based MAC schemes. A different approach is studied by authors in [36, 100, 104, 113, 114], which analyzed packet loss and delay for a TDMA medium access. Sadi et al. [114] optimized the network transmission powers to minimize energy expenditure and maximize adaptability. Furthermore, they adapted the transmission rate and scheduling algorithm [113] to fulfill the stochastic Maximum Allowable Transmission Intervals requirement of NCS. Demirel et al. [36] investigated the joint optimization of TDMA scheduling and NCS control in a lossy multi-hop IWSN. All these works, however, optimize the network parameters to allow the optimal control of NCS, however, the network QoS performance are not characterized. Park et al. [100] computed optimal schedules for contention-based and contention-free MAC schemes. Pesonen et al. [104] investigated the operation of different TDMA schedules with frequency hopping, and developed a Markov Chain to model the network reliability. Their approach, however, did not provide an analytical characterization of the delay and reliability of the network, which are only observed via simulations. With the current state of the art, a major challenge still exists in modelling the delay and reliability of IWSN to achieve a desired QoC of the NCS. In Sec. 5.3.1, this gap is filled with the introduction of a delay-reliability model of the messages in each control loop for a two-hop IWSN, which is based on the own work “Delay-Reliability Model of Industrial WSN for Networked Control Systems” [3].

5.1.3 Experimental Works

The vast majority of the existing works characterize the effect of delay and packet loss on QoC via an analytical model of the NCS. However, experimental NCS are subject to additional effects that arise from the *implementation* of the control system and communication network, which must also be characterized. As stated by Lu et al. [80], conveying full-scale practical research with a real implementation of a Cyber-Physical Systems (CPS) is a difficult task due to the complexity and the replicability of the experimental platforms. Therefore, research work in the field of NCS conducting experimental studies is rather limited. Zhang et al. [144] and Chamaken et al. [26] implemented a hybrid setup of an NCS combining hardware in the loop,

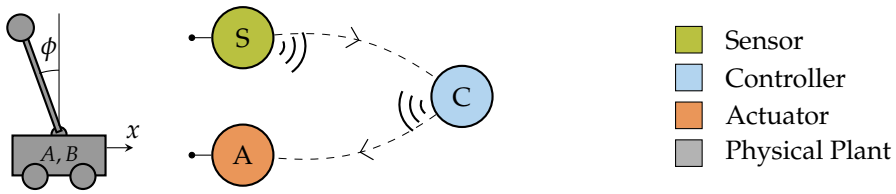

Figure 5.1: Architecture of a two-hop wireless NIP.

i.e., a simulation of the plant dynamics, with a real network. On the contrary, experimental results of Kawka et al. [70] and Eker et al. [43] used the Network-in-the-Loop approach, i.e., a simulated network with real hardware as a control system. Different research works provide prominent examples where the complete NCS consists of real hardware [17, 19, 39, 81]. Drew et al. [39] proposed an NCS design that takes into account network delays and packet dropouts, and evaluated it in an experimental scenario. They showed that, by optimizing the network-aware control logic, their system performed better than the conventional network-unaware controllers. Bachhuber et al. [17] conducted an end-to-end latency analysis of a vision-based NCS. Baumann et al. [19] presented measurement results of inverted pendulum operating in a multi-hop wireless network. Mager et al. [81] proposed a reliable multi-hop wireless protocol that enables the remote control of multiple experimental inverted pendulums. Existing experimental works represent initial investigations in the analysis of communication imperfections of real NCS. However, they only presented distinct solutions to deal with specific problems of each implementation and did not provide a full characterization of the components of NCS in terms of delay and reliability. To address this issue, in Sec. 5.4 the experimental characterization of the QoS–QoC relationship of an NCS platform is performed based on the own work "NCSbench: Reproducible Benchmarking Platform for Networked Control Systems" [4]. This is achieved by modelling the delay of each component of the platform in each step of the control loop and by relating measurements of delays to QoC.

5.2 NCS Model and QoC

The NCS considered in this work is a NIP. The Inverted Pendulum consists in a vertical bar mounted on a cart that moves using two wheels. The problem of controlling an Inverted Pendulum has a long tradition in literature [73, 93] and industry [95]. In an NIP, a remote controller has the goal of vertically suspend the Inverted Pendulum with appropriate movements of the cart wheels. As represented in Fig. 5.1, each sampling interval, a sensor measurement is sent to the controller that computes a control command for the actuator to steer the physical plant [101].

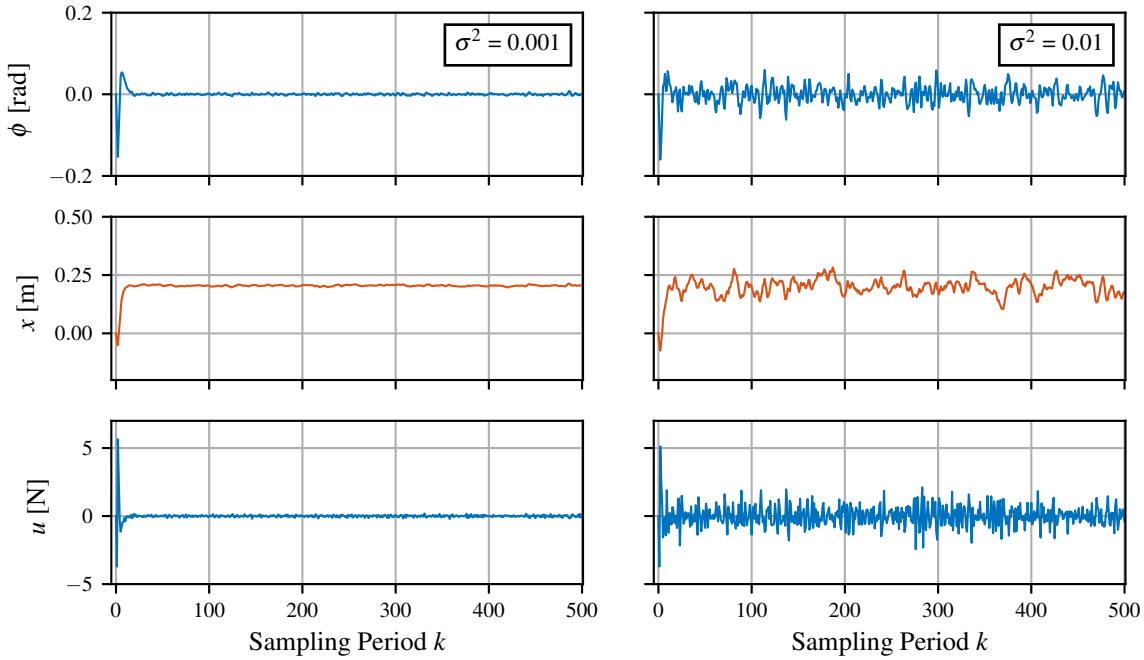

Figure 5.2: Time evolution of the Inverted Pendulum's vertical angle (top), position (middle), and actuator (bottom) for system noise values σ^2 equal to 10^{-3} (left) and 10^{-2} (right) on each component of the system state.

The dynamic of the Inverted Pendulum is described by the following linear discrete-time state equation

$$\mathbf{x}_{k+1} = A\,\mathbf{x}_k + B\,u_k + \mathbf{w}_k, \tag{5.1}$$

for time $k \in \mathbb{N}_+$ and with initial condition x_0. The state vector $\mathbf{x}_k = [x_k, \dot{x}_k, \phi_k, \dot{\phi}_k]^\top$ describes the evolution of the cart's position x and the vertical angle of the pendulum ϕ represented in Fig. 5.1. The system matrices $A \in \mathbb{R}^{4x4}$ and $B \in \mathbb{R}^{4x1}$ are obtained by linearizing and discretizing the dynamical model of the pendulum as shown in [83]. The white Gaussian noise $\mathbf{w}_k \in \mathbb{R}^{4x1}$ has zero mean and covariance $W \succ 0$, which represents the linearization error and modelling limitations of the real system.

We assume that the sensor can directly obtain measurements of the system's state and that no measurement error occurs. The measured state is used by the controller to compute

$$u_k = -K\,(\mathbf{x}_k - \mathbf{x}_r)\,, \tag{5.2}$$

where $\mathbf{x}_r = [0, 0, 0, 0]$ is the control reference, $u_k \in \mathbb{R}$ is the actuation command for the motors rotating the wheels, and $K \in \mathbb{R}^{1x4}$ is the control gain that is obtained solving the algebraic Riccati equation according to the Linear Quadratic Regulator (LQR) control algorithm.

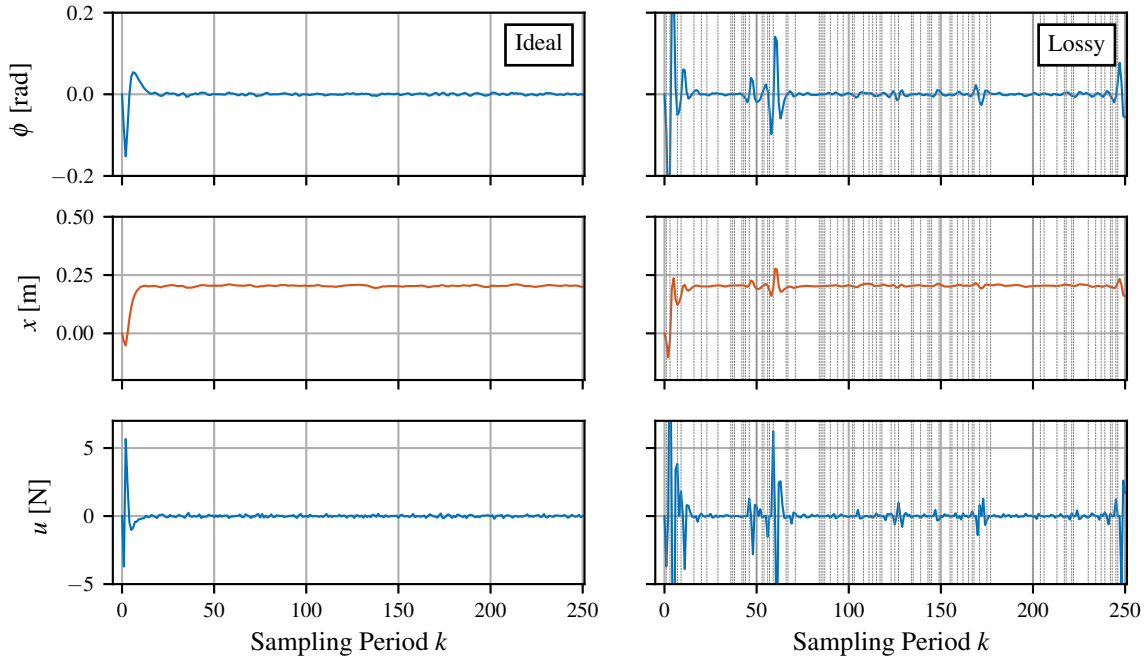

Figure 5.3: Time evolution of the Inverted Pendulum's vertical angle (top), position (middle), and actuator (bottom) in absence (left) and presence (right) of packet loss.

An exemplary run of the Inverted Pendulum is provided in Fig. 5.2 for a sampling interval $T_S = 100$ ms and system parameters

$$A = \begin{bmatrix} 1 & 9.91 \cdot 10^{-2} & 1.36 \cdot 10^{-2} & 4.50 \cdot 10^{-4} \\ 0 & 9.82 \cdot 10^{-1} & 2.79 \cdot 10^{-1} & 1.36 \cdot 10^{-2} \\ 0 & -2.32 \cdot 10^{-3} & 1.16 & 1.05 \cdot 10^{-1} \\ 0 & -4.74 \cdot 10^{-2} & 3.28 & 1.16 \end{bmatrix}, \tag{5.3}$$

$$B = \begin{bmatrix} 9.09 \cdot 10^{-3} & 1.82 \cdot 10^{-1} & 2.32 \cdot 10^{-2} & 4.74 \cdot 10^{-1} \end{bmatrix}^\top, \tag{5.4}$$

$$K = \begin{bmatrix} -18.2 & -11.5 & 41.8 & 8.04 \end{bmatrix}, \tag{5.5}$$

$$\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top, \tag{5.6}$$

for a system noise with diagonal covariance matrix W and values σ^2 equal to 10^{-3} and 10^{-2} on its diagonal. The evolution of the system states converges to the desired reference and the actuation command converges to zero. A higher value of system noise, shown on the right side of Fig. 5.2, has a strong impact on the performance of the NIP, introducing oscillations around the reference values.

When packets are correctly delivered, the physical system is flawlessly controlled. However, if packets are delayed or lost, the control loop is interrupted and the physical system deviates from the desired trajectory, potentially becoming unstable. This effect is shown by

σ^2	QoC_ϕ	QoC_x	QoC_u
0.001	1.423	3.124	42.420
0.004	3.752	5.627	129.106
0.007	6.509	9.062	215.674
0.010	8.511	11.659	302.859

Table 5.1: QoC metrics for different values of system noise σ^2, $T_E = 500$.

	QoC_ϕ	QoC_x	QoC_u
Ideal	1.022	2.183	29.845
Lossy	3.046	2.750	117.780

Table 5.2: QoC metrics in absence and presence of packet loss, $T_E = 250$, $\sigma^2 = 0.001$.

the time evolution of the NIP on the right side of Fig. 5.3. The effect of packet loss is obtained by randomly dropping actuation commands according to a Bernoulli r.v. where actuation messages are received with probability 0.7. When the actuation command is not received, the actuator assumes a default value, in this case 0, causing a divergence between the system's dynamic and the controlled trajectory.

5.2.1 QoC Metrics

In order to evaluate the performances of NCS, different QoC metrics can be defined. QoC metrics are used to benchmark different configurations of the NCS and guide the design of the control logic. Furthermore, they can be used to quantify the effect of network imperfections on the operation of the control loop. When the NCS in not subject to communication errors, the dynamic of the system states is close to the desired reference and the effort of the actuators is minimal. However, losing actuation messages causes large deviations in the system's trajectory and, in the worst case, instability.

To capture these effects, QoC metrics are generally defined with respect to the system states $\mathbf{x}_k$ and the actuation command u_k. For an experiment duration of T_E sampling periods, the *performances* of the NIP are quantified by the Integrated Absolute Error (IAE) of the system states and the total control effort

$$QoC_x = \sum_{k=1}^{T_E} \mid x_k - x_{ref} \mid, \tag{5.7}$$

$$QoC_\phi = \sum_{k=1}^{T_E} \mid \phi_k - \phi_{ref} \mid, \tag{5.8}$$

$$QoC_u = \sum_{k=1}^{T_E} \mid u_k \mid . \tag{5.9}$$

Smaller values of QoC_x and QoC_ϕ correspond to smaller oscillations of the pendulum, thus better performance. Similarly, a smaller QoC_u indicates smaller movements of the cart and hence a higher control performance. The QoC values of the evolutions shown in Fig. 5.2

and 5.3 are summarized in Tab. 5.1 and 5.2, respectively. As shown in Tab. 5.1, higher values of system noise σ^2 lead to higher values of QoC. Each QoC metric shows different magnitudes, which are characteristic of the measured physical property. For instance, QoC_ϕ is smaller than QoC_x as the vertical angle is subject to smaller deviations than the position of the NIP. Tab. 5.2 quantifies the impact of packet loss on the QoC for a probability of losing actuation commands equal to 0.7.

In some cases, the impact of system noise and network imperfections can lead to very large deviations in the trajectory of the system states, thus to the instability of the NCS. This effect is captured by an additional QoC metric, the *stability* metric QoC_S. QoC_S is defined as the probability that the pendulum is stable for the entire experiment duration. In this work, it is assumed that the pendulum is stable if the trajectory of the vertical angle is bounded by a maximum angle ϕ_T, i.e., $|\phi_k| < \phi_T \, , \forall k$. The value of ϕ_T depends on the combination of the physical characteristics of the Inverted Pendulum, such as the precision of sensor and actuators, and on the employed control logic. By performing multiple experiments, the stability QoC can be calculated as

$$QoC_S = \mathrm{P}[\mid \phi_k \mid < \phi_T] = \frac{N_S}{N_E}, \tag{5.10}$$

where N_E is the total number of experiments and N_S is the number of runs where the pendulum is stable. The impact of the IWSN communication on the different QoC metrics is evaluated in details in the next section.

5.3 Model of Network-Layer QoS and QoC

QoC is severely affected by network imperfections, which cause the interruption of the control loop. For an NCS described by Eqs. (5.1) and (5.2), the effect of network delays and packet loss can be evaluated via the Loop Success Probability (LSP), i.e., the probability that sensor and controller messages are correctly delivered at each control loop. In order to achieve a desired trajectory of the dynamical system, actuation commands have to be applied every sampling period T_S. Thus, packets must be delivered to both controller and actuator within T_S. Due to random packet drops and delays, it can happen that the actuation command does not reach the plant on time. Therefore, LSP is defined as the probability P_{LS} that the End-to-End (E2E) delay of a control packet D_{E2E}, generated at the sensor and received at the actuator after processing at the controller, is smaller than T_S

$$P_{LS} \triangleq \mathrm{P}[D_{E2E} < T_S]. \tag{5.11}$$

The definition of LSP incorporates the delay and reliability of the network and the requirements of the control loop.

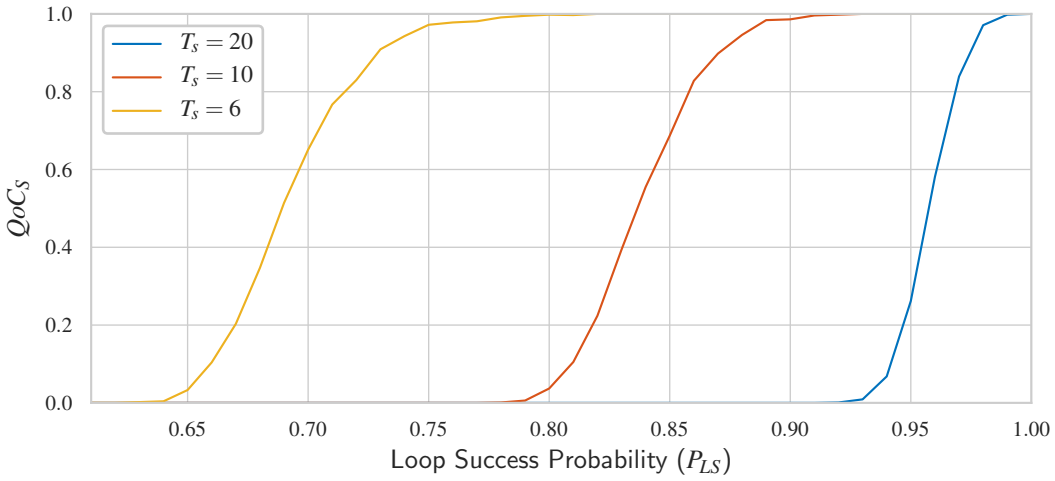

Figure 5.4: Impact of LSP on the stability of the NIP (QoC_S) for different sampling periods T_S.

In Figs. 5.4 and 5.5, the impact of different P_{LS} values on the different QoC metrics is evaluated via extensive simulations of the NIP. For every simulation, the NIP follows the same values of Sec. 5.2, with $\sigma^2 = 10^{-3}$ and $\phi_T = 30°$. In each sampling period, actuation messages are received with probability P_{LS} and lost with probability $1 - P_{LS}$. Furthermore, different sampling periods T_S have been evaluated. For each T_S, the system matrices A and B are obtained discretizing the continuous-time model of the pendulum [83] and a new LQR control gain K is calculated.

Fig. 5.4 shows the impact of P_{LS} on QoC_S for systems with different T_S. Each curve is obtained repeating 10^2 simulations of $T_E = 10^5$ sampling periods. The results show that P_{LS} has a strong impact on the stability of the system, that is, the stability reduces when the P_{LS} decreases. All systems experience a transitional region for stability for different intervals of P_{LS}. Systems with shorter sampling periods tolerate lower P_{LS} as they can control the dynamics of the system more often. The impact of P_{LS} on the QoC_ϕ is evaluated only for P_{LS} values that achieve stability for the entire experiment duration. Thus, in Fig. 5.5, QoC_ϕ is shown for $P_{LS} \geq 0.9$ and $T_S = 10$. For each P_{LS} value, an experiment with 10^5 sampling periods is repeated 10 times. All the QoC metrics follow the same trend, increasing the P_{LS} leads to smaller QoC values. Also in this case, the boxplots reflect the variability of the underlying physical parameters, showing an increased variability for QoC_x with respect to QoC_ϕ and QoC_u.

Although LSP enables the evaluation of network imperfections on QoC, it does not describe the effect of IWSN communication. To fill this gap, the LSP must be related to the performance achieved by the link- and network-layer QoS provisioning schemes described in Chapters 3 and 4. This is achieved in Sec. 5.3.1 via the definition of a packet-based delay-reliability model of a two-hop IWSN [3]. The model provides the interconnection between

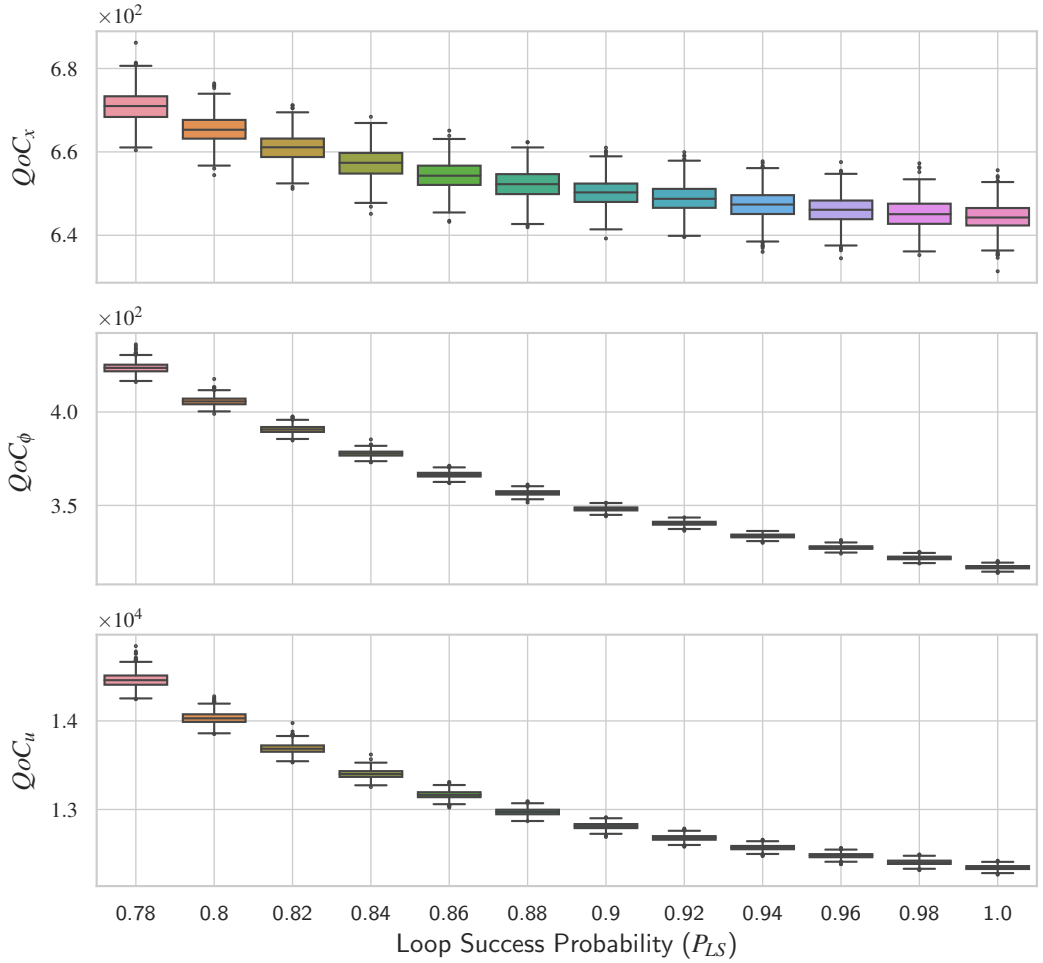

Figure 5.5: Impact of Loop Success Probability on the NCS performance (QoC_ϕ) for $T_S = 10$.

IWSN communication parameters and QoC, and is used to define their operating regions. In fact, it models the exact network-layer QoS performance of messages in a control loop and relates it to LSP. In Sec. 5.3.2, the delay-reliability model is validated via measurements of an IWSN testbed transmitting the messages of the NIP described in Sec. 5.2.

5.3.1 Packet-Based Delay-Reliability Model

The LSP determines the QoC and the delay-reliability performance achieved by lossy wireless links in the control loop. In this section, the analytical model of LSP is derived for a two-hop IWSN by deriving the exact distribution of delays in the network. The communication resources of the IWSN are modelled according to a TDMA medium access scheme with frequency hopping, following the model adopted in the previous chapters and by the IEEE Std. 802.15.4e Time Slotted Channel Hopping (TSCH). Each TDMA frame consists of $2N$ time slots and an average Packet Error Rate (PER) p_e, which is determined by a specific

hopping sequence in a given propagation environment. The time slots are equally shared between the two transmitters – the first N slots are allocated to the sensor and the remaining N to the controller. IWSN operate at low transmission powers in an interfered environment, therefore they are subject to packet loss. To increase the chance that a message is received, packet re-transmissions are needed. Let R be the maximum number of transmission attempts at both sensor and controller. If a packet exceeds this limit, it is dropped and no additional transmissions are attempted. Moreover, packets that exceed the sampling period T_S carry outdated information that cannot be used by the NCS defined in Sec. 5.2. Therefore, in our system, they are automatically discarded by each transmitter. The interconnection between the parameters of the delay-reliability model and the RRM schemes discussed in the previous chapters is detailed at the end of this section.

The E2E delay D_{E2E} of the control loop is the sum of delays from the Sensor-to-Controller (S2C) and Controller-to-Actuator (C2A) links

$$D_{E2E} = D_{S2C} + D_{C2A}. \tag{5.12}$$

The S2C and C2A delays are constituted by a stochastic buffering delay D_B and a stochastic transmission delay D_{TX}. The C2A delay additionally contains the deterministic controller processing delay D_C

$$D_{S2C} = D_B^S + D_{TX}, \tag{5.13}$$

$$D_{C2A} = D_B^C + D_{TX} + D_C. \tag{5.14}$$

As only one packet is present in the network each sampling period, buffering delay exclusively arises from the waiting time between the generation or reception of a packet and its transmission according to the schedule. The transmission delay is caused by re-transmissions in the case when packet loss occurs. For a sensor measurement generated at slot i, the buffering delays of sensor and controller are deterministic

$$D_B^S(i) = \begin{cases} 0, & \text{if } 1 \leq i < N, \\ 2N - i, & \text{if } N \leq i \leq 2N, \end{cases} \tag{5.15}$$

$$D_B^C(i) = \begin{cases} N - i, & \text{if } 1 \leq i < N, \\ 0, & \text{if } N \leq i < 2N, \\ N, & \text{if } i = 2N. \end{cases} \tag{5.16}$$

Correspondingly, the transmission delay can be modelled as a geometric r. v. with average link success probability as parameter, i.e., $D_{TX} \sim \mathrm{Ge}\,(1 - p_e)$. That is, the probability that the transmission delay of a packet lasts x slots starting from slot j is equal to

$$\mathrm{P}[D_{TX} = x \mid j] = p_e^{r(x,j)-1}\,(1 - p_e)\,, \tag{5.17}$$

where $r(x,j)$ represents the number of attempts needed to achieve a transmission delay of x slots from slot j.

$$r(x,j) = \begin{cases} x, & \text{if } 0 < x \leq N_R(j), \\ x-(k+1)N, & \text{if } N_R(j)+(2k+1)N < x \leq N_R(j)+2(k+1)N, \end{cases}$$
$$k = 0,1,\ldots,k_M, \quad k_M(j) = \left\lfloor \frac{R-N_R(j)-1}{N} \right\rfloor, \tag{5.18}$$

where the delay interval in Eq. (5.18) defines the support of D_{TX}, and $N_R(i)$ is the number of available slots in the frame at the beginning of the transmission,

$$N_R(i) = \begin{cases} N-j \bmod 2N & \text{for the sensor,} \\ 2N-j \bmod 2N & \text{for the controller.} \end{cases} \tag{5.19}$$

The E2E delay is the sum of the buffering and transmission delays of the S2C and C2A links, which are not mutually independent. In particular, the S2C transmission delay depends on the generation of the sensor measurement, and the C2A transmission delay on both S2C transmission delay and buffering. In our analysis, the stochastic arrival process A is considered, which is characterized by arbitrary arrival probabilities for every slot in the TDMA frame. Thus, the E2E delay distribution can be calculated via conditional probabilities of the three random processes: measurement arrival, S2C transmission, and C2A transmission.

$$\begin{aligned} \mathrm{P}[D_{E2E}=y] &= \sum_{i=1}^{2N}\sum_{x=0}^{y} \mathrm{P}[A=i]\,\mathrm{P}[D_{S2C}=x \mid i]\,\mathrm{P}[D_{C2A}=y-x \mid i+x] \\ &= \sum_{i=1}^{2N}\sum_{x=0}^{y} \mathrm{P}[A=i]\,\mathrm{P}[D_{TX}=x-D_B^S(i) \mid i+D_B^S(i)] \\ &\qquad \mathrm{P}[D_{TX}=y-x-D_B^C(z)-D_C \mid z+D_B^C(z)] \\ &\overset{*}{=} \sum_{i=1}^{2N}\sum_{x=0}^{y} \mathrm{P}[A=i]\ (1-p_e)^2\, p_e^{r\left(x-D_B^S(i),i+D_B^S(i)\right)-1} \\ &\qquad p_e^{r\left(y-x-D_B^C(z)-D_C,z+D_B^C(z)\right)-1}, \end{aligned} \tag{5.20}$$

where $z = i+x+D_C \bmod 2N$, and $*$ indicates a summation that only considers the terms where the support of D_{TX} for sensor and controller are defined. Therefore, the E2E delay distribution consists of a sum of exponentials weighted by the measurement arrival distribution of the sensor.

Given a sampling period T_S and the TSCH medium access parameters N, p_e and R, the P_{LS} is calculated following Eq. (5.11) as the cumulative distribution of D_{E2E}

$$P_{LS}(T_s, N, p_e, R) \overset{*}{=} \sum_{y=1}^{T_S-1} \sum_{i=1}^{N} \sum_{x=0}^{y} \mathrm{P}[A = i]\ (1-p_e)^2 \\ p_e^{r(x-D_B(i),i)+r(y-x-D_B(i+x)-D_C,i+x)-2}, \tag{5.21}$$

where the support of D_{TX} for sensor and controller is defined.

The delay-reliability model is flexible enough to accommodate the QoS provisioning schemes described in the previous chapters. The link-layer RRM scheme described in Ch. 3 can be applied to the delay-reliability model as follows. When individual Resource Blocks (RBs) are scheduled, the TDMA frame of the delay-reliability model operates as an outer TDMA frame where each "slot" represents an inner TDMA frame used by the link-layer scheduler. In this way, in the outer frame, time resources are distributed between the two transmitters taking into account the queuing effects of the two-hop network, while the resources in the inner frame are scheduled to achieve arbitrary PER levels.

Furthermore, the delay-reliability model can be applied to the network-layer RRM schemes presented in Ch. 4. In fact, similarly to the Delay Violation Probability (DVP)-based schedulers, the model considers the performance of each individual control packet until the deadline imposed by the NCS. Although some differences are present, they can easily be included in the delay-reliability model. First, in Ch. 4, the application deadline is always a multiple of the TDMA frame size. The delay-reliability model is able to capture network delays for each slot, thus, this condition is satisfied restricting the control system to sampling periods *Ts* that are integer multiples of the TDMA frame length. Second, in Ch. 4, packets are always re-transmitted. Although the delay-reliability model considers a maximum number of re-transmissions, it can be still applied to this scenario. In fact, in order to include this effect in the model, it is sufficient to select a number of re-transmission that is large enough to allow each transmitter to re-transmit each packet until the deadline. Finally, the fixed allocation of slots to both transmitters considered by the delay-reliability model can be relaxed. Although this allocation corresponds to the optimal choice when the initial queues are empty (cf. Ch. 4), when initial buffer is present, the delay-reliability model can be extended introducing new variables describing the asymmetric allocation of TDMA slots at each frame until the deadline.

5.3.2 Experimental Model Validation Results

In order to validate the delay-reliability model and verify its relationship to QoC, an IWSN testbed was implemented to transmit the messages of an emulated NIP. The emulated pendu-

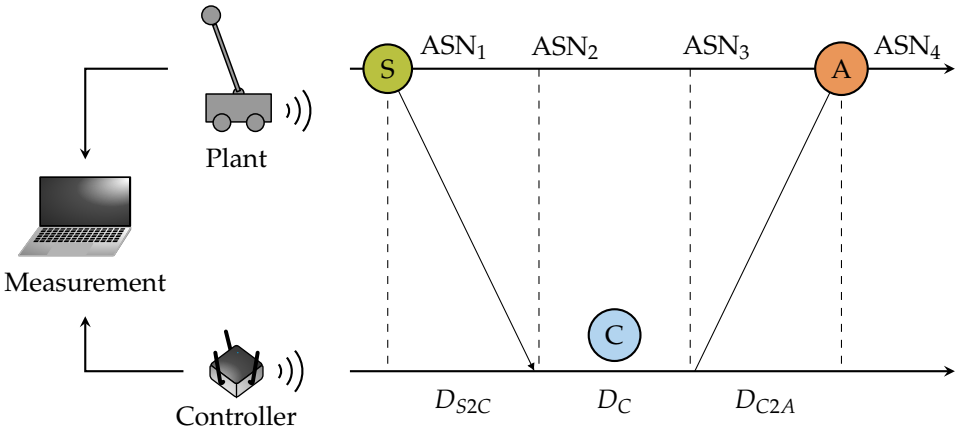

Figure 5.6: Measurement setup and model of the network delays.

lum follows the dynamical model described in Sec. 5.2, and performs sensing, control, and actuation following the same timing of a physical Inverted Pendulum, but in software. The emulation allows to capture the behaviour of the system even at border conditions where a physical pendulum may not provide accurate results. The Zolertia RE-Mote™ platform is used for implementing the IWSN. The devices execute the Contiki-NG firmware, which implements the IEEE Std. 802.15.4e TSCH MAC layer. The NullNet network configuration of Contiki-NG was selected, which provides a transparent network layer with no functionalities.

Fig. 5.6 depicts the experimental setup used for measurements. The sensor and controller functions are implemented, respectively, on the plant and controller devices, which transmit the outcome of the state evolution and control calculation of Eq. (5.1) and (5.2). As shown in Fig. 5.6, the devices are connected to a measurement computer that collects, for each packet, timestamps of transmission and reception in terms of Absolute Sequence Number (ASN), the global time provided in TSCH. Each slot has a duration of 10 ms. For every control loop, four ASNs are recorded. ASN 1 and 2 are used to calculate the S2C delay, while 3 and 4 the C2A delay. Furthermore, ASNs 2 and 3 are used to measure the processing delay of the controller, while ASNs 1 and 4 the E2E delay. Measurements are affected by packet loss introduced by a common office wireless environment. The devices are transmitting at a distance of 1 m, with a transmission power of -5 dBm, and perform frequency hopping on channels 15, 20, 25, 26 of the 2.4 GHz Industrial Scientific and Medical (ISM) band, which are moderately affected by interference.

Fig. 5.7 validates the S2C, C2A, and E2E delay distributions of Sec. 5.3.1 with measurements of the implemented IWSN. The nomenclature of a schedule is such that NX represents a schedule where both plant and controller are allocated X slots each. The results are shown for different schedules, $R = 2$ with an average link PER of $p_e = 0.08$. The sampling period

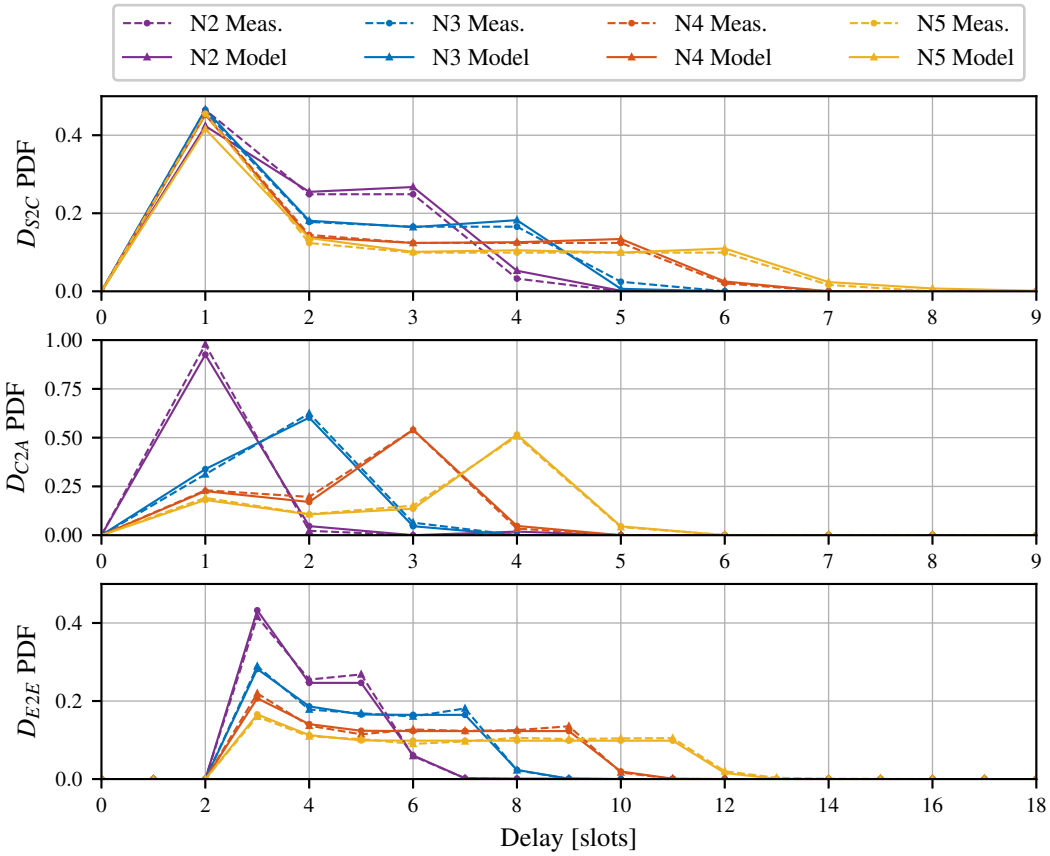

Figure 5.7: S2C (left), C2A (centre) and E2E (right) analytical and measured delay distributions of different schedules for $p_e = 0.08$ and $R = 2$.

of the inverted pendulum is equal to $T_S = 10$ slots and sensor's measurement arrivals are uniformly distributed in the TDMA frame, i.e., $\mathrm{P}[A = i] = T_S^{-1}, \forall i = 1, \ldots, T_S$. Fig. 5.7 shows that short schedules lead to distributions that achieve smaller delays compared to large ones. For every configuration, the measurements follow the analytical curves, proving that the developed analytical model precisely characterizes the delays of the TSCH medium access.

Fig. 5.8 compares the P_{LS} achieved by measurements with the analytical model, for different p_e values. Also for this case, the analytical P_{LS} corresponds to the one achieved by the real system. While schedules $N2$, $N3$, $N4$ achieve a P_{LS} at which the Inverted Pendulum is stable for the entire experiment, $N5$ operates in a condition where the pendulum is unstable. Therefore, in order to characterize the delays of $N5$, measurements are collected for a stable pendulum with $T_S = 20$ slots, which achieves an experimental $P_{LS} = 0.99$, and numerically evaluated for an E2E delay of 10 slots to enable comparison with the other schedules.

The experimental validation is concluded by comparing the QoC_ϕ, i.e., the position of the vertical angle, of the simulations with the one achieved by the emulated pendulum in the IWSN testbed. Fig. 5.9 shows QoC_ϕ for different values of P_{LS} obtained performing

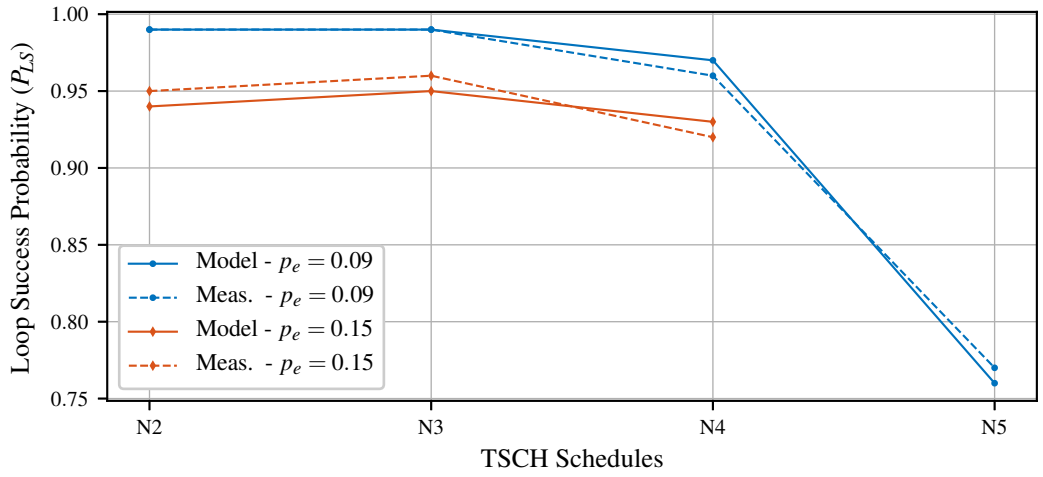

Figure 5.8: Analytical and measured LSP achieved by different schedules for two p_e levels and $R = 2$.

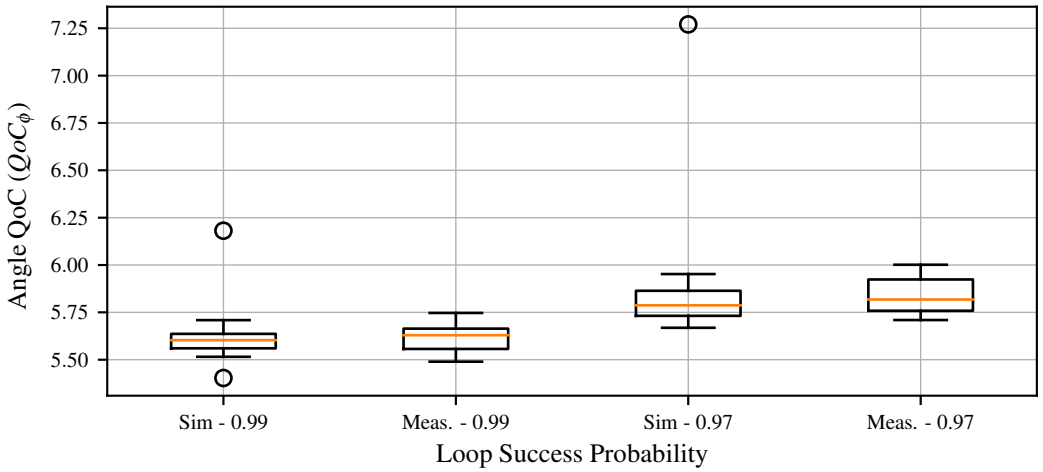

Figure 5.9: Comparison of QoC_ϕ between the simulated and emulated pendulum achieving the same LSP.

10 experiments of 2200 sampling periods. As expected, for a given P_{LS} achieved by the network, the pendulums achieve the same QoC_ϕ, proving that P_{LS} can be used to predict the QoC_ϕ performance. In the experimental evaluation QoC_S was not considered. In fact, the instability of the NCS makes it difficult to perform sufficiently long experiments that lead to comparable p_e. Furthermore, $QoC_S = 1$ is a necessary condition for the correct operation of the real system, which can be taken into account configuring the medium access according to the proposed delay-reliability model.

Thanks to the results achieved by the experimental evaluation, it is possible to conclude that the delay-reliability model precisely describes the experimental delays of a two-hop TSCH IWSN, and, thanks to its precise characterization of the LSP, it provides a direct relationship between QoC and the network-layer QoS of the IWSN.

5.4 Relationship Between QoS and QoC in Experimental NCS

Although a QoS–QoC relationship can be obtained modelling the network delays and via simulations of the NCS, only experiments of a real NCS can provide a comprehensive characterization of all the effects that influence its performance. An experimental NCS includes major modifications to the analytical model introduced in Sec. 5.2. In fact, hardware and software effects arise from a specific implementation of the control system and communication network and influence the system delays. In particular, delays and jitter are introduced by physical sensors during the acquisition of measurements and by actuators applying the control commands. Further delays are introduced by the operating system when computing the control signal and when sending or receiving messages.

Characterizing the QoC for a real NCS is challenging, as the amount of delay introduced by these effects is random in nature and does not only depend on the control logic or the communication network, but also on its implementation and the experimental environment. In fact, QoC measurements are affected by software and hardware elements as well as external factors that characterize the experiment, such as the physical and wireless environment, which must be documented in order to compare measurements across multiple experiments. To tackle these problems, a novel open-source[1] NCS benchmarking platform is presented in Sec. 5.4.2 based on the own work in [4] that allows the characterization of all the delays in an experimental NCS and their comparison across multiple experiments. The platform is based on the architecture of an experimental NCS, described in Sec. 5.4.1, which describes all its components and their interaction during the operation. Thanks to the architecture, all the delays arising in an experimental NCS are defined and allow the full characterization of QoS and QoC of the platform. Finally, the experimental relationship between QoC and QoC is presented in Sec. 5.4.3, where the distribution of delays in the NCS is characterized and related to corresponding QoC values.

5.4.1 NCS Architecture and Delay Analysis

In order to analyze the complex interplay of the components of a real NCS, a model of the experimental NCS architecture is derived and represented in Fig. 5.10. The architecture is composed of several software and hardware elements organized according to three domains [78]: control, computation, and communication. The set of elements composing the *control system* is twofold. On one side, the plant, i.e., the robot, mounts sensors and

[1] The source code is available at: https://github.com/tum-lkn/NCSbench.

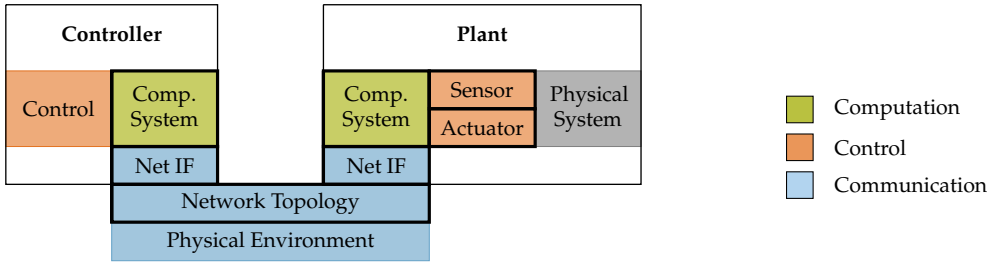

Figure 5.10: Architecture of experimental NCS. The control, computation, and communication domains are represented by different colours. Every box represents a component of the architecture, boxes surrounded by thick black contours identify hardware elements.

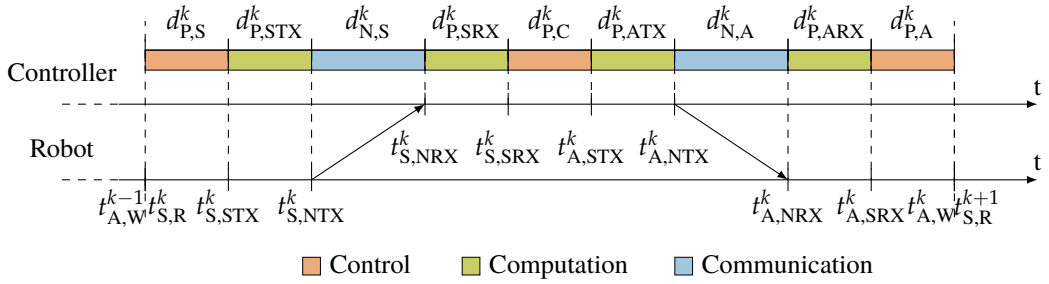

Figure 5.11: Time evolution of an NCS control loop. The timestamps of the processing (P) and networking (N) delays belonging to the control, computation, and communication domains are depicted.

actuators capable of sensing the physical system and executing the actuation commands respectively. On the other side, the controller, which is detached from the plant, receives the sensors readings, executes the control logic, and transmits instructions to the actuator. Two different *computing systems* provide computing power and access to the network interfaces to both controller and plant. The interconnection of the control application with the network interface is achieved with the implementation of the upper-layer protocols of the Open Systems Interconnection (OSI) communication stack. The *communication network* physically interconnects the computing system of the controller with the computing system of the plant and enables the flow of information between them. In the architecture, it defines the lower-layers of the OSI communication stack.

When all the components of the NCS architecture are interconnected, information regularly flows between the plant and the controller over the communication network. In particular, every sampling period, the sensor measures the state of a plant and sends it to the controller, which computes and sends a command to the actuator that steers the plant. The time evolution of the k-th sampling period is represented in Fig. 5.11. At time $t^k_{S,R}$, the sensor values (S) of the plant's sensors are read (R), handed over to the plant's network stack (STX) at $t^k_{S,STX}$ and transmitted over the communication network (NTX) at $t^k_{S,NTX}$. The controller's network

interface receives the sensor data (NRX) at $t^k_{\mathrm{S,NRX}}$ and its network stack delivers the packet to the control application (SRX) at time $t^k_{\mathrm{S,SRX}}$. Afterwards, the controller calculates the actuation values for the actuators and hands over the actuation message (A) to the network stack at $t^k_{\mathrm{A,STX}}$, which sends the packet over the network at $t^k_{\mathrm{A,NTX}}$. Finally, at time $t^k_{\mathrm{A,NRX}}$, the plant's network interface receives the actuation packet, and, at time $t^k_{\mathrm{A,SRX}}$, its network stack delivers it to the actuator application, which applies (W) the commands to the actuators at $t^k_{\mathrm{A,W}}$.

Thanks to the timing diagram shown in Fig. 5.11, it is possible to identify the delay components of the NCS and identify the delays of the control system, computing system, and communication network. Control system delays arise from the *processing time* (P) of the control algorithms. They occur at the robot during sensing $d^k_{\mathrm{P,S}}$ and actuation $d^k_{\mathrm{P,A}}$, and at the controller computing the control logic $d^k_{\mathrm{P,C}}$. Computing systems delays arise while *processing* the packets from and to the network interface at the robot $d^k_{\mathrm{P,STX}}$, $d^k_{\mathrm{P,ARX}}$ and at the controller $d^k_{\mathrm{P,SRX}}$, $d^k_{\mathrm{P,ATX}}$. Finally, network delays (N) can be classified in uplink delay $d^k_{\mathrm{N,S}}$, when sensor values are transmitted, and downlink delays $d^k_{\mathrm{N,A}}$, when actuation commands are transmitted.

In an ideal operation, all the *delays* are bounded and within the sampling period of the control loop. However, in a real implementation, the delays vary according to the chosen software and hardware of the control system, computing system, and communication network. While shorter delays can be compensated with simple techniques such as busy waiting, higher delays severely impact the QoC. In fact, high delays cause the late delivery of actuation messages and failure in controlling the plant.

5.4.2 NCS Benchmarking Platform Implementation

Following the architecture of Sec. 5.4.1 an open-source[1] NCS benchmarking platform was implemented, which is described in this section. The platform aims at controlling, using a common Internet Protocol (IP) network, a NIP and implements the measurement of delays and QoC for the characterization of its performance.

The implementation was developed keeping in mind reproducibility design principles, hence, all the software and hardware components used in the proposed platform are low-cost and highly accessible. The NIP is built using the Lego Mindstorms™ platform, communicates using standard Ethernet and WLAN network interfaces, is open-source[1], and is programmed using the Python programming language. This flexibility allows the proposed platform to be used for experiments and benchmarking of arbitrary NCS configurations. In fact, each element of the architecture shown in Fig. 5.10 can be changed. Different physical plants can be built using Lego™, new control logics can be programmed in Python, arbitrary Transmission Control Protocol (TCP)/IP network interfaces can be connected, and common computing

Parameter	Description
Control Application	Python 3 open-source[1] code.
Control Physical System	*Gyro Boy* from the Lego Mindstorms EV3™.
Control Actuator/Sensor	DC brushed EV3 Large Servo Motors, EV3 Gyro Sensor.
Network Access Point$_A$	TP-Link TL841ND.
Network Access Point$_B$	Edimax BR6208AC.
Network Stack Controller	Ubuntu 18.04 LTS (Kernel version 4.15).
Network Stack Robot	Debian Jessie (Kernel version 4.4).
Network HW Controller$_A$	Intel 82579LM 1 GbE NIC.
Network HW Controller$_B$	ASIX AX88179 1 GbE.
Network HW Robots	Edimax EW-7811Un W-LAN USB dongle.
Network Environment	Office, indoor, comm. distance 1-2 m, Wi-Fi interference.
Comp. SW Higher layers	UDP, application protocol described in Sec. 5.4.2.2.
Comp. HW Controller$_A$	Intel Core i2520M (2 cores, 2.5 GHz, 8 GiB RAM).
Comp. HW Controller$_B$	Intel Core i7-6700 (4 cores, 3.40 GHz, 16 GiB RAM).
Comp. HW Robot	32-bit ARM9 SoC (1 core, 300 MHz, 64 MiB RAM).

Table 5.3: Architecture parameters of two NCS platforms. Platform A, developed during the initial implementation, and platform B, reproduced for benchmarking purposes.

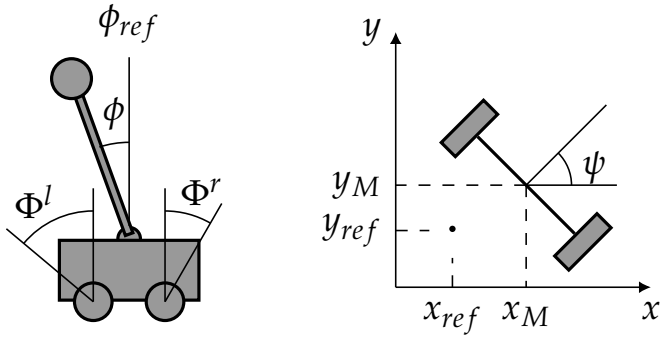

Figure 5.12: Model of the experimental NIP.

and operating systems can be used. The implementation details of the NCS benchmarking platform are provided in Sec. 5.4.2.1, 5.4.2.2, and 5.4.2.3 for the control, computation, and communication domains. Furthermore, Tab. 5.3 summarizes the implementation parameters according to each element of the NCS architecture of Fig. 5.10.

5.4.2.1 Control System

The plant is built by following the default instructions of the *Gyro Boy* robot of the Lego Mindstorms Education EV3 Core Set™ until step 61 [53]. The main difference with respect to the NCS model presented in Sec. 5.2 is that the robot is modelled on a bi-dimensional plane. For this reason, as represented in Fig. 5.12, the robot's body is supported by two wheels, each one splined to an actuator, the DC brushed EV3 Large Servo Motor™, capable of rotating

it on the bi-dimensional plane. The control variables are the voltages applied to the left and right motors, respectively, $v^l(t), v^r(t) \in [-\bar{V}, \bar{V}]$, where $\bar{V}$ is the full-scale voltage of the motor and equal to $8\,V$.

The following sensors are used to measure the plant's trajectory. An *incremental encoder* is mounted on each motor's shaft and measures the rotation angle of the corresponding wheel with regards to the robot's body. As in the left part of Fig. 5.12, $\Phi^l(t)$ and $\Phi^r(t)$ indicate the rotation of the left and rights wheels with regards to z-axis. As in [73], $\Phi(t)$ describes the average rotation angle of the two wheels with regards to the z-axis, i.e., $\Phi(t) = 0.5 \cdot \left[\Phi^l(t) + \Phi^r(t)\right]$. The robot moves onto the bi-dimensional plane formed by axes x and y in the right part of Fig. 5.12. The position of the body in the bi-dimensional plane at time t is $(x_M(t), y_M(t))$. Moreover, the orientation of the robot with regards to the x-axis at time t is denoted by $\psi(t)$ and is called the yaw angle. A *one-dimensional gyroscope*, the EV3 Gyro Sensor™, is mounted on the body and measures the pitch rate $\dot{\phi}(t)$. With regards to the right part of Fig. 5.12, $\phi(t)$ is called pitch angle and denotes the angle at time t between the z-axis and the axis passing through the robot's body.

Due to the gravity force, the position $\phi(t) = 0$ exhibits an unstable equilibrium. The *control goal* is, thus, to balance the robot, i.e., to hold $\phi(t) = 0$, while tracking a desired position and orientation in plane x-y, i.e., to hold $(x_M(t), y_M(t), \psi(t)) = \left(x_M^{ref}(t), y_M^{ref}(t), \psi^{ref}(t)\right)$. This task can be achieved by employing a closed-loop controller, that gets the sensors measurements and computes the adequate control action for the two motors. The control of the NIP is performed in two steps. First, the measurement of pitch rate and robot's position are filtered in order to remove the measurement noise and compute the six components of the state vector. Then, an LQR controller is used to compute the value of motor voltages to be applied at the two motors. The extensive derivation of the control law and filter for our platform can be found in [92].

The plant is responsible for the periodic operation of the control loop and regularly triggers sensor readings every T_S. During the operation of the platform, due to the superposition of multiple random system delays, it can happen that actuation packets arrive earlier or later than the exact sampling period. While shorter delays can be compensated by performing busy waiting, packets experiencing delays larger than the sampling period are considered lost by the control system. As discussed in Sec. 5.3, this condition incorporates the delay and reliability achieved by the network, as the loss of a control packet can occur due to packet loss in the wireless medium or due to the total system delays. In case of packet loss, a model-based prediction is performed at the controller and is used to compensate packet loss. As the controller does not know a-priori whether a packet will be lost, it calculates and sends, with each actuation message, a list of control inputs that are computed based on model-based

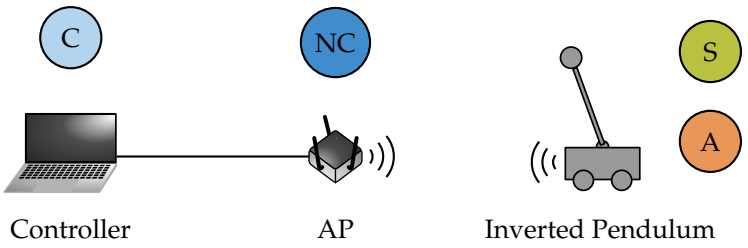

Figure 5.13: Two-hop network topology of the experimental NCS platform. The first hop connects the controller to a WLAN Access Point (AP) via Ethernet. The second hop connects the AP to the NIP via a 2.4 GHz WLAN connection.

state predictions of the next steps. The details of the model-based prediction adopted by the controller can be found in [92].

5.4.2.2 Computing Systems

Two different computing systems are deployed in the implementation: one for the controller and one for the robot. The robot is mobile and battery powered, requiring a computing system optimized for compact size and low energy consumption. The required computational complexity of the two computing systems is different. The calculation of the control commands requires a higher computational effort than their execution and the sensing operation. Therefore, the computing system of the controller is more powerful than the computing system of the plant. In the experiments described in this section, the robot deploys a 32-bit ARM9 microcontroller, which has limited processing capabilities, and the controller employs a powerful multi-purpose 64-bit computer with Linux Operating System.

In order to communicate, both computing systems implement the widespread TCP/IP network stack and the same application protocol. The *application protocol* consists of two messages: the sensor value message, created by the robot and sent to the controller, containing the sensor's measurements, and the actuation command message, created by the controller replying to the sensor value message, containing the voltages to be applied at the motors. In addition, sequence numbers and timestamps are transmitted for packet loss, reordering detection, and delay measurements.

5.4.2.3 Communication Network

The implemented communication network is designed to be easily reproducible and flexible with respect to the possible communication technologies. It is structured according to the OSI communication model and logically separated from the computing system at the network layer, i.e., everything below is part of the communication network.

The *network topology* defines the connectivity of different network nodes at the network and link layers. In our case, a simple two-hop topology is implemented and shown in Fig. 5.13. The first hop connects the controller to a WLAN AP via Ethernet. The second hop connects the AP to the robot via a WLAN network interface. In our architecture, the *network interface* defines the link-layer MAC scheme. The robot has no native network interface, the wireless connection is realized via the USB 2.0 interface.

The wireless *propagation environment* used in the experiments is a typical quiet indoor office environment, there are no moving objects and the communication is subject to low external interference. The WLAN AP and the robot communicated at an approximate distance of 1-2 m.

5.4.3 Measurements of the QoS–QoC Relationship

Measurements of the NCS benchmarking platform provide an experimental characterization of the QoS–QoC relationship. In particular, measurements of the delays shown in Fig. 5.11 are implemented for each component of the architecture. The influence of the network stack of the controller is recorded using a packet traces on the ingress/egress network interface via the tcpdump program. However, hardware limitations of the robot's computing system complicate the measurement of the individual delays and impact the NCS performance. Due to this limitation, the specific delays $d_{\mathrm{P,STX}}$, $d_{\mathrm{N,S}}$, $d_{\mathrm{P,SRX}}$, and $d_{\mathrm{N,A}}$ could not be measured and the average one-way network delay d_{N} is calculated assuming symmetrical network delays, and including the stack delays of controller and plant,

$$d_{\mathrm{N}}^{k} = 0.5 \cdot \left[t_{\mathrm{A,SRX}}^{k} - t_{\mathrm{S,STX}}^{k} \right]. \tag{5.22}$$

For each delay, the jitter is characterized as a property of the delay fluctuation. Low jitter allows a constant stream of information, supporting smooth control performance. To determine jitter, we provide quartiles and 99.9-th percentiles in addition to the median delay.

QoC measurements are performed at the controller during the operation of the platform. Thanks to the continuous transmission of sensor values from the robot, the controller is able to compute the state equation and the actuation command of Eq. (5.1) and (5.2). This way, the QoC metrics can be assessed and used to characterize the control performance. For the robot, QoC_{ϕ} corresponds to Eq. (5.8), while Eq. (5.7) and (5.9) are adapted to the bi-dimensional scenario as follows

$$QoC_{\Phi} = \sum_{k=1}^{T_E} \mid \Phi(kT_s) - \Phi_{ref} \mid = \sum_{k=1}^{T_E} \mid \Phi_k - \Phi_{ref} \mid, \tag{5.23}$$

$$QoC_{\nu} = \frac{1}{2} \sum_{k=1}^{T_E} \left(\mid \nu^{l}(kT_s) \mid + \mid \nu^{r}(kT_s) \mid \right) = \frac{1}{2} \sum_{k=1}^{T_E} \left(\mid \nu_k^{r} \mid + \mid \nu_k^{l} \mid \right). \tag{5.24}$$

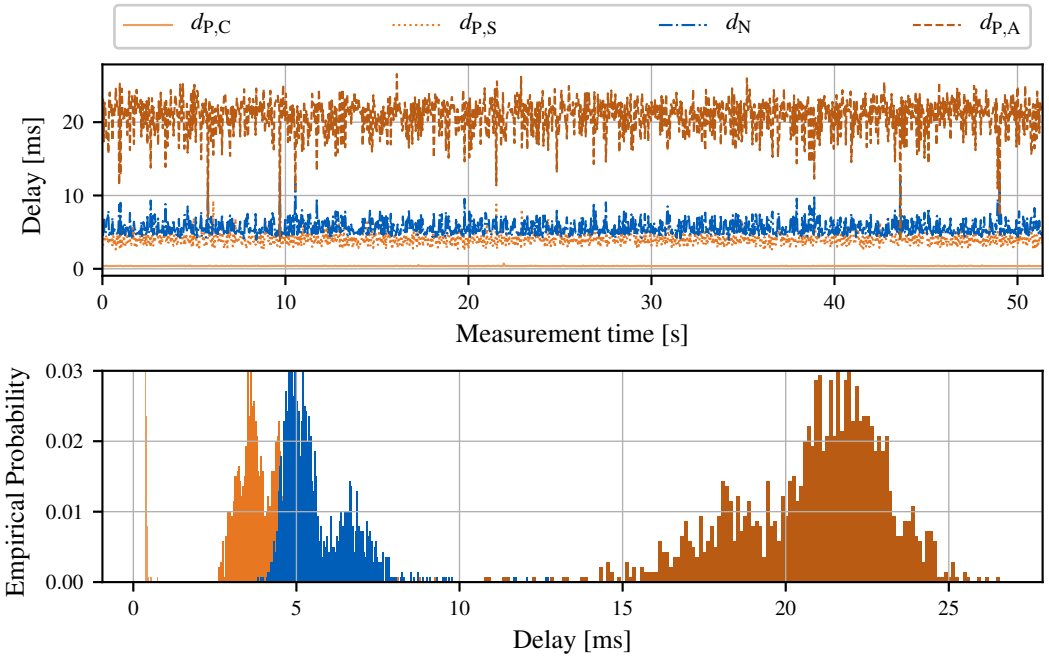

Figure 5.14: Time evolution and empirical distribution of the delays of the controller, sensor, actuator, and network.

Every experiment is conducted as follows. Before the experiment starts, the robot lies on the ground continuously sending sensor values to the controller. The controller, however, does not send actuation commands until the robot is manually lifted to the vertical position. For this reason, the beginning of the experiment is the time at which the robot manually reaches the vertical position for the first time, and corresponds to 0 s in our evaluation. Afterwards, continuous exchange of information between the robot and the controller takes place and enables the control loop to balance the NIP. The duration of the experiment is determined by the control logic and is equal to $T_E = 1400$ sampling periods, i.e., $49\,s$ with $T_S = 35\,ms$. Whenever the experiment ends, the controller stops sending actuation messages to the robot, opening the control loop. In this way, for every experiment, an even number of samples is collected, and the delay and QoC metrics can be calculated and analysed.

Fig. 5.14 shows the time evolution and the histogram of the delays of the controller, sensor, actuator, and network defined in Fig. 5.11. The sensor reading delay $d_{P,S}$ ② demonstrates a stable behaviour with occasional outliers reaching up to 5.5 ms. Similarly, the controller delay $d_{P,C}$ ① is very stable, showing almost no outliers. Overhead caused by the controller network stack is constant and marginal (approx. 37 μs) over the entire experiment. The actuator delay $d_{P,A}$ ④ shows an unstable behaviour over the entire measurement period. Its jitter, also expressed by the width of its distribution in the histogram, is attributed to the control algorithm, which implements busy waiting. Therefore, $d_{P,A}$ ④ includes a waiting period that

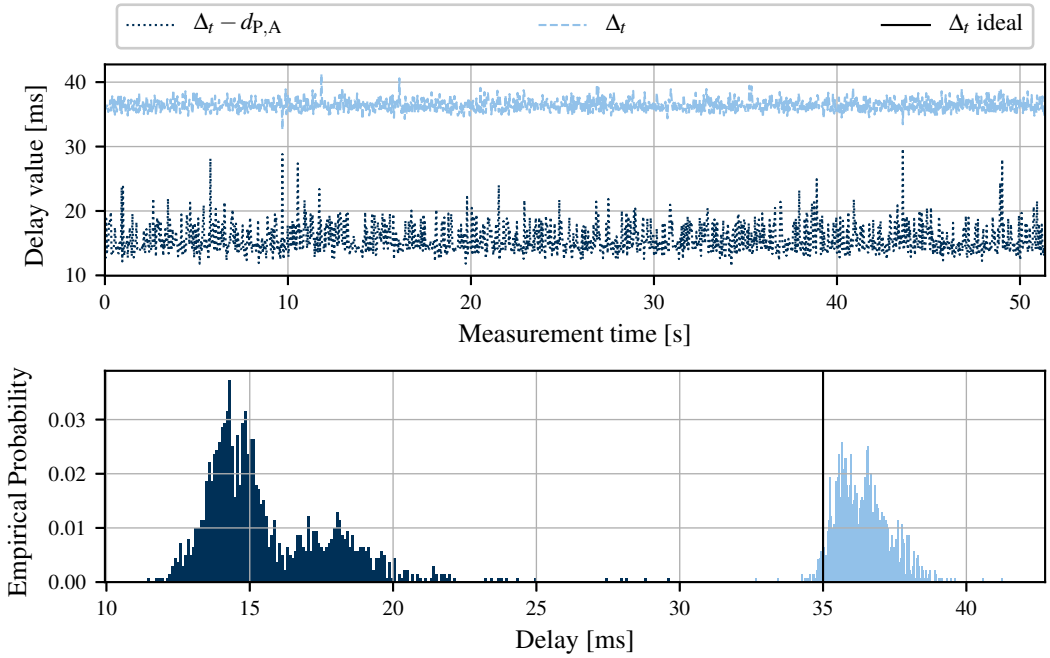

Figure 5.15: Time evolution and empirical distribution of the round-trip delays and of the measured sampling period.

directly depends on the previous steps and their individual delays. In fact, in order to instruct the motors every 35 ms, actuation commands are only applied after a delay of 29 ms from the beginning of the sampling period. The motors of the platform require approximately 6 ms to actuate the motors.

When analysing the jitter given in the histograms, $d_{P,C}$ ① shows the most stable behaviour (0.9 ms - 1.2 ms), indicating that the controller always has enough computing power to handle the control process in a timely manner. The sensor reading delay $d_{P,S}$ ② shows a minimal time of 2.4 ms for sensor readings with a tail of up to 11.1 ms. The network delay $d_{N,S}$ ③ roughly resembles a normal distribution ranging from 4.7 ms up to 15 ms, and it is originated by the Carries Sense Multiple Access (CSMA)/Collision Avoidance (CA) mechanism of WLAN in our physical environment.

Fig. 5.15 shows the time evolution and histogram of the cumulative delays. The timestamp $t_{A,SRX}$ is collected by the robot application after receiving the actuation message, resulting in the delay $\hat{\Delta}_T - d_{P,A}$ ⑤. Where $\hat{\Delta}_T$ is the measured sampling period of the NCS during the experiment. The histogram of ⑤ shows a wide distribution, ranging from 13.2 ms up to 35.1 ms, employing the jitter of all the previous steps. However, if $d_{P,A}$ is included in the plot ($\hat{\Delta}_T$ ⑦), the jitter decreases, as the actuator algorithm applies the actuation commands only 29 ms after the beginning of the sampling period. This effect results in a rather constant measured sampling period $\hat{\Delta}_T$ ⑦, and allows the compensation of the previous delays, leading to a rather

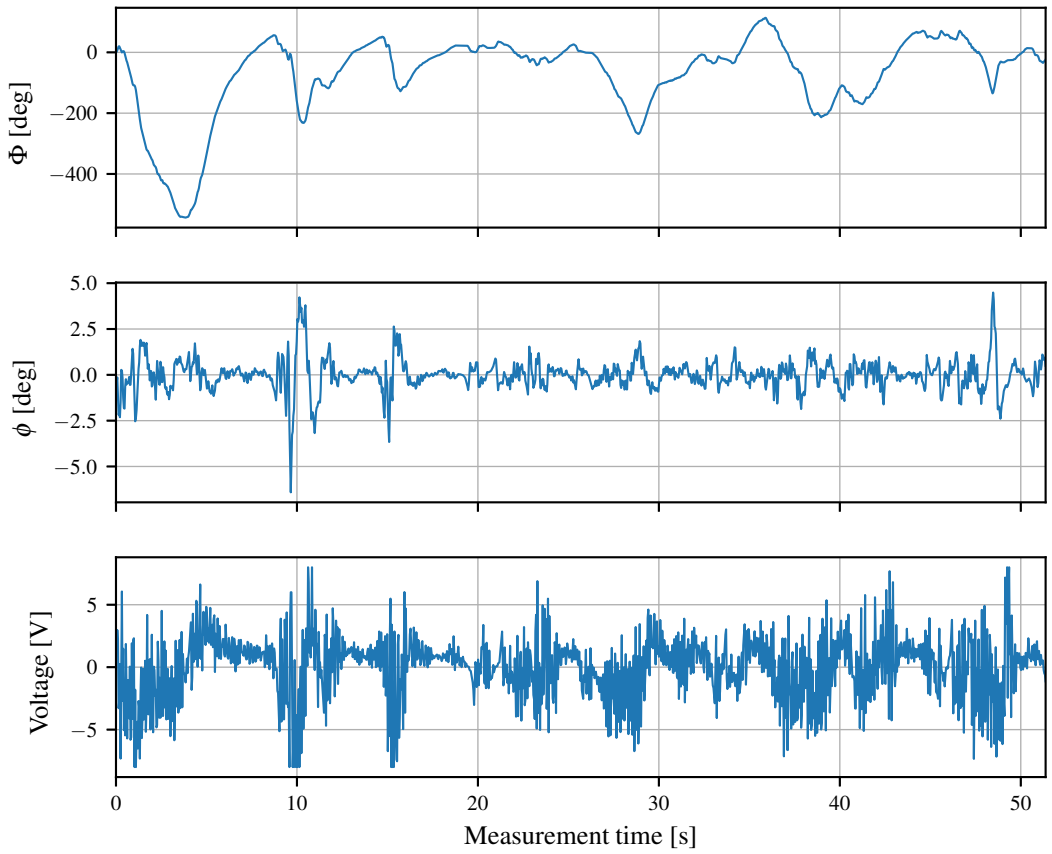

Figure 5.16: Time evolution of the filtered pitch angle ϕ, the filtered avg. rotation angle Φ, and the avg. applied voltage at the motors ν.

low jitter. Thus, the distribution of $\hat{\Delta}_T$ ⑦ is more compact and allows a constant delivery of the actuation commands close to the ideal sampling period Δ_T ⑥. Its jitter is caused by the precision of the busy waiting technique and by the time required to actuate the motors.

The impact of the control logic is reflected in Fig. 5.16, showing the evolution of the QoC metrics. The pitch angle ϕ_k of the robot is highly varying, with occasional larger spikes every few seconds. Despite this, its dynamic remains bounded during the entire execution and that its average value is equal to -0.0014 deg. The evolution of the motors' applied voltage strongly depends on the pitch angle. In fact, higher voltages are correlated with higher values of pitch angle. This effect is also shown in the position of the robot Φ_k, which presents faster and slower oscillations. Faster oscillations, visible between 3-5 s, are caused by strong and opposite actuation commands needed to compensate high values of pitch angles and to balance the robot. Slower oscillations arise whenever the control logic tries to bring the robot to its initial position. This task has lower priority compared to balancing the robot, and it is performed on a larger time scale.

	Median ±95% C.I.	Q_1	Q_3	99.9%
Wireless A				
$d_{\mathrm{P,C}}$	0.95 ± 0.002	0.92	0.96	1.05
$d_{\mathrm{P,S}}$	3.64 ± 0.049	3.03	4.36	6.20
d_{N}	8.09 ± 0.053	7.54	8.54	10.88
$d_{\mathrm{P,A}}$	15.19 ± 0.118	13.79	16.55	19.94
$\hat{\Delta}_t$	35.89 ± 0.057	35.22	36.62	38.97
Wireless B				
$d_{\mathrm{P,C}}$	0.37 ± 0.001	0.37	0.38	0.43
$d_{\mathrm{P,S}}$	3.84 ± 0.040	3.49	4.45	6.39
d_{N}	5.25 ± 0.055	4.85	6.29	8.74
$d_{\mathrm{P,A}}$	21.27 ± 0.126	19.49	22.38	24.84
$\hat{\Delta}_t$	36.32 ± 0.049	35.70	36.95	38.76

Table 5.4: Experimental delays of the NCS platforms A and B.

	QoC_ϕ	QoC_Φ	QoC_ν
Wireless A	938.296	217227.214	2636.785
Wireless B	785.725	129534.619	2724.883

Table 5.5: Experimental QoC values of the NCS platforms A and B.

In order to investigate how QoS and QoC values are affected by different implementations of the NCS platform, a second platform was implemented. The second platform, referred to as platform B, uses a different hardware for the controller and WLAN communication with respect to the first platform, referred to as platform A. The implementation details of both platforms are summarized in Tab. 5.3. QoS and QoC measurements of the platforms for an experiment of duration $T_E = 1400$ sampling periods are summarized in Tabs. 5.4 and 5.5. The delays in Tab. 5.4 are presented as median with 95% confidence intervals, 1st and 3rd quartiles Q_1, Q_3, and 99.9-th percentiles. The platforms achieve different delay performances. The median values of $d_{\mathrm{P,C}}$ is lower for platform B than platform A, despite showing similar jitter and worst-case values. A minor difference is noticeable in the sensor processing delays $d_{\mathrm{P,S}}$; platform A has smaller median delays but with a higher jitter. Platform A shows the worst network performance, with the highest median value and 99.9-th percentile. This can be explained by the fact that the experiments of platforms A and B were performed in two different environments by me and Mr. Gallenmüller. The actuator processing delays $d_{\mathrm{P,A}}$ directly depend on the busy waiting procedure. Its quartiles reflect the network delays, being wider and with larger worst-case values for worse network performance. The measured sampling period $\hat{\Delta}_T$ is comparable to both scenarios. Tab. 5.5 shows comparable values of

QoC for the two NCS platforms. Although the platforms achieve similar values of QoC_v, platform A shows higher values for QoC_ϕ and QoC_Φ, thus it was less stable than platform B. This is explained by the worse wireless performances of platform A that experience delays that are on average 2.5 ms higher in each transmission of sensor and controller.

5.5 Concluding Remarks

This chapter presented a detailed evaluation of QoC for a simulated NIP operating in an IWSN and of the QoS–QoC measurements of an experimental NCS operating in a WLAN network. Via the LSP metric, the impact of the network-layer QoS performance of IWSN on different QoC metrics was evaluated. Furthermore, thanks to a delay-reliability model, the E2E delay of each message of the control loop was characterized and related to LSP. The delay-reliability model describes the communication parameters of the IWSN and follows the link- and network-layer models described in Chapters 3 and 4. Measurements of an IWSN testbed verified the correctness of the delay-reliability model and its direct relationship to QoC. Thanks to the above contributions, given the link- and network-layer parameters of an IWSN, it is possible to determine the QoC of an NCS. This extends the existing related works via the analysis of the QoS performances of each packet of the control system in the presence of packet loss and delays. Furthermore, it provides a direct interconnection between QoC and the link- and network-layer parameters of IWSN.

The impact of wireless communication on a real NCS was further investigated via the implementation and evaluation of an experimental NCS platform. This was achieved via the derivation of a model of the NCS architecture and the delay-analysis of a control loop. Furthermore, the architecture was used to develop an open-source NCS benchmarking platform. The platform implements the measurement of all the delays in each control loop and of the QoC metrics defined in this chapter. Measurement results provide a direct interconnection between QoC and the delays arising from control, computation, and communication. This is a fundamental achievement for the operation of NCS in wireless networks. In fact, differently from existing works that considered a simulated NCS or evaluated a specific aspect of experimental NCS, the measurement results provide a comprehensive characterization of the experimental delays of hardware and software components of an NCS and their relationship to QoC.

Chapter 6

Conclusions and Outlook

This doctoral thesis presented several Radio Resource Management (RRM) methods for Industrial Wireless Sensor Networks (IWSN) in order to support the communication of feedback-based applications, such as Networked Control Systems (NCS), in the presence of packet loss and delays. This was achieved by investigating the reliability of individual IWSN transmissions, via the derivation of link- and network-layer scheduling schemes, and by evaluating their impact on the performance of NCS. Sec. 6.1 and 6.2 conclude this thesis summarizing the main findings and discussing possible directions for future work.

6.1 Summary and Discussion

The thesis investigated the problem of Quality of Service (QoS) provisioning in interfered IWSN in order to fulfil the communication requirements of NCS. Following a bottom-up approach, the problem of random packet loss caused by the harsh industrial propagation environment was tackled at different Open Systems Interconnection (OSI) layers. Therefore, different solutions have been proposed at the physical, link, networking, and application layers.

Ch. 2 investigated the physical-layer reliability of IWSN transmissions by characterizing the average communication Packet Error Rate (PER) in different scenarios. Via an analytical model, the PER was evaluated in propagation environments with different attenuation and multipath fading characteristics. Furthermore, the impact of external interference on the PER was evaluated thanks to measurement- and software-based estimators. Measurement-based estimators characterize the impact of the propagation environment, interference, and noise on the PER via measurements of received power. Software-based estimators estimate the PER by observing the received packets and are suitable to estimate PER in the presence of time-varying interference.

Ch. 3 addressed the problem of QoS provisioning in the presence of interference via the definition of link-layer RRM schemes. A stochastic model of the link-layer radio resources was developed to evaluate the QoS performance achieved by schedules. In particular, the whitening hopping sequence design was presented to provision reliable consecutive re-transmissions to each device using both interfered and non-interfered channels. Results showed that whitening achieves higher reliability compared to existing methods. Furthermore, a reliability-based scheduler was developed to provide link-layer QoS in the presence of dynamic interference. The results proved that, thanks to the timely estimation of PER performed by an Exponentially Weighted Moving Average (EWMA) estimator, bounded delays and a target reliability can be provisioned in the presence of time-varying interference.

In Ch. 4, network-layer scheduling policies have been investigated to support the transmission of feedback-based applications over a two-hop network path. With respect to a desired application deadline, a transient model of the network was derived to characterize the delay distribution of individual packets by computing their Delay Violation Probability (DVP). Static and dynamic scheduling policies have been developed to optimize the network resources in order to minimize DVP, which exploit, respectively, initial and continuous information of the queue states. Results showed that static policies improve the DVP by one order of magnitude with respect to a resource allocation that ignores the initial queue states. Furthermore, the dynamic scheduling policy provides an improvement of several orders of magnitude with respect to static policies. The performance improvement increases for longer frames and high deadline values.

Ch. 5 presented a detailed evaluation of the Quality of Control (QoC) achieved by the communication of NCS in an IWSN and a Wireless Local-Area Network (WLAN) network. QoC in the IWSN was characterized via the Loop Success Probability (LSP), which models the End-to-End (E2E) delay distribution of packets in a control loop and relates it with the requirements of the NCS. The analytical derivation of LSP was presented and verified via measurements of an IWSN testbed for a simulated Networked Inverted Pendulum (NIP). Results prove that, given the communication parameters of the IWSN, the QoC of a NIP can be fully characterized. The relationship between network QoS and QoC was additionally investigated for an experimental NCS. A complete characterization of the NCS was performed thanks to a model of the NCS architecture and the delay-analysis of a control loop. Following the model, an open-source NCS benchmarking platform was implemented for experimental measurements. Measurement results provide a direct interconnection between QoC and the delays arising from control, computation, and communication elements of the platform.

6.2 Directions for Future Work

The results of this doctoral thesis leave space for several future research works in order to improve the communication of NCS in IWSN.

PER estimation is a fundamental enabler of NCS communication in IWSN, however, it introduces a communication overhead that reduces the amount of available resources for the transmission of data packets. Therefore, a detailed analysis of the resources needed by the estimators can be performed to characterize their impact on the QoS of the proposed RRM schemes and on the QoC of the NCS. In this way, the parameters and the overhead of the estimators can be optimally tuned according to the propagation scenario and application requirements.

The proposed link-layer RRM schemes prove their suitability for QoS provisioning in the presence of interference. Their application, however, is suitable to scenarios where interference changes are slow compared to the timings of the medium access and when QoS can be provisioned to an entire flow of packets. Therefore, faster RRM schemes can be investigated in order to react to sudden changes in the communication PER and to provide QoS to individual messages subject to these disturbances.

Thanks to the developed network-layer static and dynamic scheduling policies, the network resources can be optimized to minimize the DVP of control packets. The network model, however, assumes a symmetric PER for the transmission of sensor and controller. Furthermore, it assumes that, for each measurement generated by the sensor, the controller computes an actuation message of the same size. Future works can investigate optimal scheduling policies that relax these assumptions. Furthermore, novel scheduling policies can be investigated in order to take into account the DVP of multiple packets generated by different control loops sharing the network.

The analysis of QoC performed by the thesis leaves space for further interesting investigations. The delay-reliability model of IWSN can be extended to include the delays arising from additional aspects of the NCS, such as sensing, computation, and actuation delays. This can be achieved following the NCS architecture developed for the implementation of the open-source NCS platform. In this way, both the delay-reliability model and the NCS platform can be extended, respectively, to include the effects of a real implementation and of IWSN transmissions on experimental NCS.

Bibliography

Publications by the author

Journal publications

[1] S .Zoppi and J. P. Champati and J. Gross and W. Kellerer. "Scheduling of Wireless Edge Networks for Feedback-Based Interactive Applications." In: *IEEE Transactions on Communications* (2022), pp. 1–15. DOI: 10.1109/TCOMM.2022.3163761.

[2] S. Zoppi, A. Van Bemten, H. M. Gürsu, M. Vilgelm, J. Guck, et al. "Achieving Hybrid Wired/Wireless Industrial Networks With WDetServ: Reliability-Based Scheduling for Delay Guarantees." In: *IEEE Transactions on Industrial Informatics* 14.5 (May 2018), pp. 2307–2319. DOI: 10.1109/TII.2018.2803122.

Conference publications

[3] S. Zoppi and S. P. Shantharam and W. Kellerer. "Delay-Reliability Model of Industrial WSN for Networked Control Systems." In: *2020 IEEE Global Communications Conference: Communication QoS, Reliability and Modeling (IEEE Globecom 2020 CQRM)*. Dec. 2020. DOI: 10.1109/GLOBECOM42002.2020.9348072.

[4] S. Zoppi, O. Ayan, F. Molinari, Z. Music, S. Gallenmüller, et al. "NCSbench: Reproducible Benchmarking Platform for Networked Control Systems." In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. Jan. 2020, pp. 1–9. DOI: 10.1109/CCNC46108.2020.9045199.

[5] S. Zoppi, J. P. Champati, J. Gross, and W. Kellerer. "Dynamic Scheduling for Delay-Critical Packets in a Networked Control System Using WirelessHART." In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. June 2020, pp. 1–7. DOI: 10.1109/ICC40277.2020.9148930.

[6] S. Zoppi, H. M. Gürsu, M. Vilgelm, and W. Kellerer. “Reliable Hopping Sequence Design for Highly Interfered Wireless Sensor Networks.” In: *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. June 2017, pp. 1–7. DOI: 10.1109/LANMAN.2017.7972164.

General publications

[7] N. Abu-Ali, A. M. Taha, M. Salah, and H. Hassanein. “Uplink Scheduling in LTE and LTE-Advanced: Tutorial, Survey and Evaluation Framework.” In: *IEEE Communications Surveys Tutorials* 16.3 (2014), pp. 1239–1265. DOI: 10.1109/SURV.2013.1127.00161.

[8] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler. “Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things.” In: *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 2013, pp. 1–6. DOI: 10.1109/WoWMoM.2013.6583485.

[9] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, et al. “Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation.” In: *IEEE Internet of Things Journal* 2.6 (2015), pp. 455–470. DOI: 10.1109/JIOT.2015.2476915.

[10] A. Aijaz and U. Raza. “DeAMON: A Decentralized Adaptive Multi-Hop Scheduling Protocol for 6TiSCH Wireless Networks.” In: *IEEE Sensors Journal* 17.20 (2017), pp. 6825–6836. DOI: 10.1109/JSEN.2017.2746183.

[11] K. Akkaya and M. Younis. “A survey on routing protocols for wireless sensor networks.” In: *Ad Hoc Networks* 3.3 (May 2005), pp. 325–349. DOI: 10.1016/j.adhoc.2003.09.010.

[12] I. F. Akyildiz and M. C. Vuran. *Wireless Sensor Networks*. Vol. 4. John Wiley & Sons, Ltd, 2010. DOI: 10.1002/9780470515181.

[13] E. Ancillotti, R. Bruno, and M. Conti. “The role of the RPL routing protocol for smart grid communications.” In: *IEEE Communications Magazine* 51.1 (2013), pp. 75–83. DOI: 10.1109/MCOM.2013.6400442.

[14] Y. P. Aneja and K. Nair. “The Constrained Shortest Path Problem.” In: *Naval Research Logistics Quarterly* 25.3 (1978), pp. 549–555. DOI: 10.1002/nav.3800250314.

[15] I. S. of Automation. *ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications*. 2011.

[16] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, et al. “Radio Link Quality Estimation in Wireless Sensor Networks: A Survey.” In: *ACM Transactions on Sensor Networks* 8.4 (Sept. 2012). DOI: 10.1145/2240116.2240123.

[17] C. Bachhuber, S. Conrady, M. Schütz, and E. Steinbach. “A Testbed for Vision-Based Networked Control Systems.” In: *Computer Vision Systems*. Springer International Publishing, 2017, pp. 26–36. DOI: 10.1007/978-3-319-68345-4_3.

[18] J. Bai, E. P. Eyisi, Y. Xue, and X. D. Koutsoukos. “Dynamic Tuning Retransmission Limit of IEEE 802.11 MAC Protocol for Networked Control Systems.” In: *2010 IEEE/ACM Int’l Conference on Green Computing and Communications Int’l Conference on Cyber, Physical and Social Computing*. 2010, pp. 666–671. DOI: 10.1109/GreenCom-CPSCom.2010.48.

[19] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe. “Evaluating Low-Power Wireless Cyber-Physical Systems.” In: *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*. 2018, pp. 13–18. DOI: 10.1109/CPSBench.2018.00009.

[20] D. Blokh and G. Gutin. “An Approximate Algorithm for Combinatorial Optimization Problems with Two Parameters.” In: *Australasian Journal of Combinatorics* 14 (1996), pp. 157–164.

[21] G. Boggia, P. Camarda, L. A. Grieco, and G. Zacheo. “Toward wireless Networked Control Systems: An experimental study on real-time communications in 802.11 WLANs.” In: *2008 IEEE International Workshop on Factory Communication Systems*. 2008, pp. 149–155. DOI: 10.1109/WFCS.2008.4638711.

[22] R. Brummet, D. Gunatilaka, D. Vyas, O. Chipara, and C. Lu. “A Flexible Retransmission Policy for Industrial Wireless Sensor Actuator Networks.” In: *2018 IEEE International Conference on Industrial Internet (ICII)*. 2018, pp. 79–88. DOI: 10.1109/ICII.2018.00017.

[23] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda. “Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey.” In: *IEEE Communications Surveys Tutorials* 15.2 (2013), pp. 678–700. DOI: 10.1109/SURV.2012.060912.00100.

[24] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino. “Evaluation of Response Times in Industrial WLANs.” In: *IEEE Transactions on Industrial Informatics* 3.3 (2007), pp. 191–201. DOI: 10.1109/TII.2007.903219.

[25] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin. "Statistical model of lossy links in wireless sensor networks." In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* 2005, pp. 81–88. DOI: 10.1109/IPSN.2005.1440900.

[26] A. Chamaken and L. Litz. "Joint design of control and communication in wireless networked control systems: A case study." In: *Proceedings of the 2010 American Control Conference*. 2010, pp. 1835–1840. DOI: 10.1109/ACC.2010.5531426.

[27] J. P. Champati, H. Al-Zubaidy, and J. Gross. "Transient Analysis for Multihop Wireless Networks Under Static Routing." In: *IEEE/ACM Transactions on Networking* 28.2 (2020), pp. 722–735. DOI: 10.1109/TNET.2020.2975616.

[28] M. Cheffena. "Propagation Channel Characteristics of Industrial Wireless Sensor Networks [Wireless Corner]." In: *IEEE Antennas and Propagation Magazine* 58.1 (2016), pp. 66–73. DOI: 10.1109/MAP.2015.2501227.

[29] T. Chen, S. Kuo, and C. Kuo. "Scheduling for Data Collection in Multi-hop IEEE 802.15.4e TSCH Networks." In: *2016 International Conference on Networking and Network Applications (NaNA)*. 2016, pp. 218–221. DOI: 10.1109/NaNA.2016.23.

[30] Y. Chen, H. Zhang, N. Fisher, L. Y. Wang, and G. Yin. "Probabilistic Per-Packet Real-Time Guarantees for Wireless Networked Sensing and Control." In: *IEEE Transactions on Industrial Informatics* 14.5 (2018), pp. 2133–2145. DOI: 10.1109/TII.2018.2795567.

[31] T. D. Chung, R. B. Ibrahim, V. S. Asirvadam, N. B. Saad, and S. M. Hassan. "Simulation of WirelessHART networked control system with packet dropout." In: *2015 10th Asian Control Conference (ASCC)*. 2015, pp. 1–6. DOI: 10.1109/ASCC.2015.7244389.

[32] I. L. S. Committee. "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)." In: *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)* (2011), pp. 1–314. DOI: 10.1109/ieeestd.2011.6012487.

[33] A. Conti, D. Dardari, G. Pasolini, and O. Andrisano. "Bluetooth and IEEE 802.11b coexistence: analytical performance evaluation in fading channels." In: *IEEE Journal on Selected Areas in Communications* 21.2 (2003), pp. 259–269. DOI: 10.1109/JSAC.2002.807345.

[34] G. Daneels, B. Spinnewyn, S. Latré, and J. Famaey. "ReSF: Recurrent Low-Latency Scheduling in IEEE 802.15.4e TSCH networks." In: *Ad Hoc Networks* 69 (Feb. 2018), pp. 100–114. DOI: 10.1016/j.adhoc.2017.11.002.

[35] E. Dauti. “Link Quality Estimation for Industrial Wireless Sensor Networks.” MA. Thesis. Chair of Communication Networks, Department of Electrical and Computer Engineering, Technical University of Munich, May 2019.

[36] B. Demirel, Z. Zou, P. Soldati, and M. Johansson. “Modular Design of Jointly Optimal Controllers and Forwarding Policies for Wireless Control.” In: *IEEE Transactions on Automatic Control* 59.12 (2014), pp. 3252–3265. DOI: 10.1109/TAC.2014.2351972.

[37] F. Dobslaw, T. Zhang, and M. Gidlund. “End-to-End Reliability-Aware Scheduling for Wireless Sensor Networks.” In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 758–767. DOI: 10.1109/TII.2014.2382335.

[38] W. Dong, Y. Liu, Y. He, T. Zhu, and C. Chen. “Measurement and Analysis on the Packet Delivery Performance in a Large-Scale Sensor Network.” In: *IEEE/ACM Transactions on Networking* 22.6 (2014), pp. 1952–1963. DOI: 10.1109/TNET.2013.2288646.

[39] M. Drew, Xiangheng Liu, A. Goldsmith, and K. Hedrick. “Networked Control System Design over a Wireless LAN.” In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 6704–6709. DOI: 10.1109/CDC.2005.1583239.

[40] P. Du and G. Roussos. “Adaptive time slotted channel hopping for wireless sensor networks.” In: *2012 4th Computer Science and Electronic Engineering Conference (CEEC)*. 2012, pp. 29–34. DOI: 10.1109/CEEC.2012.6375374.

[41] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert. “6TiSCH: deterministic IP-enabled industrial internet (of things).” In: *IEEE Communications Magazine* 52.12 (2014), pp. 36–41. DOI: 10.1109/MCOM.2014.6979984.

[42] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne. “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH.” In: SenSys 2015 - Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. Association for Computing Machinery, 2015, pp. 337–350. DOI: 10.1145/2809695.2809714.

[43] J. Eker, A. Cervin, and A. Hörjel. “Distributed Wireless Control Using Bluetooth.” In: *IFAC Proceedings Volumes* 34.22 (2001), pp. 360–365. DOI: 10.1016/S1474-6670(17)32965-8.

[44] Á. A. Farías and D. Dujovne. “A queue-based scheduling algorithm for PCE-enabled Industrial Internet of Things networks.” In: *2015 Sixth Argentine Conference on Embedded Systems (CASE)*. 2015, pp. 31–36. DOI: 10.1109/SASE-CASE.2015.7295844.

[45] E. Felemban, Chang-Gun Lee, and E. Ekici. "MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks." In: *IEEE Transactions on Mobile Computing* 5.6 (2006), pp. 738–754. DOI: 10.1109/TMC.2006.79.

[46] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. "High-reliability scheduling in deterministic wireless multi-hop networks." In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–6. DOI: 10.1109/PIMRC.2016.7794839.

[47] P. Gaj, J. Jasperneite, and M. Felser. "Computer Communication Within Industrial Distributed Environment—a Survey." In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 182–189. DOI: 10.1109/TII.2012.2209668.

[48] G. Gamba, F. Tramarin, and A. Willig. "Retransmission Strategies for Cyclic Polling Over Wireless Channels in the Presence of Interference." In: *IEEE Transactions on Industrial Informatics* 6.3 (2010), pp. 405–415. DOI: 10.1109/TII.2010.2050777.

[49] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005. DOI: 10.1017/CBO9780511841224.

[50] P. H. Gomes, T. Watteyne, and B. Krishnamachari. "MABO-TSCH: Multihop and blacklist-based optimized time synchronized channel hopping." In: *Transactions on Emerging Telecommunications Technologies* 29.7 (2018), e3223. DOI: 10.1002/ett.3223.

[51] T. Gong, T. Zhang, X. S. Hu, Q. Deng, M. Lemmon, et al. "Reliable dynamic packet scheduling over lossy real-time wireless networks." In: *Leibniz International Proceedings in Informatics, LIPIcs*. Vol. 133. July 2019. DOI: 10.4230/LIPIcs.ECRTS.2019.11.

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

[53] L. Group. *LEGO MINDSTORMS Education EV3 Core Set*. Visited Nov. 11, 2019. URL: https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set-/5003400.

[54] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer. "Unicast QoS Routing Algorithms for SDN: A Comprehensive Survey and Performance Evaluation." In: *IEEE Communications Surveys Tutorials* 20.1 (2018), pp. 388–415. DOI: 10.1109/COMST.2017.2749760.

[55] Guocong Song and Ye Li. “Cross-layer optimization for OFDM wireless networks-part II: algorithm development.” In: *IEEE Transactions on Wireless Communications* 4.2 (2005), pp. 625–634. DOI: 10.1109/TWC.2004.843067.

[56] R. A. Gupta and M. Chow. “Networked Control System: Overview and Research Trends.” In: *IEEE Transactions on Industrial Electronics* 57.7 (2010), pp. 2527–2535. DOI: 10.1109/TIE.2009.2035462.

[57] M. Gürsu, M. Vilgelm, S. Zoppi, and W. Kellerer. “Reliable Co-Existence of 802.15.4e TSCH-based WSN and Wi-Fi in an Aircraft Cabin.” In: *2016 IEEE International Conference on Communications Workshops (ICC)*. May 2016, pp. 663–668. DOI: 10.1109/ICCW.2016.7503863.

[58] H. Hajian, M. Nabi, M. Fakouri, and F. Veisi. “LaDiS: a Low-Latency Distributed Scheduler for Time-Slotted Channel Hopping Networks.” In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 2019, pp. 1–7. DOI: 10.1109/WCNC.2019.8885896.

[59] G. Y. Handler and I. Zang. “A Dual Algorithm for the Constrained Shortest Path Problem.” In: *Networks* 10.4 (1980), pp. 293–309. DOI: 10.1002/net.3230100403.

[60] A. Hannachi and A. Bachir. “Distributed cell scheduling for multichannel IoT MAC protocols.” In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2017, pp. 1166–1171. DOI: 10.1109/IWCMC.2017.7986450.

[61] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi. “The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey.” In: *IEEE Communications Surveys Tutorials* 15.1 (2013), pp. 101–120. DOI: 10.1109/SURV.2012.040412.00105.

[62] “Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints.” In: *Information Sciences* 278 (2014), pp. 34–55. DOI: 10.1016/j.ins.2014.02.006.

[63] IEEE LAN/MAN Standards Committee. *IEEE Standard Part 15.4e: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*. Vol. 2012. April. 2012, pp. 1–225. DOI: 10.1109/IEEESTD.2012.6185525.

[64] “IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation.” In: *IEEE Std 1646-2004* (2005), pp. 1–36. DOI: 10.1109/IEEESTD.2005.95748.

[65] Y. Jiang. "A Basic Stochastic Network Calculus." In: *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, Aug. 2006. DOI: 10.1145/1159913.1159929.

[66] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara. "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks." In: *2016 IEEE Wireless Communications and Networking Conference*. 2016, pp. 1–6. DOI: 10.1109/WCNC.2016.7565002.

[67] J. Jung, D. Kim, T. Lee, J. Kang, N. Ahn, et al. "Distributed Slot Scheduling for QoS Guarantee over TSCH-based IoT Networks via Adaptive Parameterization." In: *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2020, pp. 97–108. DOI: 10.1109/IPSN48710.2020.00029.

[68] A. Juttner, B. Szviatovski, I. Mecs, and Z. Rajko. "Lagrange relaxation based method for the QoS routing problem." In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. Vol. 2. 2001, pp. 859–868. DOI: 10.1109/INFCOM.2001.916277.

[69] J. N. Al-Karaki and A. E. Kamal. "Routing techniques in wireless sensor networks: a survey." In: *IEEE Wireless Communications* 11.6 (2004), pp. 6–28. DOI: 10.1109/MWC.2004.1368893.

[70] P. A. Kawka and A. G. Alleyne. "Stability and feedback control of wireless networked systems." In: *Proceedings of the 2005, American Control Conference, 2005.* Vol. 4. 2005, pp. 2953–2959. DOI: 10.1109/ACC.2005.1470423.

[71] Kay Soon Low, W. N. N. Win, and Meng Joo Er. "Wireless Sensor Networks for Industrial Environments." In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. Vol. 2. 2005, pp. 271–276. DOI: 10.1109/CIMCA.2005.1631480.

[72] S. Kim, H.-S. Kim, and C. Kim. "ALICE: Autonomous Link-Based Cell Scheduling for TSCH." In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. Association for Computing Machinery, 2019, pp. 121–132. DOI: 10.1145/3302506.3310394.

[73] Y. Kim, S. H. Kim, and Y. K. Kwak. "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot." In: *Journal of Intelligent and Robotic Systems* 44.1 (Sept. 2005), pp. 25–46. DOI: 10.1007/s10846-005-9022-4.

[74] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre. "Whitelisting Without Collisions for Centralized Scheduling in Wireless Industrial Networks." In: *IEEE Internet of Things Journal* 6.3 (2019), pp. 5713–5721. DOI: 10.1109/JIOT.2019.2905217.

[75] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre. "LABeL: Link-Based Adaptive BLacklisting Technique for 6TiSCH Wireless Industrial Networks." In: *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*. Association for Computing Machinery, 2017, pp. 25–33. DOI: 10.1145/3127540.3127541.

[76] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen. "An Industrial Perspective on Wireless Sensor Networks — A Survey of Requirements, Protocols, and Challenges." In: *IEEE Communications Surveys and Tutorials* 16.3 (2014), pp. 1391–1412. DOI: 10.1109/SURV.2014.012114.00058.

[77] J. Li, P. Zeng, X. Zong, M. Zheng, and X. Zhang. "Communication and control co-design for wireless sensor networked control systems." In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*. 2014, pp. 156–161. DOI: 10.1109/WCICA.2014.7052705.

[78] C. Liu, F. Chen, J. Zhu, Z. Zhang, C. Zhang, et al. "Characteristic, Architecture, Technology, and Design Methodology of Cyber-Physical Systems." In: *International Conference on Industrial IoT Technologies and Applications*. Springer International Publishing, 2017, pp. 230–246. DOI: 10.1007/978-3-319-60753-5_25.

[79] J. Liu, P. Wang, and J. Lin. "Real time scheduling for wireless mesh networked control systems." In: *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. 2017, pp. 382–386. DOI: 10.1109/ICCSN.2017.8230140.

[80] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, et al. "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems." In: *Proceedings of the IEEE* 104.5 (2016), pp. 1013–1024. DOI: 10.1109/JPROC.2015.2497161.

[81] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, et al. "Feedback Control Goes Wireless: Guaranteed Stability over Low-Power Multi-Hop Networks." In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. Association for Computing Machinery, 2019, pp. 97–108. DOI: 10.1145/3302509.3311046.

[82] M. S. Mahmoud. "Wireless networked control system design: An overview." In: *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. 2014, pp. 2335–2340. DOI: 10.1109/ISIE.2014.6864983.

[83] C. T. for Matlab and Simulink. *Inverted Pendulum: System Modeling*. 2019. URL: http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum%5C&section=SimulinkModeling. (Last Accessed: 30.12.2020).

[84] M. A. Mehaseb, Y. Gadallah, A. Elhamy, and H. Elhennawy. "Classification of LTE Uplink Scheduling Techniques: An M2M Perspective." In: *IEEE Communications Surveys Tutorials* 18.2 (2016), pp. 1310–1335. DOI: 10.1109/COMST.2015.2504182.

[85] J. Miranda, R. Abrishambaf, T. Gomes, P. Gonçalves, J. Cabral, et al. "Path loss exponent analysis in Wireless Sensor Networks: Experimental evaluation." In: *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. 2013, pp. 54–58. DOI: 10.1109/INDIN.2013.6622857.

[86] V. P. Modekurthy, A. Saifullah, and S. Madria. "DistributedHART: A Distributed Real-Time Scheduling System for WirelessHART Networks." In: *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 2019, pp. 216–227. DOI: 10.1109/RTAS.2019.00026.

[87] "Modeling and stabilization of networked control systems with bounded packet dropouts and occasionally missing control inputs subject to multiple sampling periods." In: *Journal of the Franklin Institute* 354.12 (2017), pp. 4675–4696. DOI: 10.1016/j.jfranklin.2017.04.016.

[88] R. Moraes, F. Vasques, and P. Portugal. "A TDMA-based mechanism to enforce real-time behavior in WiFi networks." In: *2008 IEEE International Workshop on Factory Communication Systems*. 2008, pp. 109–112. DOI: 10.1109/WFCS.2008.4638758.

[89] A. Morell, X. Vilajosana, J. L. Vicario, and T. 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.

[90] E. Municio and S. Latré. "Decentralized Broadcast-Based Scheduling for Dense Multi-Hop TSCH Networks." In: *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*. Association for Computing Machinery, 2016, pp. 19–24. DOI: 10.1145/2980137.2980143.

[91] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, et al. "Addressing Burstiness for Reliable Communication and Latency Bound Generation in Wireless Sensor Networks." In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 2010, pp. 303–314. DOI: 10.1145/1791212.1791248.

[92] Z. Music, F. Molinari, S. Gallenmüller, O. Ayan, S. Zoppi, et al. “Design of a Networked Controller for a Two-Wheeled Inverted Pendulum Robot.” In: *IFAC-PapersOnLine* 52.20 (2019), pp. 169–174. DOI: 10.1016/j.ifacol.2019.12.153.

[93] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman. “Real-Time Control of a Two-Wheeled Inverted Pendulum Mobile Robot.” In: *International Journal of Electrical and Computer Engineering* 2.3 (2008), pp. 406–412. DOI: 10.5281/zenodo.1332206.

[94] M. J. Neely. “Stochastic network optimization with application to communication and queueing systems.” In: *Synthesis Lectures on Communication Networks* 3.1 (2010). DOI: 10.2200/S00271ED1V01Y201006CNT007.

[95] H. G. Nguyen, J. Morrell, K. D. Mullens, A. B. Burmeister, S. Miles, et al. “Segway robotic mobility platform.” In: *Mobile Robots XVII*. Vol. 5609. SPIE, 2004, pp. 207–220. DOI: 10.1117/12.571750.

[96] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia. “Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks.” In: *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*. 2012, pp. 327–332. DOI: 10.1109/PIMRC.2012.6362805.

[97] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, et al. “On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH.” In: *IEEE Sensors Journal* 13.10 (2013), pp. 3655–3666. DOI: 10.1109/JSEN.2013.2266417.

[98] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, et al. “On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks.” In: *IEEE Sensors Journal* 16.2 (2016), pp. 550–560. DOI: 10.1109/JSEN.2015.2480886.

[99] A. K. Parekh and R. G. Gallager. “A generalized processor sharing approach to flow control in integrated services networks: the single-node case.” In: *IEEE/ACM Transactions on Networking* 1.3 (June 1993). DOI: 10.1109/90.234856.

[100] B. Park, J. Nah, J.-Y. Choi, I.-J. Yoon, and P. Park. “Transmission Scheduling Schemes of Industrial Wireless Sensors for Heterogeneous Multiple Control Systems.” In: *Sensors* 18.12 (2018), p. 4284. DOI: 10.3390/s18124284.

[101] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson. “Wireless Network Design for Control Systems: A Survey.” In: *IEEE Communications Surveys Tutorials* 20.2 (2018), pp. 978–1013. DOI: 10.1109/COMST.2017.2780114.

[102] P. Park, P. Di Marco, C. Fischione, and K. H. Johansson. "Modeling and Optimization of the IEEE 802.15.4 Protocol for Reliable and Timely Communications." In: *IEEE Transactions on Parallel and Distributed Systems* 24.3 (2013), pp. 550–564. DOI: 10.1109/TPDS.2012.159.

[103] G. Patti and L. Lo Bello. "A Priority-Aware Multichannel Adaptive Framework for the IEEE 802.15.4e-LLDN." In: *IEEE Transactions on Industrial Electronics* 63.10 (2016), pp. 6360–6370. DOI: 10.1109/TIE.2016.2573754.

[104] J. Pesonen, H. Zhang, P. Soldati, and M. Johansson. "Methodology and tools for controller-networking codesign in WirelessHART." In: *2009 IEEE Conference on Emerging Technologies Factory Automation*. 2009, pp. 1–8. DOI: 10.1109/ETFA.2009.5347006.

[105] J. Polastre, J. Hill, and D. Culler. "Versatile Low Power Media Access for Wireless Sensor Networks." In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. Association for Computing Machinery, 2004, pp. 95–107. DOI: 10.1145/1031495.1031508.

[106] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[107] F. Qin, X. Dai, and J. E. Mitchell. "Effective-SNR estimation for wireless sensor network using Kalman filter." In: *Ad Hoc Networks* 11.3 (May 2013), pp. 944–958. DOI: 10.1016/j.adhoc.2012.11.002.

[108] L. Qiu, Q. Luo, F. Gong, S. Li, and B. Xu. "Stability and stabilization of networked control systems with random time delays and packet dropouts." In: *Journal of the Franklin Institute* 350.7 (2013), pp. 1886–1907. DOI: 10.1016/j.jfranklin.2013.05.013.

[109] D. V. Queiroz, M. S. Alencar, R. D. Gomes, I. E. Fonseca, and C. 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.

[110] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. "Energy-Efficient Collision-Free Medium Access Control for Wireless Sensor Networks." In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. 2003, pp. 181–192. DOI: 10.1145/958491.958513.

[111] A. S. Rammurthy. "Experimental Evaluation of Statistical Delay Bound for IEEE 802.15.4 TSCH Networks." MA. Thesis. Chair of Communication Networks, Department of Electrical and Computer Engineering, Technical University of Munich, Aug. 2019.

[112] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, and L. A. Grieco. “Autonomous and traffic-aware scheduling for TSCH networks.” In: *Computer Networks* 135 (Apr. 2018), pp. 201–212. DOI: 10.1016/j.comnet.2018.02.023.

[113] Y. Sadi and S. Coleri Ergen. “Energy and Delay Constrained Maximum Adaptive Schedule for Wireless Networked Control Systems.” In: *IEEE Transactions on Wireless Communications* 14.7 (2015), pp. 3738–3751. DOI: 10.1109/TWC.2015.2411602.

[114] Y. Sadi and S. Coleri Ergen. “Joint optimization of communication and controller components of wireless networked control systems.” In: *IEEE International Conference on Communications*. Vol. 2015-Septe. Institute of Electrical and Electronics Engineers Inc., Sept. 2015, pp. 6487–6493. DOI: 10.1109/ICC.2015.7249358.

[115] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. “End-to-End Communication Delay Analysis in Industrial Wireless Networks.” In: *IEEE Transactions on Computers* 64.5 (2015), pp. 1361–1374. DOI: 10.1109/TC.2014.2322609.

[116] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. “Real-Time Scheduling for WirelessHART Networks.” In: *2010 31st IEEE Real-Time Systems Symposium*. 2010, pp. 150–159. DOI: 10.1109/RTSS.2010.41.

[117] A. Saifullah, P. Babu Tiwari, B. Li, C. Lu, and P. Babu. *Accounting for Failures in Delay Analysis for WirelessHART Accounting for Failures in Delay Analysis for WirelessHART Networks Networks Recommended Citation Recommended Citation "Accounting for Failures in Delay Analysis for WirelessHART Networks " Report*. Tech. rep. Jan. 2012, pp. 2012–2028. DOI: 10.7936/K7X0658J.

[118] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. “End-to-end delay analysis for fixed priority scheduling in WirelessHART networks.” In: *Real-Time Technology and Applications - Proceedings*. 2011, pp. 13–22. DOI: 10.1109/RTAS.2011.10.

[119] L. Seno, F. Tramarin, and S. Vitturi. “Performance of Industrial Communication Systems: Real Application Contexts.” In: *IEEE Industrial Electronics Magazine* 6.2 (2012), pp. 27–37. DOI: 10.1109/MIE.2012.2193292.

[120] S. P. Shantharam. “Development and Implementation of WSN TSCH Scheduler for NCS.” MA. Thesis. Chair of Communication Networks, Department of Electrical and Computer Engineering, Technical University of Munich, Dec. 2019.

[121] C. Shih, A. E. Xhafa, and J. Zhou. “Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks.” In: *2015 IEEE International Conference on Communications (ICC)*. 2015, pp. 6494–6499. DOI: 10.1109/ICC.2015.7249359.

[122] R. Singh and P. R. Kumar. "Throughput Optimal Decentralized Scheduling of Multihop Networks With End-to-End Deadline Constraints: Unreliable Links." In: *IEEE Transactions on Automatic Control* 64.1 (2019), pp. 127–142. DOI: 10.1109/TAC.2018.2874671.

[123] P. Soldati, H. Zhang, Z. Zou, and M. Johansson. "Optimal Routing and Scheduling of Deadline-Constrained Traffic over Lossy Networks." In: *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. 2010, pp. 1–6. DOI: 10.1109/GLOCOM.2010.5683716.

[124] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, et al. "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control." In: *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. 2008, pp. 377–386. DOI: 10.1109/RTAS.2008.15.

[125] R. Soua, E. Livolant, and P. Minet. "MUSIKA: A multichannel multi-sink data gathering algorithm in wireless sensor networks." In: *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2013, pp. 1370–1375. DOI: 10.1109/IWCMC.2013.6583756.

[126] R. Soua, P. Minet, and E. Livolant. "DiSCA: A distributed scheduling for convergecast in multichannel wireless sensor networks." In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015, pp. 156–164. DOI: 10.1109/INM.2015.7140288.

[127] R. Soua, P. Minet, and E. Livolant. "MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks." In: *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*. 2012, pp. 91–100. DOI: 10.1109/PCCC.2012.6407742.

[128] R. Soua and P. Minet. "Multichannel assignment protocols in wireless sensor networks: A comprehensive survey." In: *Pervasive and Mobile Computing* 16 (2015), pp. 2–21. DOI: 10.1016/j.pmcj.2014.04.004.

[129] R. Soua, P. Minet, and E. Livolant. "Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks." In: *Transactions on Emerging Telecommunications Technologies* 27.4 (2016), pp. 557–575. DOI: 10.1002/ett.2991.

[130] P. Suriyachai, U. Roedig, and A. Scott. "A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks." In: *IEEE Communications Surveys and Tutorials* 14.2 (2012), pp. 240–264. DOI: 10.1109/SURV.2011.020211.00036.

[131] N. Taheri Javan, M. Sabaei, and V. Hakami. “IEEE 802.15.4.e TSCH-Based Scheduling for Throughput Optimization: A Combinatorial Multi-Armed Bandit Approach.” In: *IEEE Sensors Journal* 20.1 (2020), pp. 525–537. DOI: 10.1109/JSEN.2019.2941012.

[132] L. Tang, M. Liu, K. C. Wang, Y. Huang, F. Yang, et al. “Study of path loss and data transmission error of IEEE 802.15.4 compliant wireless sensors in small-scale manufacturing environments.” In: *International Journal of Advanced Manufacturing Technology* 63.5-8 (Nov. 2012), pp. 659–669. DOI: 10.1007/s00170-012-3928-3.

[133] L. Tassiulas and A. Ephremides. “Dynamic server allocation to parallel queues with randomly varying connectivity.” In: *IEEE Transactions on Information Theory* 39.2 (1993), pp. 466–478. DOI: 10.1109/18.212277.

[134] L. Tassiulas and A. Ephremides. “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks.” In: *29th IEEE Conference on Decision and Control*. 1990, 2130–2132 vol.4. DOI: 10.1109/CDC.1990.204000.

[135] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens. “Dependable Interference-Aware Time-Slotted Channel Hopping for Wireless Sensor Networks.” In: *ACM Trans. Sen. Netw.* 14.1 (Jan. 2018). DOI: 10.1145/3158231.

[136] G. Tian, S. Camtepe, and Y. Tian. “A Deadline-Constrained 802.11 MAC Protocol With QoS Differentiation for Soft Real-Time Control.” In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 544–554. DOI: 10.1109/TII.2016.2520398.

[137] E. Toscano and L. Lo Bello. “A middleware for reliable soft real-time communication over IEEE 802.11 WLANs.” In: *SIES 2011 - 6th IEEE International Symposium on Industrial Embedded Systems, Conference Proceedings*. 2011, pp. 115–122. DOI: 10.1109/SIES.2011.5953653.

[138] A. Ulusoy, O. Gurbuz, and A. Onat. “Wireless model-based predictive networked control system over cooperative wireless network.” In: *IEEE Transactions on Industrial Informatics* 7.1 (Feb. 2011), pp. 41–51. DOI: 10.1109/TII.2010.2089059.

[139] M. Vilgelm, M. Gürsu, S. Zoppi, and W. Kellerer. “Time Slotted Channel Hopping for Smart Metering: Measurements and Analysis of Medium Access.” In: *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Nov. 2016, pp. 109–115. DOI: 10.1109/SmartGridComm.2016.7778747.

[140] N. Wang and Y. Zhang. "Co-design of Scheduling and Control for Networked Control Systems with Time-Delay and Communication Constraints." In: *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*. Vol. 2. 2015, pp. 256–259. DOI: 10.1109/IHMSC.2015.81.

[141] Q. Wang, K. Jaffrès-Runser, Y. Xu, and J. Scharbarg. "A certifiable resource allocation for real-time multi-hop 6TiSCH wireless networks." In: *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*. 2017, pp. 1–9. DOI: 10.1109/WFCS.2017.7991957.

[142] T. Watteyne, A. Mehta, and K. Pister. "Reliability through Frequency Diversity: Why Channel Hopping Makes Sense." In: *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. Association for Computing Machinery, 2009, pp. 116–123. DOI: 10.1145/1641876.1641898.

[143] Wei Ye, J. Heidemann, and D. Estrin. "Medium access control with coordinated adaptive sleeping for wireless sensor networks." In: *IEEE/ACM Transactions on Networking* 12.3 (2004), pp. 493–506. DOI: 10.1109/TNET.2004.828953.

[144] Wei Zhang, M. S. Branicky, and S. M. Phillips. "Stability of networked control systems." In: *IEEE Control Systems Magazine* 21.1 (2001), pp. 84–99. DOI: 10.1109/37.898794.

[145] Y. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, et al. "RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications." In: *2013 IEEE 34th Real-Time Systems Symposium*. 2013, pp. 140–149. DOI: 10.1109/RTSS.2013.22.

[146] A. Woo and D. Culler. *Evaluation of Efficient Link Reliability Estimators for Low-Power Wireless Networks*. Tech. rep. EECS Department, University of California, Berkeley, 2003.

[147] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, et al. "Analysis of EDF scheduling for Wireless Sensor-Actuator Networks." In: *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. 2014, pp. 31–40. DOI: 10.1109/IWQoS.2014.6914298.

[148] Xi Zhang, Jia Tang, Hsiao-Hwa Chen, Song Ci, and M. Guizani. "Cross-layer-based modeling for quality of service guarantees in mobile wireless networks." In: *IEEE Communications Magazine* 44.1 (2006), pp. 100–106. DOI: 10.1109/MCOM.2006.1580939.

[149] Xiangheng Liu and A. Goldsmith. “Wireless network design for distributed control.” In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 3. 2004, 2823–2829 Vol.3. DOI: 10.1109/CDC.2004.1428892.

[150] W. Yadong and D. Shihong. “Link layer time-varying model for IEEE 802.15.4 radio in industrial environment.” In: *International Journal of Distributed Sensor Networks* 2014 (Dec. 2014). DOI: 10.1155/2014/240256.

[151] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, et al. “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. DOI: 10.1109/TIE.2015.2402642.

[152] T. C. Yang. “Networked control system: A brief survey.” In: *IEE Proceedings: Control Theory and Applications* 153.4 (July 2006), pp. 403–412. DOI: 10.1049/ip-cta:20050178.

[153] K. Yu, M. Gidlund, J. Åkerberg, and M. Björkman. “Performance Evaluations and Measurements of the REALFLOW Routing Protocol in Wireless Industrial Networks.” In: *IEEE Transactions on Industrial Informatics* 13.3 (2017), pp. 1410–1420. DOI: 10.1109/TII.2016.2587842.

[154] Y. Yuan, Z. Yang, Z. He, and J. He. “An integrated energy aware wireless transmission system for QoS provisioning in wireless sensor network.” In: *Computer Communications* 29.2 (2006), pp. 162–172. DOI: 10.1016/j.comcom.2005.05.015.

[155] M. Z. Zamalloa and B. Krishnamachari. “An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links.” In: *ACM Transactions on Sensor Networks* 3.2 (June 2007), 7–es. DOI: 10.1145/1240226.1240227.

[156] L. Zhang, H. Gao, and O. Kaynak. “Network-Induced Constraints in Networked Control Systems—A Survey.” In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 403–416. DOI: 10.1109/TII.2012.2219540.

[157] X. Zhang, Q. Han, X. Ge, D. Ding, L. Ding, et al. “Networked control systems: a survey of trends and techniques.” In: *IEEE/CAA Journal of Automatica Sinica* 7.1 (2020), pp. 1–17. DOI: 10.1109/JAS.2019.1911651.

[158] M. Zheng, W. Liang, H. Yu, and Y. Xiao. “Performance Analysis of the Industrial Wireless Networks Standard: WIA-PA.” In: *Mobile Networks and Applications* 22.1 (Feb. 2017), pp. 139–150. DOI: 10.1007/s11036-015-0647-7.

[159] D. Zorbas, V. Kotsiou, F. Théoleyre, G. Z. Papadopoulos, and C. Douligeris. "LOST: Localized blacklisting aware scheduling algorithm for IEEE 802.15.4-TSCH networks." In: *2018 Wireless Days (WD)*. 2018, pp. 110–115. DOI: 10.1109/WD.2018.8361705.

[160] R. Zurawski. *Industrial communication technology handbook*. CRC Press, 2014.

[161] B. Zurita Ares, P. G. Park, C. Fischione, A. Speranzon, and K. H. Johansson. "On power control for wireless sensor networks: System model, middleware component and experimental evaluation." In: *2007 European Control Conference (ECC)*. 2007, pp. 4293–4300. DOI: 10.23919/ECC.2007.7069054.

Acronyms

3GPP 3rd Generation Partnership Project 31

ACK Acknowledgment 37

AGV Automatic Guided Vehicles 16

ALOHA Additive Links On-line Hawaii Area 35

AMC Adaptive Modulation and Coding 31

AP Access Point 21, 24, 44, 45, 46, 47, 48, 49, 53, 55, 56, 57, 108, 109, 142, 143, 145

ASN Absolute Sequence Number 37, 46, 100

BER Bit Error Rate 18, 19, 22, 23, 40, 141

BP Backpressure 8, 79, 80, 81

C.I. Confidence Interval 24

C2A Controller-to-Actuator 97, 98, 100, 101, 144

CA Collision Avoidance 35, 111

CCA Clear Channel Assessment 36

CD Collision Detection 35

CDF Cumulative Distribution Function 53, 54, 55, 56, 57, 69, 143

CMA Cumulative Moving Average 25, 27, 40, 44

CPS Cyber-Physical Systems 89

CRC Cyclic Redundancy Check 19

CSMA Carries Sense Multiple Access 35, 111

CSP Constrained Shortest Path 51

DSSS Direct Sequence Spread Spectrum 14, 15, 18, 26

DVP Delay Violation Probability iii, iv, v, vi, 7, 8, 9, 10, 11, 59, 60, 61, 62, 64, 66, 67, 68, 72, 73, 76, 77, 78, 79, 80, 81, 82, 83, 84, 99, 116, 117, 143, 144

DVPUB Delay Violation Probability Upper Bound 68, 69, 72, 79

E2E End-to-End iv, vi, 4, 10, 94, 97, 98, 100, 101, 114, 116, 144

EB Enhanced Beacon 37, 38, 52, 68

EWMA Exponentially Weighted Moving Average 25, 26, 27, 28, 40, 52, 53, 54, 58, 116, 142, 146

FHSS Frequency Hopping Spread Spectrum 15

FIFO First in, first out 40

GUI Graphical User Interface 86

IAE Integrated Absolute Error 93

IEEE Institute of Electrical and Electronics Engineers 15, 22, 23, 24, 25, 26, 36, 53, 86, 96, 100, 141, 142

IP Internet Protocol 105, 108

ISM Industrial Scientific and Medical 3, 14, 16, 24, 26, 40, 44, 53, 100

IWSN Industrial Wireless Sensor Networks iii, iv, v, vi, 1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14, 15, 16, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 44, 45, 46, 49, 50, 52, 53, 57, 58, 60, 61, 63, 64, 70, 84, 85, 86, 87, 88, 89, 94, 95, 96, 97, 99, 100, 101, 102, 114, 115, 116, 117, 141, 142

LPL Low Power Listening 36

LQI Link Quality Information 53, 54, 143

LQR Linear Quadratic Regulator 91, 95, 107

LSP Loop Success Probability 9, 10, 12, 94, 95, 96, 102, 114, 116, 144

LTE Long Term Evolution 31

LTI Linear Time-Invariant 21, 85

MAC Medium Access Control 3, 29, 31, 34, 35, 36, 37, 38, 86, 87, 89, 100, 109, 142

MDP Markov Decision Process 73, 75, 76, 79, 80, 81

MIMO Multiple-Input Multiple-Output 30, 31

MSE Mean Squared Error 26, 27, 28, 142, 146

MT Maximum Throughput 31

MW Max Weight 8, 79, 80, 81

NC Network Coordinator 1, 3, 4, 6, 28, 29, 30, 36, 37, 38, 49, 50, 52, 59, 63, 64, 67, 68, 69, 75, 142

NCS Networked Control Systems iii, iv, v, vi, 2, 3, 4, 7, 9, 10, 11, 12, 58, 64, 84, 85, 86, 87, 88, 89, 90, 93, 94, 97, 99, 102, 103, 104, 105, 106, 108, 109, 111, 113, 114, 115, 116, 117, 143, 145, 146

NIP Networked Inverted Pendulum iv, vi, 85, 86, 90, 92, 93, 94, 95, 96, 99, 105, 106, 107, 108, 110, 114, 116, 144, 145

NPI Noise Plus Interference 20, 21, 22, 28, 141

OQPSK Offset Quadrature Phase Shift Keying 15, 18

OSI Open Systems Interconnection 104, 108, 115

PDR Packet Delivery Ratio 24, 25, 27, 33, 38, 50, 52, 53, 54, 56

PER Packet Error Rate iii, v, 4, 5, 6, 7, 10, 13, 14, 15, 18, 19, 20, 22, 23, 24, 26, 27, 28, 29, 30, 33, 35, 38, 40, 43, 44, 46, 50, 52, 53, 54, 58, 76, 96, 99, 100, 115, 116, 117, 141

PF Proportional Fair 31

PSD Power Spectral Density 25, 26, 142

QoC Quality of Control iv, vi, 2, 4, 9, 10, 12, 85, 86, 89, 90, 93, 94, 95, 96, 99, 102, 103, 105, 109, 110, 112, 113, 114, 116, 117, 146

QoS Quality of Service iii, iv, v, vi, 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 19, 23, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 49, 50, 52, 55, 56, 57, 58, 59, 62, 63, 65, 84, 85, 86, 88, 89, 90, 95, 96, 99, 102, 103, 109, 113, 114, 115, 116, 117

RB Resource Block 37, 38, 40, 41, 50, 51, 52, 99, 142

RF Radio Frequency 14, 20, 21, 22, 23, 40

RPL Routing Protocol for Low-Power and Lossy Networks 33

RRM Radio Resource Management iii, iv, v, vi, 3, 4, 5, 6, 7, 9, 10, 13, 14, 19, 23, 26, 28, 29, 30, 31, 32, 34, 35, 36, 38, 39, 49, 57, 58, 59, 65, 85, 86, 97, 99, 115, 116, 117

RSSI Received Signal Strength Indicator 5, 20, 21, 22, 23, 28, 141

S2C Sensor-to-Controller 97, 98, 100, 101, 144

SEP Schedule Exchange Protocol 36

SINR Signal-to-Interference-and-Noise-Ratio 5, 17, 18, 19, 20, 21, 22, 23, 40, 141

SMA Simple Moving Average 25, 26, 27, 28, 40, 142

SNC Stochastic Network Calculus 66

SNR Signal-to-Noise-Ratio 22

SQP Sequential Quadratic Programming 71

Std. Standard 15

TCP Transmission Control Protocol 105, 108

TCP/IP Transmission Control Protocol/Internet Protocol 30, 40

TDMA Time Division Multiple Access 3, 7, 29, 35, 36, 37, 38, 40, 41, 46, 47, 48, 52, 53, 88, 89, 96, 98, 99, 101, 142

TSCH Time Slotted Channel Hopping 3, 6, 29, 30, 31, 32, 33, 36, 40, 41, 96, 99, 100, 101, 102, 141

UDP User Datagram Protocol 44

WFQ Weighted-Fair Queuing 8, 79, 80, 81

WLAN Wireless Local-Area Network 3, 9, 14, 21, 24, 25, 26, 40, 44, 45, 46, 47, 49, 53, 55, 56, 57, 85, 87, 88, 105, 108, 109, 111, 113, 114, 116, 142, 143, 145

WMEWMA Window Mean with Exponentially Weighted Moving Average 25, 27, 28, 34

WSN Wireless Sensor Networks iii, v, 1, 13, 14, 20, 24, 30, 36, 46, 63

WTB Wireless Transient Bound 69, 70, 72, 79

List of Figures

List of Tables