Fast, Reliable, Low-power Wireless Monitoring and Control with Concurrent Transmissions

Matteo Trobinger

Advisor	**Prof. Gian Pietro Picco** University of Trento, Italy
Committee	**Prof. Carlo Alberto Boano** Graz University of Technology, Austria **Prof. Chenyang Lu** Washington University in St. Louis, USA **Dr. Philipp Sommer** ABB Research in Baden, Switzerland

University of Trento
Trento, Italy
July 2021

To my parents and girlfriend.

Abstract

Low-power wireless technology is a part and parcel of our daily life, shaping the way in which we behave, interact, and more generally live. The ubiquity of cheap, tiny, battery-powered devices augmented with sensing, actuation, and wireless communication capabilities has given rise to a "smart" society, where people, machines, and objects are seamlessly interconnected, among themselves and with the environment. Behind the scenes, low-power wireless protocols are what enables and rules all interactions, organising these embedded devices into wireless networks, and orchestrating their communications.

The recent years have witnessed a persistent increase in the pervasiveness and impact of low-power wireless. After having spawned a wide spectrum of powerful applications in the consumer domain, low-power wireless solutions are extending their influence over the industrial context, where their adoption as part of feedback control loops is envisioned to revolutionise the production process, paving the way for the Fourth Industrial Revolution. However, as the scale and relevance of low-power wireless systems continue to grow, so do the challenges posed to the communication substrates, required to satisfy ever more strict requirements in terms of reliability, responsiveness, and energy consumption. Harmonising these conflicting demands is far beyond what is enabled by current network stacks and control architectures; the need to timely bridge this gap has spurred a new wave of interest in low-power wireless networking, and directly motivated our work. In this thesis, we take on this challenge with a main conceptual and technical tool: concurrent transmissions (CTX), a technique that, by enforcing nodes to transmit concurrently, has been shown to unlock unprecedented fast, reliable, and energy efficient multi-hop communications in low-power wireless networks, opening new opportunities for protocol design.

We first direct our research endeavour towards industrial applications, focusing on the popular IEEE 802.15.4 narrowband PHY layer, and advance the state of the art along two different directions: interference resilience and aperiodic wireless control. We tackle radio-frequency noise by extensively analysing, for the first time, the dependability of CTX under different types, intensities, and distributions of reproducible interference patterns, and by devising techniques to push it further. Specifically, we concentrate on CRYSTAL, a recently proposed communication protocol that relies on CTX to rapidly and dependably collect aperiodic traffic. By integrating channel hopping and noise detection in the protocol operation, we provide a novel communication stack capable of supporting aperiodic transmissions with near-perfect reliability and a per-mille radio duty cycle despite harsh external interference. These results lay the ground towards the exploitation of CTX for aperiodic wireless control; we explore this research direction by co-designing the Wireless Control Bus (WCB), our second contribution. WCB is a clean-slate CTX-based communication stack tailored to event-triggered control (ETC), an aperiodic control strategy holding the capability to significantly improve the efficiency of wireless control systems, but whose real-world impact has been hampered by the lack of appropriate networking support. Operating in conjunction with ETC, WCB timely and dynamically adapts the network operation to the control demands, unlocking an order-of-magnitude reduction in energy costs w.r.t. traditional periodic approaches while retaining the same control performance, therefore unleashing and concretely demonstrating the true ETC potential for the first time.

Nevertheless, low-power wireless communications are rapidly evolving, and new radios striking novel trade-offs are emerging. Among these, in the second part of the thesis we focus on ultra-wideband (UWB). By providing hitherto missing networking primitives for multi-hop dissemination and collection over UWB, we shed light on the communication potentialities opened up by the high data throughput, clock precision, and noise resilience offered by this technology. Specifically, as a third contribution, we demonstrate that CTX not only can be successfully exploited for multi-hop UWB communications but, once embodied in a full-fledged system, provide reliability and energy performance akin to narrowband. Furthermore, the higher data rate and clock resolution of UWB chips unlock up to 80% latency reduction w.r.t. narrowband CTX, along with orders-of-magnitude improvements in network-wide time synchronization. These results showcase how UWB CTX could significantly benefit a multitude of applications, notably including low-power wireless control. With Weaver, our last contribution, we make an additional step towards this direction, by supporting the key functionality of data collection with an ultra-fast convergecast stack for UWB. Challenging the internal mechanics of CTX, Weaver interleaves data and acknowledgements flows in a single, self-terminating network-wide flood, enabling the concurrent collection of different packets from multiple senders with unprecedented latency, reliability, and energy efficiency.

Overall, this thesis pushes forward the applicability and performance of low-power wireless, by contributing techniques and protocols to enhance the dependability, timeliness, energy efficiency, and interference resilience of this technology. Our research is characterized by a strong experimental slant, where the design of the systems we propose meets the reality of testbed experiments and evaluation. Via our open-source implementations, researchers and practitioners can directly use, extend, and build upon our contributions, fostering future work and research on the topic.

Keywords:
Concurrent transmissions, low-power wireless, ultra-wideband, cyber-physical systems, wireless control systems, Industrial Internet of Things, event-triggered control, interference resilience.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Gian Pietro Picco. The first time I had the pleasure to collaborate with him, when I was working on my M.Sc. thesis, I remained impressed by the passion, enthusiasm, and dedication with whom he carried out research. Quite some years have passed since then, but each time I'm engaged in a meeting or in writing a paper with him, my astonishment is still the same. His attitude has been a source of inspiration and motivation for me. Furthermore, throughout this PhD, he helped and supported me in several ways, demonstrating a rare patience during the tough periods, while pushing me when it was probably needed.

Next, I want to thank Carlo Alberto Boano, Chenyang Lu, and Philipp Sommer for serving in my examination committee, and for the precious feedbacks and comments they already provided about my work. I'm truly honoured to have them in my committee.

I express here my heartfelt gratitude to Timofei Istomin. He has been a friend and a mentor for me. Having the opportunity to share this research journey with him has been a honour and a big luck. With his outstanding knowledge, technical skills, and willingness, Timofei has substantially contributed to my research.

A special thanks go to all the members of the D3S group. In particular, I'm grateful to Amy Lynn Murphy for her active role in the contributions presented in the first part of this thesis, and for the many advices she dispensed during these years. Davide Vecchia and Diego Lobba have been other two key figures in my PhD experience. It has been very nice and motivating for me to directly collaborate with them, sharing both nice moments and hectic deadline periods. Davide Molteni provided a huge help for my research, developing and maintaining the low-power wireless testbed whose availability has been a key asset for many of the works presented in this dissertation. I want to also thank Pablo Corbalán for keeping me company late at night in the university, for his many valuable suggestions about research and life, for sharing with me some of his knowledge about Python and Pandas, and for the nice moments we spent together.

I'm also truly thankful to Manuel Mazo Jr. and the members of his research group at TU Delft for nicely hosting me in The Netherlands as a visiting PhD student. They immediately made me part of their research ecosystem; I really much enjoyed all the numerous discussions we had about research, politics, and life. Although unfortunately short, that period has been very stimulating for me.

I want to mention here also the friends that made my days in Povo a bit more funny and less stressful: Fiore, Quiri, Nick, Genc, Mesay, Yu Yu, Giulio, Giacomino, ...

thanks for keeping me company at lunch, during our breaks, and also in Trento at night.

Last but not least, I want to express my deepest gratitude to my parents, my sister, and my beautiful girlfriend Jessica for their unconditioned love and support. They have been a real driving force in this experience, remaining at my side through every high and every low, accepting my sometimes crazy rhythms, helping me to get out of the tough periods, and pushing me to become more proud, aware, and glad of my accomplishments.

Trento, Italy
14 July 2021

Matteo Trobinger

List of Publications

Journals

1. M. Trobinger, G. de Albuquerque Gleizer, T. Istomin, M. Mazo Jr., A. L. Murphy, and G. P. Picco. "The Wireless Control Bus: Enabling Efficient Multi-hop Event-Triggered Control with Concurrent Transmissions". Submitted to *ACM Transactions on Cyber-Physical Systems (TCPS)*. 2021. arXiv: arXiv:2101.10961 [eess.SY]. (Chapter 4)

International Conferences

1. T. Istomin, M. Trobinger, A. L. Murphy, and G. P. Picco. "Interference-resilient Ultra-low Power Aperiodic Data Collection". In *Proc. of the* 17*th* *ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN. 2018. DOI: `10.1109/IPSN.2018.00015`. (Chapter 3)
2. D. Lobba, M. Trobinger, D. Vecchia, T. Istomin, and G. P. Picco. "Concurrent Transmissions for Multi-hop Communication on Ultra-wideband Radios". In *Proc. of the* 17*th* *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2020. (Chapter 6)
3. M. Trobinger, D. Vecchia, D. Lobba, T. Istomin, and G. P. Picco. "One Flood to Route Them All: Ultra-fast Convergecast of Concurrent Flows over UWB". In *Proc. of the* 18*th* *ACM Conference on Embedded Networked Sensor Systems*. SenSys. 2020. (Chapter 7)

EWSN Dependability Competition

1. M. Trobinger, T. Istomin, A. L. Murphy, and G. P. Picco. "Competition: CRYSTAL Clear: Making Interference Transparent". In *Proc. of the* 15*th* *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2018.
   Second place. (Chapter 3)
2. M. Trobinger, T. Istomin, A. L. Murphy, and G. P. Picco. "Competition: CRYSTAL". In *Proc. of the* 16*th* *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2019. **Third place in Category II.** (Chapter 3)

Contents

List of Figures

List of Tables

1
Introduction

The world is rapidly and increasingly evolving towards a higher level of automation and connectivity. We live in a digital society where people, machines, and objects are seamlessly interconnected, to an extent that was not only hard to predict, but even imagine just a few decades ago. Low-power wireless technology, and the underlying networking protocols orchestrating communications, are a key enabler of this quiet but *disruptive* revolution.

The rise of low-power wireless technology. Underpinned by a wave of research interest from both academia and industry, in about twenty years low-power wireless communications in the form of wireless networks of tiny, battery-powered, embedded devices with processing, sensing, and actuation capacities have undergone a radical transformation. From the pioneering vision of "Smart Dust" [4], they have rapidly evolved into a well-established technology, proven to be very dependable and energy-efficient. The proliferation of standards [5, 6, 7, 8, 9], protocols [9, 10], and real-world deployments [11, 12] practically demonstrating the applicability and versatility of low-power wireless, as well as the benefits it bears w.r.t. traditional wired solutions, have been a stepping stone towards the widespread adoption of low-power wireless solutions. Higher scalability and (re)placement flexibility, lower installation and maintenance costs, support for mobility, and finer-grained sensing and actuation capabilities are only some of these advantages.

Similarly, the recent availability of a plethora of new radio chips and PHY layers providing very different trade-offs in terms of bandwidth, data rate, range of communication, and energy consumption have further broadened the landscape of systems that can now benefit from this technology, and in turn have attracted additional interest in this research field.

For many years, the main application use case of low-power wireless networking has been network-wide data collection, e.g., from many sensors to one or more sinks, a scenario where extending battery lifetime to enable long-term, reliable operation of the deployed wireless network was the main goal. Recently, however, this staple use case has made way for ever more winsome and high-impact uses of low-power wireless communications. Low-power wireless systems are nowadays the leading solution in a wide domain of applications encompassing wildlife tracking [13], smart cities [14], cyber-physical systems [15], contact tracing [16], and the Internet of Things (IoT) at large [17]. Furthermore, this technology is becoming appealing also in traditionally wired domains like industrial control [18, 19, 20]. The inclusion of low-power wireless networks as part of feedback control loops is at the basis of the Fourth Industrial Revolution [21, 22, 23] (also known with the moniker of Industry 4.0 or Industrial IoT), which is envisioned to drastically reshape and optimise production processes, and similarly promises to lead the largest economic income for IoT.

Networking challenges. However, hand in hand with the rise in pervasiveness of low-power wireless technology, the complexity of the requirements posed to the communication substrates orchestrating the nodes' interactions have substantially increased. Best effort reliability and latency demands, common to most of the monitoring applications that characterized the dawn of the IoT era, have been rapidly replaced by ever more stringent dependability and timeliness guarantees, which become wired-like in the industrial context [22, 24, 25]. Delivering packets with near-perfect reliability across multi-hop networks and meeting hard end-to-end deadlines are prerequisites for low-power wireless stacks targeting industrial applications, as control loops demand real-time guarantees from the communication system. Losing packets or missing deadlines might be harmful for the production chain, and even dangerous for the safety of the workers. Preserving energy efficiency is also of utmost importance; many Industrial IoT (IIoT) applications require battery-powered devices to work for several years, unattended, as replacing batteries is expensive and might be impractical. Matters are further complicated by external interference; industrial plants are often characterized by strong radio-frequency noise that, by hindering node-to-node transmissions, hampers the dependability, latency, and energy consumption of low-power wireless protocols [26, 27]. Finally, to close feedback control loops, wireless control systems (WCSs) demand support for a variety of traffic patterns, beyond the simple periodic data collection that thus far has catalysed the majority of the efforts of the low-power wireless community.

The challenge of harmonising these conflicting requirements has laid bare the limitations inherent in current low-power wireless protocol stacks and control architectures. This, along with the enormous benefits that unlocking a wider adoption of low-power wireless technology entails, have recently sparkled a new surge of interest in low-power wireless networking, and directly motivated our work.

Goal, contributions, and methodology. In this thesis, our goal is

> *to push the envelope of low-power wireless networking, by exploring strategies to enhance its dependability, timeliness, energy efficiency, and interference resilience, with particular attention to industrial applications and wireless control.*

We achieve this goal by leveraging a main conceptual and technical tool: *concurrent* *transmissions*[1] (CTX), arguably one of the most promising recent advancements in low-power wireless networking [10, 28]. As detailed in Chapter 2, the term refers to the fact that tightly-synchronized simultaneous transmissions do not necessarily result in a collision; instead, under some conditions determined by the underlying PHY radio layer, one of the concurrent packets can be received with very high probability. The adoption of CTX has triggered a paradigm shift in low-power wireless protocol design. Pioneered by Glossy [29], CTX have inspired a breed of systems that intentionally enforce nodes to transmit concurrently and, by embracing interference, achieve unprecedented reliability, latency, and energy efficiency. By enabling fast and dependable low-power wireless communications across multi-hop networks, CTX are a potential asset for closing feedback loops among sensors, actuators, and controller(s) via low-power wireless links; thence, we leverage such technique as the primary building block to achieve our above-mentioned goal.

Further, we broaden and diversify the scope and the impact of our solutions by extending our research effort to two different PHY layers: IEEE 802.15.4 narrowband and ultra-wideband (UWB), which are the reference technology of the first and second part of this thesis, respectively.

Part I directly targets industrial applications, advancing the state-of-the-art in industrial wireless control along two directions. Our first contribution tackles radio-frequency noise, one of the key hampering factors for the dependability of low-power wireless in industrial settings. This aspect is largely unexplored in the literature, including the CTX one. A witness to this is that all staple CTX protocols (e.g., [29, 30, 31, 32]) have been tested only under natural interference, i.e., the one present in office buildings, mostly generated by Wi-Fi access points and Bluetooth devices. Natural interference is *i) relatively mild*, a far cry from the conditions found in industrial settings [11, 33, 34], and *ii) not controllable* by researchers, preventing the repeatability of the experiments [35, 36, 37]. This makes it extremely hard, if not infeasible, to correlate packet loss with specific typologies of interference, and thus devise effective countermeasures enhancing noise resilience. In Chapter 3 we fill this gap by *i)* providing a first, extensive exploration of CTX-based communication under various types, intensities, and distributions of *reproducible* interference patterns, *ii)* identifying strengths and potential limitations thereof, and *iii)* designing strategies to push the dependability of low-power wireless solutions further. Specifically, we studied the recently-proposed Crystal protocol [31], a CTX-based system devised to efficiently deliver aperiodic and sparse data collection traffic, which we considered particularly interesting for different reasons. First, this traffic pattern arguably poses challenging requirements for interference resilience, given the detrimental impact packet loss has on such systems where "every packet counts". Second, aperiodic control strategies are emerging as the way to go for minimising consumption in WCSs [18, 38, 39, 40], making the efficient support of aperiodic communications a necessity.

Once endowed with techniques known to mitigate interference—channel hopping and noise detection—our noise-resistant extension of Crystal achieves performance

[1] Some authors use the label *synchronous transmissions* instead, with equivalent meaning in this context.

akin to the original, *ensuring near-perfect reliability along with a per-mille radio duty cycle with aperiodic, sparse traffic,* even under multiple sources of radio-frequency noise. We validated and further extended our findings to different network topologies and even more challenging interference patterns by participating in the EWSN Dependability Competition [41], where Crystal—notably one of the few systems *not* purposely designed to match the specific competition scenario—got the *second* [3] and *third* [42] place in the 2018 and 2019 edition, respectively. These results suggest that floods of CTX, if carefully orchestrated, can provide the long-awaited reliability, interference resilience, timeliness, and energy efficiency guarantees demanded by aperiodic wireless control systems, fostering the adoption of this technique in the industrial context.

We explore this latter research direction with the Wireless Control Bus (WCB), our second contribution in Chapter 4. WCB is a full-fledged CTX-based networking stack for event-triggered control (ETC) [40, 43, 44, 45, 46], one of the most promising aperiodic control paradigms in the literature. Veering away from classic periodic sampling approaches, ETC systems exploit feedback from the plants to acquire sensor readings and distribute actuation commands only when needed to preserve the stability of the control loop, abating communication demands without impairing control performance. More than a decade of research efforts have settled the theoretical foundations of this adaptive control strategy, demonstrating the energy savings ETC *can* unlock; nevertheless, the lack of networking layers able to transfer its benefits from theory to practice has been a major roadblock towards the real-world adoption of ETC solutions [19]. Efficiently supporting the aperiodic and unpredictable communication ETC induces requires a networking stack capable to dynamically adapt its operations to the suddenly changing ETC demands; a difficult task that the few existing wireless ETC implementations (e.g., [47, 48])—mostly built atop beacon-enabled IEEE 802.15.04 TDMA protocols and tested only in single-hop network—fall short in achieving. Overly conservative protocol designs have hitherto prevented savings in energy and bandwidth in line with the drastic decrease in control traffic offered by the ETC sampling. Our new Wireless Control Bus reverses this trend: by leveraging carefully organised floods of concurrent transmissions, WCB minimizes the network overhead during quiescent, steady-state periods, while ensuring timely and reliable reactions when required by the event-triggered controller to retain control performance. Using a cyber-physical testbed emulating a water distribution system controlled over a real-world multi-hop wireless network, we show that ETC over WCB achieves the *same* quality of periodic control (itself realised over CTX) with an *order-of-magnitude* reduction in energy costs, therefore unleashing and concretely demonstrating the full potential of ETC-based WCS architectures for the first time.

We continue our research endeavour in Part II, approaching our overarching goal of pushing further the boundaries of performance, and consequently the real-world adoption, of low-power wireless from a different perspective. Specifically, we focus on the UWB PHY layer, concisely described in Chapter 5. Whilst not as pervasive as IEEE 802.15.4 narrowband, UWB is rapidly reaching the forefront of IoT enabling technologies, catalysing a great deal of attention thanks to the capability of UWB chips to combine high-rate communications with decimetre-level distance estimation. Major vendors like Apple and Samsung have already equipped their latest products with UWB transceivers [49, 50, 51], a witness to the rising interest this technology is attracting not only within academia, but also in industry. However, whilst localization-centric applications have been the Holy Grail of UWB research [52, 53], the communication facet has been only marginally explored and, to the best of our knowledge, *no* staple UWB protocol stack implementation exists.

UWB communications is exactly the focus of the second part of this thesis. By providing hitherto missing networking primitives for multi-hop dissemination and collection over UWB, we aim at fostering its adoption also for sensing, communication, and control, besides ranging and localization, where we expect UWB technology to have a similar, outstanding impact. For instance, noise- and latency-sensitive applications, like the industrial control systems we concentrate our efforts on in Part I, could largely benefit from the high interference resilience and data rate UWB entails, up to 27 Mbps according to the IEEE 802.15.4-2011 standard [8].

Motivated by the tremendous progress concurrent transmissions have fostered in IEEE 802.15.4 narrowband research, Chapter 6 investigates the applicability of this technique to UWB radios. Specifically, we analyse if and to what extent CTX on UWB can achieve a performance akin to those repeatedly proven in narrowband. This desirable outcome can not be taken a priori for granted, given *i)* the significantly different characteristics of the two PHY layers, and *ii)* the strong dependency of CTX performance on PHY-level phenomena. We adopt a system-driven approach, where techniques and codebases representative of the state-of-the-art (i.e., Glossy and Crystal) are adapted for UWB and evaluated in an indoor testbed yielding multi-hop topologies. This in-depth analysis enables us to directly face the opportunities and challenges in implementing and exploiting UWB CTX, which we distill as guidelines for other researchers to efficiently leverage this powerful synergy. Experimental results show that, once embodied in a full-fledged system, UWB CTX yield benefits similar to narrowband, i.e., near-perfect reliability and low energy consumption, along with a drastic reduction of end-to-end latency, *up to 80% lower than narrowband*. Furthermore, thanks to the higher clock resolution of the DW1000 UWB chip [54, 55] we rely on, our implementation of Glossy over UWB achieves almost *three orders of magnitude* improvements in network-wide time synchronization w.r.t. its narrowband counterpart. These results, along with the qualitative lesson learned, pave the way for the exploitation of concurrent transmissions in UWB, pushing us towards Weaver, our next and last contribution in Chapter 7.

Weaver directly tackles the current lack of a convergecast stack for UWB, and specifically targets *ultra-fast* collection of aperiodic traffic. To minimise latency, Weaver's design departs from the large body of CTX literature built atop Glossy, and explores an alternate, finer-grained approach leveraging *individual* concurrent transmissions rather than *monolithic* Glossy floods. Indeed, exploiting Glossy as a primitive building block leads to an unavoidable drawback: it enforces a sequential, flooding-based protocol operation, where *each* Glossy flood is devoted to deliver *a single* packet network-wide, making the overall latency increase *proportionally* with the number of messages and network size; something we instead avoid in Weaver.

By weaving together multiple data and acknowledgement flows, our system unlocks the *concurrent* collection of *different packets* from multiple senders in a *single, self-terminating*, network-wide flood. Analytical and experimental results confirm that Weaver provides significantly faster and more efficient delivery of concurrent flows than state-of-the-art Glossy-based protocols, e.g., *enabling the dependable collection of 30 simultaneous messages over a 6 hop network in about 100 ms.* Furthermore, it achieves higher reliability, while ensuring resilience to topology changes. This unprecedented performance opens up the door for the adoption of UWB technology in general, and Weaver in particular, for a multitude of applications, notably including wireless control, effectively rejoining the first and second part of this thesis. Moreover, they practically demonstrate the benefits unlocked by the fine-grained use of CTX, an innovative design mindset adaptable to *any* PHY layer supporting concurrent transmissions. The intrinsic flexibility individual CTX provide could be leveraged to design highly customizable solutions, further enhancing the pervasiveness and impact of low-power wireless communication at large. We encourage the adoption of this alternate CTX design paradigm via our Time Slot Manager (TSM) kernel, a reusable component efficiently handling the low-level mechanics of CTX. By ensuring tight synchronization and accurate scheduling of TX and RX operations, TSM facilitates the development of network stacks based on individual CTX, enabling researchers to directly focus on protocol design while empowering them with all the degrees of freedom offered by this technique.

Throughout this dissertation we follow common best practices in experimental system research. The design of the various low-power wireless protocol stacks we propose is corroborated by implementations for popular hardware platforms (i.e., TelosB [56], Firefly [57], and DecaWave EVB1000 [58]) and real-world evaluations of the full-fledged systems. We make all the contributions presented in this thesis openly available [59, 60] for researchers and practitioners to use, extend, and build upon, and generally to inspire future work and research on the topic.

Overall, this thesis pushes forward the applicability and performance of low-power wireless, advancing the state of the art along multiple directions, encompassing interference resilience, aperiodic low-power wireless control, and UWB communications. We foster the adoption of IEEE 802.15.4 narrowband solutions in noise-prone industrial environments by designing and implementing effective techniques for withstanding external interference. We practically demonstrate the potential of ETC sampling strategies, proposing a first communication system capable of unleashing the remarkable energy savings ETC entails. Finally, by providing hitherto missing fast, reliable, and energy efficient protocol stacks for UWB, we broaden the scope and the impact of this technology even outside the localization-centric field where UWB research has been thus far confined. In Chapter 8, which concludes our work, we reflect on these contributions, and provide a detailed outlook on the follow-up opportunities unlocked by our research endeavour.

2 Concurrent Transmissions in Low-power Wireless Communications

The wireless channel is a broadcast medium: upon a transmission, the radio signal propagates through the air potentially reaching any device within a certain communication range. As a side effect, whenever multiple spatially close nodes transmit at the same time and on the same frequency their signals interfere; a receiver in the communication range of different concurrent senders sees a superposition of the transmitted signals.

Link-based communication. For years, such interference was assumed to be destructive, and thus harmful for wireless communications. As suggested by the common wisdom, researchers expected packet collisions to unavoidably lead to unsuccessful receptions of any useful information. A cornerstone rule in designing low-power wireless protocols was therefore to stagger transmissions in time and frequency to prevent neighbouring nodes from TX concurrently, effectively treating the wireless channel as a *set of point-to-point links* instead of a broadcast medium. Under this artificial abstraction, wireless networks were considered as graphs, and routing protocols (e.g., RPL [61]) have been exploited as the only mean to achieve multi-hop communication across such graph-like networks. Different techniques, like channel sensing with backoff periods (e.g., carrier sense multiple access, CSMA), time division multiple access (TDMA) with local or network-wide scheduling, and handshaking, have been proposed to achieve the desired goal of separating transmissions in time, frequency, and space, thus forcing collocated nodes to transmit one after the other, only.

Concurrent transmission. In the last decade a new technique based on the notion of *concurrent transmissions* (CTX) has taken by storm the research field, leading to a paradigm shift in low-power wireless protocol design. The core intuition behind this approach is that allowing multiple nodes to TX concurrently *does not* necessarily lead to a destructive collision; under specific constraints on the timing and the power of the received signals, many types of low-power wireless radios are indeed capable of decoding a message with high probability, despite the superposition of multiple signals from different senders [10, 28, 29]. This phenomenon opens a novel and intriguing perspective for protocol design: leverage the broadcast nature of the wireless channel to construct extremely fast and reliable network floods. Instead of staggering potentially colliding transmissions by constraining communication along artificial point-to-point links, and force the network to acquire and maintain topological information—a time and energy demanding operation, which should be performed frequently given the volatility of wireless links—CTX protocols embrace interference, deliberately expecting nodes to transmit at the same time and on the same frequency channel. This massive concurrency is exploited to *i) minimise multi-hop communication latency*, as packet retransmissions are not governed by channel access policies, but nodes are free to relay messages immediately after reception; *ii) enhance end-to-end reliability*, leveraging the sender diversity and spatial redundancy achieved by spreading packets along multiple paths to increase the chances of correct reception despite link failures and external interference; and *iii) simplify multi-hop protocol design*, by abstracting the communication medium into a virtual shared bus [30], where messages are broadcasted to the entire network in bounded time and without the overhead and complexity of acquiring and maintaining stateful topological information. A witness to the efficacy and unprecedented performance offered by concurrent transmissions is that most of the teams, and *all the top ones*, in the four editions of the EWSN Dependability Competition [41]—an international event that year after year has attracted researchers from both academia and industry coming from all over the world—rely on CTX-based solutions [3, 62, 63, 64, 65, 66, 67, 68].

Since 2011, when the Glossy protocol [29] pioneered the use of concurrent transmissions in low-power wireless, CTX have rapidly catalysed the attention of the low-power wireless networking community. A considerable research effort has been profused by researchers, mainly funnelled along two, no disjoint paths:

1. investigate the underlying physical phenomena enabling CTX on off-the-shelf low-power wireless devices [69, 70, 71, 72, 73, 74], and

2. leverage CTX to design novel communication protocols and network services, pushing the envelope of what can be achieved by low-power wireless communication at large [10, 28].

After about one decade of research, CTX are nowadays a de facto state-of-the-art asset for designing low-power wireless protocol stacks supporting different traffic patterns and communication demands. Furthermore, although initially relegated to IEEE 802.15.4 narrowband where the concept was first applied, recent works (including the contributions presented in Part II) have paved the way for the exploitation of the same principle with different radio technologies—e.g., Bluetooth Low Energy (BLE), ultra-wideband (UWB), and sub-GHz long-range (LoRa)—with very heterogeneous bandwidth, data rate, range of communication, and energy consumption. This, further expands the horizon of research directions and application use cases that can benefit from concurrent transmissions, and makes CTX-based solutions potentially applicable to millions (if not billions) of off-the-shelves low-power wireless devices already deployed around the world.

The contributions presented in this thesis fall along the second research path mentioned above. However, prior to delving into them, in the remaining part of this chapter we provide the readers with basic knowledge about the PHY-layer principles enabling CTX (§2.1), and offer a primer on the Glossy flooding primitive (§2.2)—both necessary background for the contributions presented in the next chapters.

2.1 The PHY-layer Enablers of Concurrent Transmissions

Even though concurrent transmissions and their enabling factors are still under study and debate within the community, the current understanding suggests that the practical feasibility of CTX mostly depends on two PHY-level phenomena, *non-destructive interference* and *capture effect*, presented next. Both factors are tightly correlated with the receiver's implementation and the power, timing, modulation, and encoding of the incoming signals.

Non-destructive interference. This phenomenon enables the correct reception of a message when multiple copies of the *same* packet arrive at the receiver with a similar power and a very small time difference. Whilst initial works ascribed this successful reception to "constructive interference" [29]—namely, signals perfectly overlapping in time and phase leading to a cumulative and persistent energy gain at the receiver—recent studies demonstrate that pure constructive interference is extremely difficult (if ever possible) to happen in practice [70, 72, 74], and alternative terms such as "non-destructive interference" [72, 75] or "cooperative gain" [76] are nowadays typically preferred. Specifically, *i)* the absence of a shared clock among local radio oscillators, *ii)* non-idealities in the manufacturing process causing subtle carrier frequency variations, and *iii)* path differences among transmitters, lead independent radio transceivers to unavoidably generate non-coherent waves with a time varying phase offset. This holds even when nodes transmit the same message with a very tight time synchronization. As a result, the dominating phenomena in CTX is the beating effect. Due to the carrier frequency offsets (CFOs) between the oscillators of concurrent senders the waveform resulting from the overlapping of identical CTX signals has a beating envelope where peaks of constructive interference are interleaved with valleys caused by destructive collisions. Constructive interference peaks translate into energy gain—hence the name *cooperative gain*—potentially enhancing the *PRR* of low- and medium-quality links w.r.t. single sender transmissions; however, destructive interference periods are prone to cause bit errors that, if not mitigated, can lead to erroneous packet receptions.

In practice, together with a high degree of synchronization among transmitters, three factors appear to be key in favouring a correct reception in the presence of multiple concurrent transmissions of the same packet, and drive the success of non-destructive interference in IEEE 802.15.4: *i)* the receiver implementation; *ii)* the type of modulation; *iii)* the adoption of channel codes and error-correction mechanisms [74]. Specifically, researchers agree that non-coherent detectors and frequency (de)modulation techniques, both widely adopted by off-the-shelf IEEE 802.15.4 narrowband (and BLE) compliant radios[1], are the most natural, high-performance, and cost-effective choice for demodulating beating weaves [74]. Moreover, the direct-sequence spread spectrum (DSSS) encoding, employed by IEEE 802.15.4 2.4 GHz radios to limit the impact of narrowband interference, introduces a not negligible physical level redundancy, key in fighting demodulation errors. By *i)* mapping every 4-bit symbol into a 32-bit (chip) pseudo-random noise sequence among 16 available, and *ii)* leveraging a Maximum Likelihood Estimation (MLE) criterion to map the received chip sequence back into symbols, DSSS lets the receiver tolerate up to *12 bit* (chip) errors per symbol [10], enhancing the probability of correct reception despite CFOs.

Experimental studies [29, 69] show that, thanks to the combination of the three above mentioned factors, when identical packets from multiple senders arrive at an IEEE-802.15.4 narrowband receiver with a time displacement $<$0.5 μs (the chip duration), non-destructive interference occurs with high probability, and a message can be successfully decoded even in absence of capture effect, described next.

Capture effect. As the name suggests, when this phenomenon occurs a receiver *"captures"* and correctly demodulates one signal despite the interference of other incoming messages, whether they carry *the same or different payloads*. Sine qua non for the capture effect to hold is that two constraints—one on the power and one on the timing of the concurrently transmitted packets—are satisfied at the receiver. First, the received signal strength of one of the incoming signals should exceed by a *capture threshold* the sum of all other signals plus noise. Second, such stronger signals should either arrive first, or reach the receiver within a *capture window* from the reception of the first (weaker) signal. The width of the capture window is tightly related to the duration of the preamble; common off-the-shelves radios are indeed capable of switching from one signal to a higher-energy one only prior to receive a complete preamble. After that, and until completing the packet reception, the receiving radio remains locked on the first valid preamble detected. If in the meanwhile a stronger signal arrives, a disruptive collision happens, preventing a successful reception. Instead, when both power and timing conditions are satisfied, the stronger signal is correctly demodulated with high probability.

Experimental results show that the capture effect occurs and is practically exploitable in several popular low-power wireless radios, including BLE [72, 77], LoRa [78, 79], IEEE 802.11 [80], UWB [81], and IEEE 802.15.4 narrowband [31, 32, 82, 83]. The capture threshold and window vary in function of the peculiarities of the physical layers; for instance, IEEE 802.15.4 2.4 GHz radios experience a capture threshold of $\sim$3 dB [69] and a capture window of 160 μs [32].

The large body of CTX literature relying on the capture effect—e.g., all protocols and systems enabling nodes to TX *different* packets concurrently [1, 31, 32, 65, 84, 85]—provides a practical and unquestionable evidence of how this phenomenon can be

[1] Offset-quadrature phase shift keying (O-QPSK) with half-sine chip shaping is equivalent to a minimum shift keying (MSK) modulation, an can therefore be demodulated via a frequency modulation.

efficiently exploited with low-power wireless radios, and of its key role in enabling CTX-based communications.

2.2 Network-wide Flooding with Glossy

The development of communication protocols and network services based on concurrent transmissions has been a real breakthrough in low-power wireless networking, making CTX one of the most influential techniques proposed by the low-power wireless community in the last decade. Since 2013, on average more than 8 peer-reviewed papers, either concerning protocols or systems based on CTX, have been published per-year [10].

Hereafter, we focus on the Glossy system [29], as *i)* similarly to the overwhelming majority of works leveraging CTX, all the contributions presented in this thesis have been influenced by it, and most of them directly rely on its flooding primitive; and *ii)* Glossy provides a clear, practical example of the potentialities CTX unlock once employed for low-power wireless networking. Readers interested in a more comprehensive review of the CTX literature are referred to recent surveys and tutorials [10, 28].

Glossy in a nutshell. Glossy was one of the first systems to demonstrate the effectiveness of CTX for multi-hop low-power wireless networking, by systematically exploiting non-destructive interference and the capture effect to achieve at once extremely fast and reliable network flooding and time synchronization. The public availability of the Glossy source code, along with its outstanding performance, fueled an unprecedented interest in CTX research, until then only marginally explored (e.g., to enable 1-hop acknowledgment for anycast communications [86, 87]). Figure 2.1 illustrates the flooding procedure implemented by Glossy. A node, called

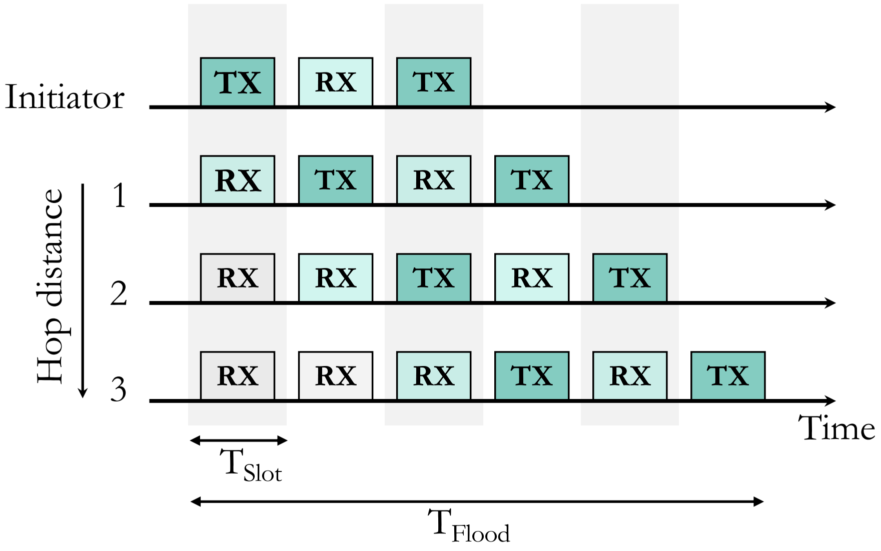


Figure 2.1: Glossy operation in a 3-hop network with $N = 2$ (re)transmissions per node.

initiator, begins a flood by broadcasting a packet. As the rest of the network is assumed to be already listening on the channel, the packet is received by the initiator's neighbours that concurrently retransmit it in the next slot. Glossy ensures that such concurrent transmissions happen within 0.5 μs, enabling receivers to enjoy both non-destructive interference and the capture effect (§2.1). This favours a correct reception at the two-hop neighbours of the initiator, which in turn retransmit concurrently in the next slot. The process repeats following the same set of actions: nodes receiving a packet promptly retransmit it in the next time slot, then go back to receiving until the next successful reception. The predictability of Glossy operations enables nodes to accurately time-synchronize with the initiator—e.g., achieving sub-μs synchronization accuracy across a 8-hop network of TelosB devices [56], as demonstrated in the original paper [29]. For redundancy, each node retransmits the packet up to N times, after which the flood completes. The value of N is key to determine the balance among reliability and energy consumption: large redundancy values enhance the flood dependability, but force nodes to remain awake for longer, consuming more energy and increasing the flood duration.

Fast, reliable, energy efficient network-wide flooding. This approach has been tested extensively by the low-power wireless community [10, 28, 29] and has repeatedly proven to provide unprecedented reliability and latency performance, along with minimal energy consumption.

Reliability benefits from the spatio-temporal diversity inherent in the Glossy flooding procedure. By forwarding packets multiple times along all possible routes, Glossy sharply reduces the impact of time- and space-localized interference, and de-correlates reception errors [88]. Frequency and receiver diversity, obtained via slot-by-slot channel hopping [63, 64] and by enabling multiple receivers to overhear the same packet [2], can further enhance Glossy dependability. Moreover, by forwarding data without relying on topology and link quality information, CTX-based protocols in general, and Glossy in particular, support high degrees of network variability, e.g., node crashes, link failures, and node mobility.

In terms of latency, Glossy disseminates a message network-wide with a timing very close to the theoretical minimum when using half-duplex radios—few milliseconds for few hops, orders of magnitude lower than what typically achieved by routing based solutions. The reason behind this very fast dissemination can be found in the Glossy (re)transmission scheme depicted in Figure 2.1. By allowing nodes to immediately retransmit upon reception, Glossy floods proceed by one-hop at every time slot, making the dissemination latency *independent* of the number of nodes in the network, but *only* a function of the network radius.

Together with latency, this benefits tremendously energy consumption. Nodes need to activate the radio only briefly to disseminate a message to the whole network, explaining how in a somewhat counterintuitive manner energy savings are enabled by a flooding-based technique. CTX protocols have minimal idle listening and control overhead, therefore sparing the significant traffic generated by mainstream stateful routing approaches, especially centralized ones, e.g., for coordination at the MAC layer and topology maintenance and link estimation at the routing layer.

In summary, Glossy demonstrates how CTX can be exploited in full-fledged systems to achieve long-awaited reliability, timeliness, and energy consumption performance. This makes such technique extremely appealing for a variety of applications, spanning from general-purpose IoT scenarios (e.g., collection of sensor readings with best-effort requirements), to increasingly complex and relevant ones, like industrial wireless control, the focus of the next part of this thesis.

Part I

Low-power Wireless Networking Meets Industrial Control Systems

Pushed by the proliferation of fast, dependable, and energy efficient protocols and by the continuous advancements in micro-electro-mechanical systems, low-power wireless networking is becoming appealing also in traditionally wired domains like industrial control [22, 89, 90], where such technology offers several tangible advantages. The replacement of bulky, expensive and difficult to maintain wired solutions with low-cost and energy efficient wireless networks abates installation and maintenance costs, while enhancing flexibility, scalability, and modularity of industrial sites. Wireless systems are easier to (re)place and (re)position, drastically simplifying the design and the deployment phase while fostering fast, on-demand, plant reconfigurability, a prerequisite of modern factories given the frenetic evolution of the production requirements. This translates into a substantial reduction of capital and operational expenditure of the network, without loss in quality of service. At the same time, multi-hop wireless communications are exploitable in circumstances where wirings might be impractical—e.g., on moving and rotating devices—or in places difficult and/or dangerous to access and instrument like steel mines, oil refineries, and chemical plants. By enabling a higher automation of such processes, wireless solutions effectively reduce the need of human activity and intervention, enhancing the safety of the workers. Last but not least, the extensive adoption of wireless technologies offers unprecedented sensing and actuation capabilities, improving the coverage of the factory floor and providing large volumes of real-time, fine-grained data that can be leveraged by powerful analytics to further enhance the efficiency of the production systems. Feed and trained with sensor data, artificial intelligence (AI) solutions can provide at run-time valuable insights about processes and products, which in turn are exploitable to instruct actuation units and workers. This synergy is expected to improve the quality of the products, promote flexible and ever more sophisticated exploitations and reorganizations of the production line—e.g., minimising wastage and machinery underutilization—and drastically reduce downtime periods via accurate predictive maintenance and timely reactions to malfunctions.

At the heart of this industrial revolution [21, 22, 23] (often described in the context of Industry 4.0), which promises to transfigure the current state of production and is already driving the IoT related incomes, lies the capability *to enable fast, reliable and energy efficient wireless control.* It is therefore not surprising that wireless control systems (WCS), the focus of the contributions presented in this first part of the thesis, have recently become one of the key topics of low-power wireless communications in the industrial domain, gaining momentum in academia and becoming a business priority for many organizations [18, 19, 20, 91, 92, 93, 94].

Nonetheless, although low-power wireless sensor networks (WSNs) are widely adopted for *monitoring* tasks, their use for control and automation of plans and process is still in the early stages. Various factors hinder the widespread adoption of low-power wireless technology for industrial control [19, 22, 24, 25, 37, 90, 95]. *First*, WCSs push the network requirements in terms of reliability and latency far beyond the targets set by "plain" IoT applications, which are typically best-effort. Sensing and control data should be delivered with very high dependability across multi-hop networks; "five nines" reliability (99.999%) is a common demand in industrial processes. Similarly, timeliness is a must as many industrial tasks have real-time constraints; late reactions to system changes might be detrimental for the production chains or even dangerous for the safety of the workers. Communications are therefore typically subjected to hard end-to-end deadlines; packets that are correctly delivered but reach the destination after the deadline are considered as lost by the application. *Second*, to fully unleash the scalability and flexibility advantages provided by wireless technology, power cords should be avoided, making energy efficiency of utmost importance given the cost and complexity of replacing batteries. This entails the capability to minimize radio activity, both listening and transmitting, which is well-known to be the main source of power leakage for low-power wireless devices, without affecting the dependability and timeliness of multi-hop communications, and in turn the performance of the control system. Towards this end, the co-design of new WCS architectures that, by leveraging a tighter coupling between the control and network layer, dynamically optimise the number of communications while retaining closed-loop performance is a must [24]. *Third*, industrial plants are often prone to strong radio frequency noise, e.g., due to refractions caused by machineries and moving metallic objects, but also radiations emitted by coexisting devices, like omnipresent Wi-Fi equipments, microwave ovens, and electrical engines, to name a few. Radio interference is a key hampering factor for low-power wireless protocols [26, 27]: by drastically reducing—if not completely breaking—link quality, it hinders node-to-node communications, compromising both data delivery and energy efficiency. Withstanding external interference is therefore vital to meet the stringent dependability, timeliness, and energy consumption requirements demanded by WCSs. *Fourth*, industrial control applications require a multitude of traffic patterns to co-exist and be efficiently handled by the protocol stack. For instance, beyond the periodic data collection that thus far has catalysed most of the research interest on low-power wireless networking, the presence of control loops entails support for point-to-point and point-to-multipoint communications from the controller to one or more actuators. The same traffic patterns are also needed to enable centralized network management, e.g., to adapt the sampling period of the sensors to the status of the process that is being investigated. Furthermore, advanced applications like distributed control and coordination demand many-to-many communications, as the control intelligence does not reside in a single, centralized entity, rather it is (at least partially) spread within the network. Last but not least, aperiodic transmissions are key to guarantee the responsiveness of the system to unpredictable events, e.g., to disseminate alarms when anomalies are detected, or to minimise communications (and in turn energy consumption) when unattended periodic transmissions are not needed.

Addressing these conflicting requirements at once is an extremely challenging task that, notwithstanding the substantial research effort profused by the low-power wireless community, has not yet been fully accomplished.

In the following chapters we advance the state-of-the-art in industrial wireless control by presenting two important contributions. Chapter 3 tackles radio-frequency noise by analysing its impact on CTX-based solutions and by proposing an interference-resilient ultra-low power wireless communication protocol for aperiodic data collection. In Chapter 4, instead, we demonstrate in practice the benefits of event-triggered control (ETC)—a promising aperiodic control paradigm—by co-designing and extensively evaluating *Wireless Control Bus*, the first low-power wireless protocol stack tailored to ETC control and communication demands.

3
Interference-Resilient Ultra-Low Power Aperiodic Data Collection

Ensuring fast, dependable, energy-efficient, and noise-resistant collection of aperiodic events is rapidly becoming a precondition for many industrial wireless control systems, e.g., to minimise downtime periods by enabling accurate predictive maintenance and timely reaction to malfunctions, and to provide efficient networking support for ever more popular and settled aperiodic control strategies [40, 46]. Nevertheless, we observe that aperiodic data collection received little attention in wireless sensor networks, especially if compared to its periodic counterpart, and interference resilience is a largely unexplored topic in the low-power wireless literature. A notable exception is the Crystal system, recently proposed in [31]. Originally designed to exploit the synergy with data prediction, Crystal uses concurrent transmissions [29] to support *aperiodic* and *sparse* traffic with near-perfect reliability, low latency, and ultra-low power consumption.

Is Crystal resilient to strong interference? However, the remarkable performance of Crystal was reported only under mild interference; the authors ran experiments on channel 20 and 26 which *"showed very similar performance [...] during the night runs; however, the daytime results were inconsistent and difficult to assess"* and therefore *"the results only from night runs on channel 26"* were included [31].

Statements like these are not uncommon in the related literature, as discussed later.

This chapter revises the publication [1]: T. Istomin, M. Trobinger, A. L. Murphy, and G. P. Picco. "Interference-resilient Ultra-low Power Aperiodic Data Collection". In *Proc. of the* 17th *ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN. 2018. doi: 10.1109/IPSN.2018.00015.

Nonetheless, this is of particular concern here because interference *i)* potentially undermines Crystal at the core by hampering the capture effect it heavily relies on, and *ii)* increases the overhead of achieving near-perfect reliability of aperiodic and sparse traffic in which "every packet counts", possibly precluding ultra-low power consumption.

Hence, whether the remarkable performance in [31] holds under strong interference—a prerequisite for the adoption of Crystal, or other protocols similarly relying on CTX, in noise-prone environments like industrial settings—is an open question, answered in this chapter by analyzing the performance of Crystal under several, increasingly disruptive noise patterns and introducing techniques to boost its resilience to strong interference without sacrificing ultra-low power consumption.

Natural vs. generated interference. We report experiments in a 49-node indoor testbed and exploit its *natural* interference, mostly WiFi, in line with the evaluation of well-known concurrent transmissions systems [29, 30, 32, 96].

Actually, reliance on natural interference is the *only* methodology hitherto adopted for evaluating them. Despite Glossy and derivatives being commonly considered highly resilient to interference, the extent to which this holds has never been ascertained under noise patterns that are *i) repeatable*, and *ii)* more extensive and disruptive than natural ones.

We raise the standard of evaluating concurrent transmissions under interference by reporting, for the first time, results based on the *reproducible generation* of realistic noise patterns. We use JamLab [35], described in §3.1, to emulate WiFi devices and microwave ovens in our experimental setup (§3.2).

Performance metrics and comparison baselines. We evaluate Crystal using packet delivery rate (*PDR*) and duty cycle (*DC*) as metrics for reliability and energy consumption, respectively. Moreover, as Crystal relies on unmodified Glossy, we indirectly evaluate it with the same experiments under interference; as mentioned above, we argue this is a contribution per se.

We observe that none of the proposals tackling interference found its way into the mainstream. Hence, we choose the readily-available RPL [61] and ORPL [97] as baselines (§3.3), in line with analogous works [98, 99, 100].

Results and contributions. We show (§3.4) that all protocols sustain natural interference, but only Glossy and Crystal achieve near-perfect *PDR*, with a much lower *DC*. Under JamLab-emulated WiFi, RPL reliability degrades even with a single jammer; with several covering the entire testbed, ORPL also degrades, while Crystal still achieves near-perfect *PDR*. Interestingly, roles are reversed when an emulated microwave oven is placed 1m from the sink; ORPL achieves near-perfect *PDR*, while Crystal falls below 80%.

These results pushed us to explore two techniques to improve the resilience of Crystal (§3.5). The first allows nodes to *escape* interference by executing each transmission-acknowledgement pair—a core Crystal constituent—on different channels, based on a network-wide hopping sequence. This approach, which uses Glossy unmodified,

Table 3.1: An aperiodic, sparse traffic profile; number and fraction of epochs with U concurrent senders.

	Epochs	
U	#	%
0	84.3K	82.1
1	15.5K	15.1
2	2.2K	2.2
5	606	0.14
10	46	0.038
20	1	0.005

is notably different from protocols in the literature that apply channel hoping *inside* Glossy [63, 64, 101]. Second, noise detection at all nodes enables them to schedule extra transmissions in a decentralized way, increasing packet delivery. This *fights* interference, effectively providing a "safety net" when channel hopping alone is insufficient, but may keep nodes unnecessarily active, which is detrimental in the sparse traffic targeted by Crystal.

Our experimental results (§3.6) show that the *combination* of these two techniques, to the best of our knowledge novel in the context of concurrent transmissions, achieves near-perfect reliability in the very challenging scenarios where *both* microwave ovens *and* WiFi are simultaneously present. Overall, we confirm that the original Crystal (and the underlying Glossy) can tolerate the moderate levels of interference commonly found in office environments. However, Crystal can also be modified with relative ease to sustain much stronger interference patterns while retaining its ultra-low power consumption, effectively becoming a suitable candidate for low-power wireless control applications in industrial environments.

Finally, we concisely survey related work (§3.7), before ending the paper with brief concluding remarks (§3.8).

3.1 Background

We offer the necessary background on Crystal, along with the JamLab infrastructure used to generate reproducible interference patterns. As Crystal heavily relies on concurrent transmissions in general, and on Glossy in particular, we refer interested readers to Chapter 2 for a primer on this technique and protocol.

3.1.1 Crystal: Efficient & Reliable Aperiodic Data Collection

Crystal [31] builds a schedule atop Glossy that, unlike works geared towards periodic data collection [30, 102], is designed to efficiently support aperiodic, sparse traffic like the one stemming from applying data prediction [103, 104] to regular, periodic traffic. Prediction quenches the majority of application messages, inducing sporadic traffic interleaved with long, quiescent intervals. However, a sudden change in the monitored phenomena may invalidate the prediction model, which must be re-generated and sent to the sink, possibly by multiple nodes at once. Table 3.1, adapted from [31], shows an example traffic profile resulting from applying data prediction to the well-known 36-day Intel dataset [105] containing temperature samples gathered with a period of 30s, hereafter called *epoch*. After data prediction is applied, the majority (82.1%) of the total 102686 epochs is empty, as the sink can predict the next value based on the last model reported by each node. However, in a non-negligible fraction of epochs, $U > 1$ concurrent senders must send model updates. Further, as packets carry models rather than raw data, the loss of a single one has a much larger impact on the reliability of the overall system.

Crystal in a nutshell. To reconcile these requirements, CRYSTAL builds a network-wide transport protocol, in which *i)* a transmission (T) slot is used by U concurrent senders to disseminate their packet; these floods "compete" until, thanks to the capture effect (§2.1) and Glossy redundancy (§2.2), one reaches the sink with high probability *ii)* a subsequent acknowledgment (A) slot is used by the sink to flood the identifier of the sender whose packet it received, informing the others whether retransmission is needed because their packet was "overcome" by another or no packet was received at the sink.

Figure 3.1 illustrates the concept in a simplified setting with only 2 nodes and the sink. A synchronization (S) phase is performed at the beginning of each epoch to ensure time synchronization. Communication occurs via the aforementioned TA pairs, which are repeated by fewer and fewer senders until all have successfully transmitted their packet and the entire network goes to sleep for the rest of the epoch. This termination condition is in principle easily identified by the first *silent pair*, i.e., one without transmissions in T and whose A contains a negative acknowledgment. In practice, matters are complicated by packet losses in either T or A, which may cause a node or the sink to become prematurely inactive. Therefore, CRYSTAL detects termination after R *consecutive* silent pairs; larger values improve reliability but with higher energy consumption. Other parameters are described in [31], e.g., the duration G of guards and the number Z of consecutive missed acknowledgements.

Baseline configuration. In essence, CRYSTAL builds a reliability layer atop Glossy, which strikes different tradeoffs w.r.t. energy consumption by exploiting the interplay

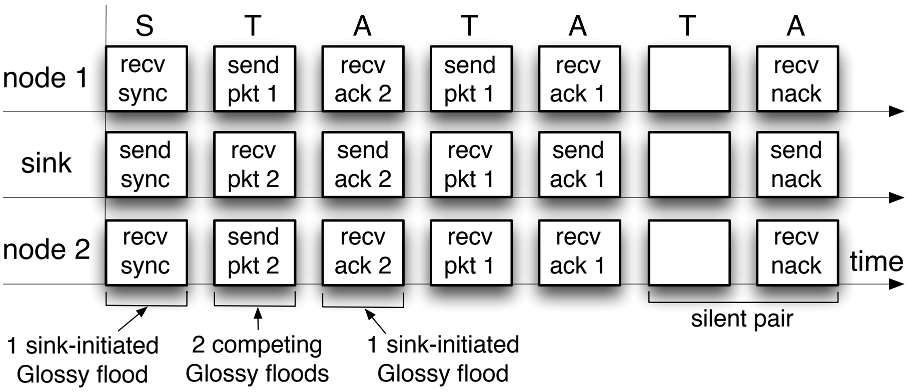


Figure 3.1: CRYSTAL in a nutshell.

Table 3.2: Crystal configurations used in the chapter in function of the nodes' TX power. The values of W_x and G are in milliseconds.

	TX Power			
	Low (-7 dBm)		High (0 dBm)	
$\boldsymbol{N_S}$	3	4	3	3
$\boldsymbol{N_T}$	3	4	2	3
$\boldsymbol{N_A}$	3	4	3	3
$\boldsymbol{W_S}$	12	14	10	10
$\boldsymbol{W_T}$	10	12	6	8
$\boldsymbol{W_A}$	10	12	8	8
$\boldsymbol{G}$	0.15	0.15	0.15	0.15
$\boldsymbol{R}$	2	2	2	2
$\boldsymbol{Z}$	4	4	4	4

between the two layers. As in Glossy, the number N of retransmissions in each flood is key, but in Crystal this can be set independently (N_S, N_T, N_A) for each phase; the same holds for the maximum slot duration W, another key Glossy parameter.

The configuration used in this chapter (Table 3.2) is adapted from the original paper. First, our testbed has a larger diameter than Indriya [106], used in [31]. This forced us to use larger values for the intervals W_T and W_A to allow Glossy floods to complete; we determined the optimal value using the methodology of [31]. Second, we experiment with combinations of N_T and N_A values to analyse the impact of the T and A phase w.r.t. interference, and to explore different tradeoffs between reliability and energy consumption. The values of the remaining parameters W_S, G, R, Z are unchanged.

Finally, we use two power settings, high (0dBm) and low (-7dBm); the former is the default throughout the chapter. Overall, we explore four different configurations (Table 3.2), two for each choice of the TX power.

3.1.2 Generating Interference: JamLab

The ability to reproduce interference patterns is key to our study. Therefore, we rely on JamLab [35], which achieves this goal using the same mote-class nodes available in a testbed, and whose software faithfully emulates various types of interference relevant to IEEE 802.15.4, including Bluetooth, WiFi, and microwave ovens. These have very different characteristics. Bluetooth interferes with all IEEE 802.15.4 channels, as it uses a channel hopping scheme. WiFi typically spans 4 IEEE 802.15.4 channels with interference that is significantly stronger than Bluetooth, but also based on the type of data traffic [26]. Microwave ovens, depending on model and load, may interfere with several consecutive channels, if not all, and induce very strong, continuous interference for 5-10 ms, alternated with inactive periods of 10-15 ms [26, 27, 35]. According to [35], channels 20–26 are affected the most.

Hereafter, to put ourselves in the worst-case scenario, we focus only on WiFi and microwave ovens, as they yield the strongest interference. Similarly, we select the most challenging of the WiFi patterns offered by JamLab (JL_WIFI4) and configure the jammers to transmit modulated carrier at the maximum power (0dBm).

Coping with JamLab "limitations". One criticism of JamLab is that the interference sources it can mimic are limited, and real environments may contain different ones. While this is true, the aforementioned characteristics of WiFi and microwave are different enough to cover a broad spectrum of noise patterns; further, by *combining* them (§3.6), we create an even more challenging interference scenario for our experiments.

Another JamLab limitation is that real interference sources often interfere with many contiguous IEEE 802.15.4 channels at the same time; in contrast, a JamLab node generates noise on a single channel. The majority of the proposed protocols, including staple concurrent transmissions ones (e.g., [29, 30, 31, 32, 96]) and the mainstream solutions in §3.3, operate on a single channel; therefore this constraint does not affect the experiments in §3.4. However, in §3.6 we explore channel hopping and address this JamLab limitation with a channel mapping strategy.

Finally, although we use the maximum TX power (0 dBm) of motes, this is much smaller than real interference sources (e.g., 25 and 60 dBm for WiFi and microwave ovens, respectively). As suggested in [35], we exploit therefore *multiple* motes, strategically placed in our testbed (Figure 3.2), effectively increasing both the level and the coverage of noise.

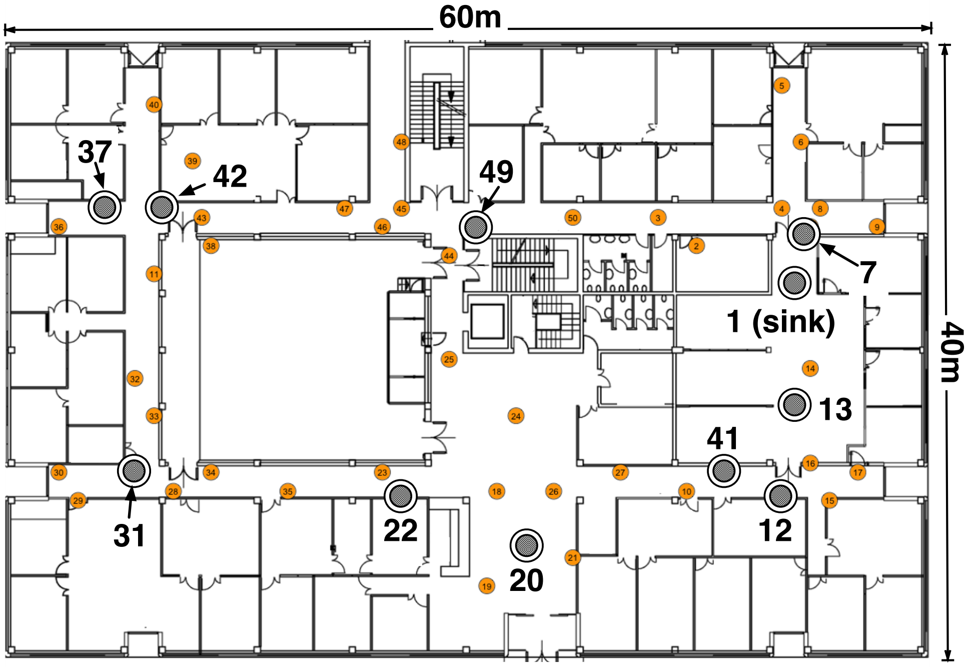


Figure 3.2: Position of the JamLab jammers (grey circles), the sink (node 1), and the other nodes (orange circles) in the testbed.

Table 3.3: Types of interference.

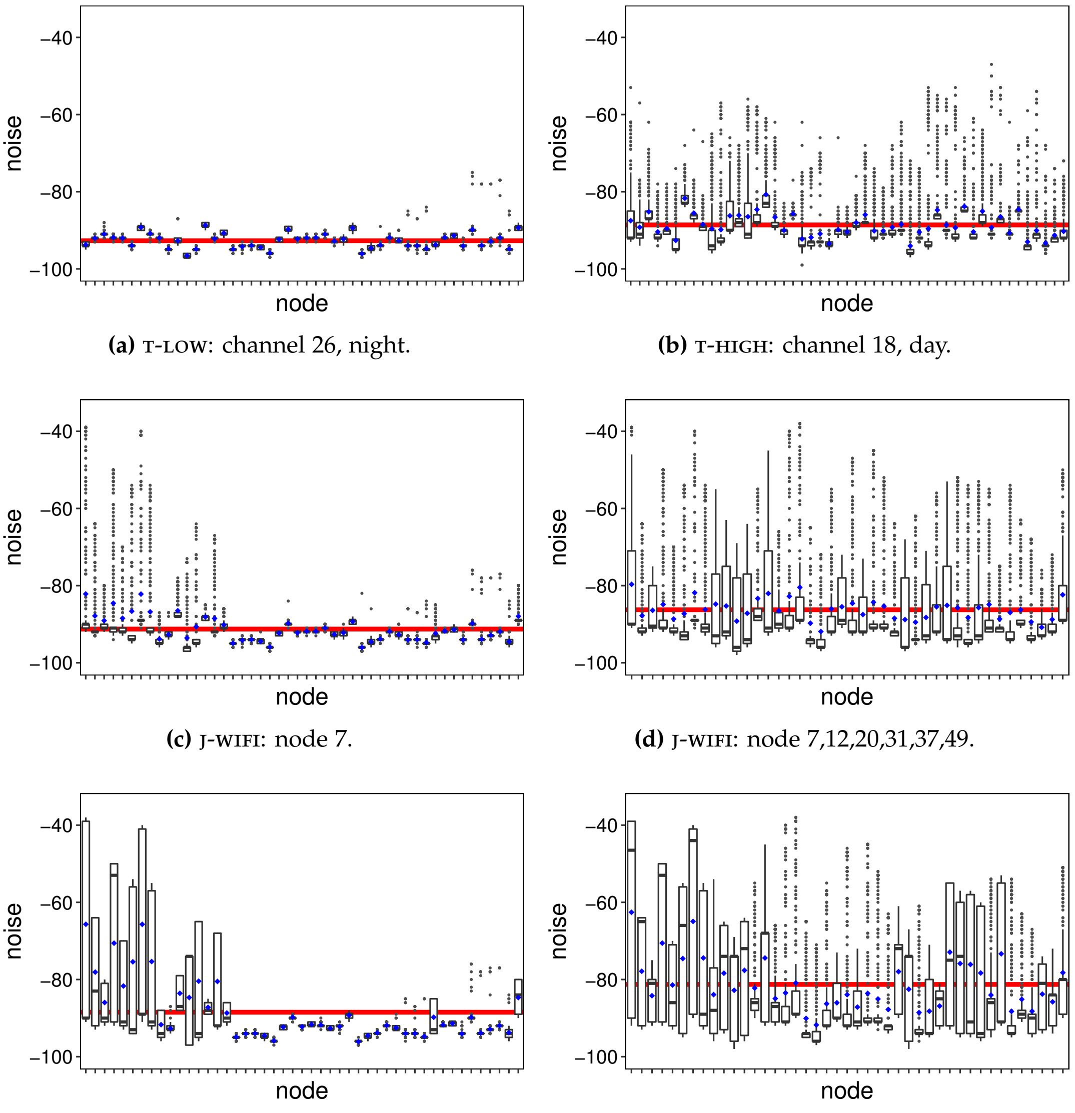
Type of interference	Description
Natural	
T-LOW	Testbed at night/weekends, channel 26
T-HIGH	Testbed during the day, channel 18
Generated	
J-WIFI	JamLab WiFi interference (JL_WIFI4)
J-MWO	JamLab microwave oven interference

3.2 Testbed Interference Scenarios

The experiments we report were performed in a local testbed, composed of 49 TMote Sky nodes deployed as shown in Figure 3.2 in a 60×40 m^2 office area, subject to WiFi interference. Similar to other reports [99], the latter *i)* is more intense during the day and less at night and during the weekends, and *ii)* varies depending on the channel considered. In addition to this *natural* interference, we leverage controlled JamLab *generated* interference, enabling repeatable experiments. Overall, we define four types of interference described in Table 3.3. The choice of channels for natural interference derives from an extensive, cross-channel measurement campaign, which identified the best (26) and worst (18) channels during night and day, respectively. The generated interference is created at night on channel 26 (i.e., under natural T-LOW interference), relying on interfering nodes configured to TX JamLab modulated carrier at the maximum power of 0 dB. Our evaluation uses varying numbers of JamLab jammers for each type, and combines different types of interference in the same experiments, to obtain challenging, realistic setups. Node 1 is the sink in all experiments.

Figure 3.3, obtained via dedicated 30-minute experiments with each node (apart from the designated JamLab jammers, if any) reading 100 times every 15 ms the received signal strength indicator (RSSI) value provided by the radio and no communication performed within the network, quantitatively compares the various types of interference. Figure 3.4, instead, shows the noise effect on the number of links, their qualities and the network radius (Glossy hopcount). The natural T-LOW (Figure 3.3a) exhibits an average noise of -93dBm, rather stable and uniform across the network. The interference in natural T-HIGH is drastically different (Figure 3.3b). The average noise is -88dBm, but several nodes are exposed to much higher noise, reaching -50dBm. This affects the network topology by reducing the number of perfect links by one third, yielding a 10% increase in the average hopcount (Figure 3.4).

The interference generated via JamLab yields stronger noise than the natural one. Figure 3.3c shows the J-WIFI interference generated by node 7 alone, the closest (1m) to the sink. Figure 3.3d shows instead the effect of 6 J-WIFI jammers, including node 7, chosen to cover the entire testbed (Figure 3.2). Compared to Figure 3.3b, the 6 J-WIFI sources subject the network to a noise slightly higher in average (-86dBm) and variance (Figure 3.3d); this affects significantly the network topology, increasing the



(a) T-LOW: channel 26, night.

(b) T-HIGH: channel 18, day.

(c) J-WIFI: node 7.

(d) J-WIFI: node 7,12,20,31,37,49.

(e) J-MWO: node 7.

(f) J-WIFI (node 12,20,31,49) and J-MWO (node 7,42) combined.

Figure 3.3: Noise levels for the scenarios in Table 3.3.

average hopcount by 20% w.r.t. T-HIGH (Figure 3.4).

Figure 3.3e shows the noise generated by a J-MWO jammer on node 7. About one fourth of the network—obviously including the sink—is severely affected, with an average noise from −80 to −65dBm, far higher than the previous scenarios. We experiment with alternate placements of the J-MWO jammer which clearly affects differently the sink, but also has different global effects on the network (Figure 3.4). Moreover, we also experiment with the *combination* of 2 J-MWO and 4 J-WIFI; the resulting noise (Figure 3.3f) is significantly higher than in all previous scenarios, yielding a stronger impact on network topology (Figure 3.4). This scenario combined with a reduced TX power of the network nodes (LP in Figure 3.4) is the most challenging we consider in this paper, translating to a maximum network diameter > 11 hops. Further, when studying specific effects of different jammers and their combinations on protocols, to eliminate the topology bias, we stick to a single 43-node network with the remaining 6 nodes being either active as jammers or switched off. Obviously, the 43-node network is more challenging as it is less connected (Figure 3.4).

3.3 Baseline Mainstream Protocols

We describe the protocols we use as a baseline to compare against Crystal, along with the configuration used in the experiments. All protocols in this paper run atop Contiki [107].

3.3.1 Protocol Descriptions

RPL [61], the Routing Protocol for Low-power Lossy Networks, is an IETF standard. RPL can be seen as an evolution of CTP [108] that, instead of a tree, maintains a directed acyclic graph rooted at the sink. Therefore, each node maintains multiple parents towards the root; a preferred one is used for actual packet forwarding, while

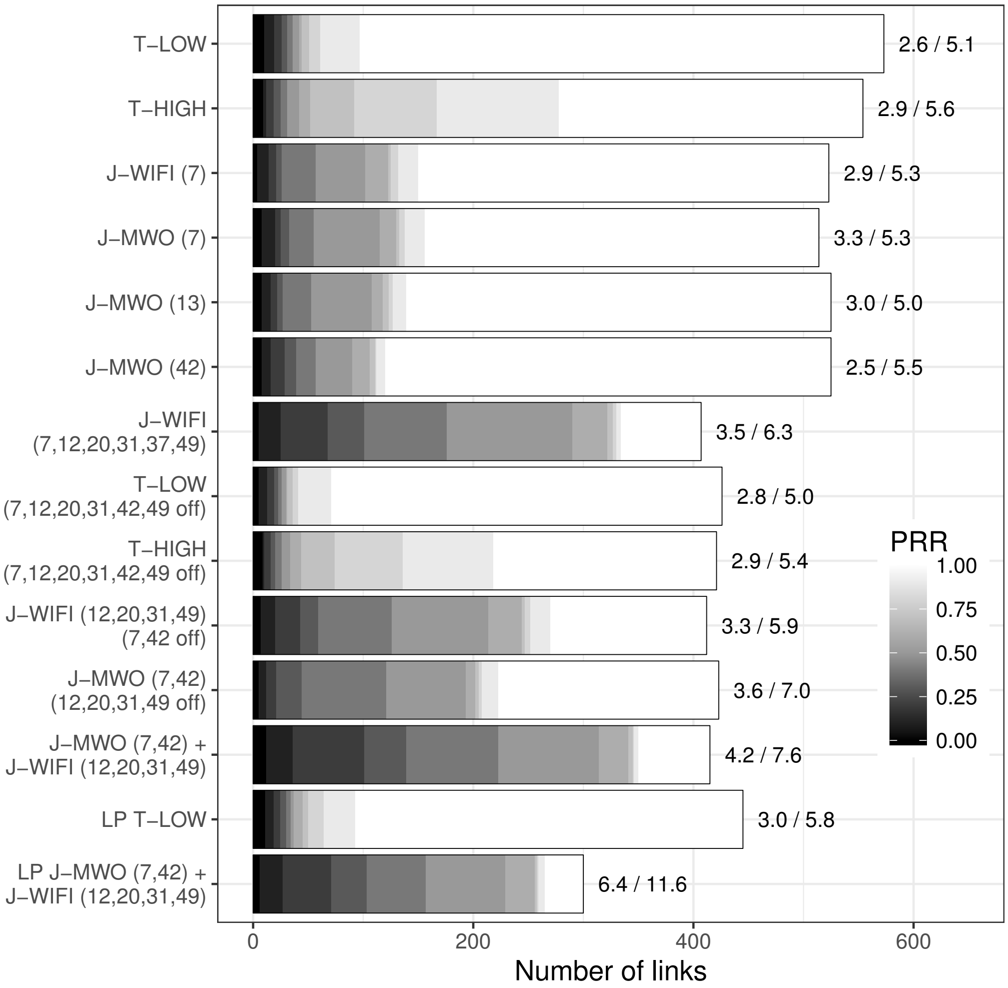


Figure 3.4: Link quality distribution (*PRR*) and network radius (mean/max) in various interference scenarios.

the others are kept as backup routes.

ORPL [97] is an opportunistic routing protocol that inherits many design choices from RPL, but replaces unicast forwarding with anycast. Instead of relaying a packet to the parent, the forwarder broadcasts it; any neighbor closer to the sink is free to catch the packet, acknowledge it, and forward it in the same way. This increases resilience to interference; packets following different paths dynamically avoid noisy areas.

3.3.2 Protocol Configurations

MAC wake-up interval. Both protocols rely on ContikiMAC [109] for medium access control and duty cycling; the value of the wake-up interval is therefore a key parameter affecting performance. We initially chose a value of 8Hz; this is the default, commonly used in the literature. Although our goal in this work is *not* to explore the best configuration of these mainstream protocols, we experimented also with values of 1, 2, 4 Hz, as they may provide better performance under interference. We observed this to be the case for ORPL, which performs best at 2Hz. Therefore, hereafter we report only about wake-up intervals of 2Hz and 8Hz; in general, these also strike a different balance between *PDR* and *DC*, and are therefore interesting to compare. The other configurations perform worse, and are thus omitted.

Choosing the right CCA. The Clear Channel Assessment (CCA) mechanism is used by CSMA link layers to deter a packet transmission if the medium is busy. Its configuration significantly affects the interference resilience of the stack.

The CC2420 radio [110] offers three modes where the CCA reports a busy medium upon detecting *1)* energy above threshold; *2)* valid IEEE 802.15.4 data, regardless of energy threshold; *3)* energy above threshold *or* valid IEEE 802.15.4 data.

We verified that the default -90dBm energy threshold in ContikiMAC yields unacceptable performance; baseline protocols achieve $PDR < 30\%$ even with natural T-HIGH interference. We tested them with several values ranging from -60 to -90 dBm under T-HIGH and generated interference. The value of -77 dBm yielded the best performance and is our choice; in fact, this is the default for CC2420.

As for the CCA mode, the protocols considered use mode 3, the default. With JamLab nodes generating modulated carrier, the question arises whether the noise patterns they emit can be detected by other nodes as legitimate IEEE 802.15.4 data, instead of interference. We performed dedicated experiments comparing results obtained with CCA modes 1 and 3, observing essentially the same performance. Therefore, hereafter we used the default mode 3.

Retransmissions. RPL and ORPL employ different strategies w.r.t. layer 2 retransmissions. When an acknowledgement is not received, a maximum of 7 retransmissions is allowed by RPL, and 4 by ORPL. However, a retransmission can be triggered also by a CCA detecting a busy channel, in which case a few subtleties of the Contiki operating system come into play. Contiki v.3.0, used by RPL, considers 5 busy CCAs as equivalent to a failed transmission. The two events are completely unrelated in Contiki v.2.7, used by ORPL, allowing for an unlimited number of CCAs till the channel is free.

We did not modify these settings, as changing these *default* parameters may have unexpected and undesired effects whose analysis is outside the scope of this paper. We mention them here because they are useful in interpreting the results we present in the next section, e.g., the superior performance of ORPL under strong interference next to the sink.

3.4 Crystal vs. the Mainstream

We compare the protocols in §3.3 against Crystal, and indirectly Glossy, when exposed to the same interference. Aside from the intrinsic value and novelty of this experimental comparison, this serves a stepping stone towards a Crystal design tolerating stronger interference, discussed in §3.5.

3.4.1 Experimental Setup

We analyze Crystal and the baseline protocols in the interference scenarios described in §3.2. We setup a number U of concurrent senders between 0 and 48, offering micro-benchmarks for understanding the individual reliability and energy consumption in function of the amount of traffic; $U = 0$ means absence of traffic while $U = 48$ offers a stress case where *all* nodes but the sink are senders. These parameters match the use cases described in [31] in which a data prediction scheme is applied to periodic data collection applications (e.g., sensing light in a road tunnel or temperature in an office environment). Data prediction reshapes traffic from periodic into sporadic; yet, in a single epoch, U nodes may need to transmit data. The *PDR* of Glossy is derived from Crystal experiments as the *PDR* of the T phase when $U = 1$ (Table 3.7).

In Crystal, all U senders attempt their data packet transmission at *exactly* the same time, i.e., in the first T phase of the epoch, whose duration we set to $E = 2$ s. Due to the very low latency of concurrent transmissions, this value is enough for dissemination to complete in all our tests. Baseline protocols have much higher latency, especially under interference; we set a longer $E = 10$ s for them, denoting solely the period according to which packets are generated. In reporting *DC*, we re-scale the values measured for Crystal to 10 s, to enable direct comparison between the two protocol classes. As discussed in [31], such re-scaling is legitimate as the inner structure of the active part of an epoch is independent from the epoch duration, and the clock skew compensation mechanism implemented in Crystal enables S guard times to be left unchanged for longer, 10 s epochs. Unlike Crystal, the epochs of baseline protocols are *not* synchronized.

Finally, all results are based on several 1-hour runs. For baseline protocols, these are preceded by a 30-minute period since bootstrap during which interference was active, allowing the network topology to stabilize. For Crystal, the total number of packets sent *per configuration* varied from *5k to 500k*, but typically was around 5k–40k. *PDR*

Table 3.4: Natural interference: Baseline, $U = 1$.

		T-LOW		T-HIGH	
Protocol	Wake-up (Hz)	*PDR* (%)	*DC* (%)	*PDR* (%)	*DC* (%)
RPL	2	92.0	1.36	49.3	1.60
RPL	8	93.7	1.2	92.5	1.53
ORPL	2	99.8	0.380	98.2	0.71
ORPL	8	98.7	0.737	98.6	1.45

Table 3.5: Natural interference: ORPL (2Hz) vs. U.

	T-LOW		T-HIGH	
U	*PDR* (%)	*DC* (%)	*PDR* (%)	*DC* (%)
0	—	0.295	—	0.571
1	99.8	0.380	98.2	0.710
5	98.9	0.859	97.4	1.312
10	98.9	1.497	98.4	2.140
20	97.8	2.977	86.3	4.718
48	73.0	6.845	65.5	7.402

is computed over the total number of packets sent. Instead, *DC* is the averaged over values from each 1-hour run, whose variation is anyway negligible.

3.4.2 Natural Interference: T-LOW

We first consider the T-LOW scenario (§3.2), which is akin to several evaluations in the literature, including [31], and offers a good baseline to compare higher interference against.

The performance of mainstream protocols in T-LOW (left-hand side of Table 3.4) is in line with experiments in the literature [97, 111]. As expected, the MAC wake-up interval bears a significant effect: RPL performs best at 8Hz, while ORPL achieves near-perfect *PDR* at 2Hz. Further, its *DC* is much lower than RPL thanks to opportunistic behavior.

These results were derived with a single sender, $U = 1$. Table 3.5 shows results for other values of U; we consider only ORPL as the performance of RPL is significantly lower. The *PDR* of ORPL decreases when traffic increases; ORPL still achieves a good $PDR = 97.8\%$ with $U = 20$, but drops to $PDR = 73\%$ when all nodes transmit in each epoch. Similarly, *DC* increases sharply with U.

These trends are expected; however, Table 3.6 shows that, in the same conditions, CRYSTAL performs significantly better, in line with [31]. Regardless of the $\langle N_T, W_T \rangle$ combination used, CRYSTAL *always* achieves perfect *PDR*, even in the extreme $U = 48$ when *all* nodes transmit concurrently. In these experiments, *not a single packet was lost of total 600k sent.* This is largely to be ascribed to the excellent performance of

Table 3.6: Natural interference: CRYSTAL.

			T-LOW		T-HIGH		
N_T	W_T	U	*PDR* (%)	*DC* (%)	*PDR* (%)	Lost 1 pkt in	*DC* (%)
2	6	0	—	0.293	—	—	0.297
2	6	1	100	0.387	100	∞	0.396
2	6	2	100	0.479	100	∞	0.491
2	6	5	100	0.751	100	∞	0.773
2	6	10	100	1.205	99.9988	83659	1.233
2	6	20	100	2.107	99.9993	134077	2.162
2	6	48	100	4.883	100	∞	4.982
3	8	0	—	0.332	—	—	0.334
3	8	1	100	0.442	100	∞	0.451
3	8	2	100	0.551	100	∞	0.564
3	8	5	100	0.868	99.9984	61482	0.890
3	8	10	100	1.391	100	∞	1.421
3	8	20	100	2.448	99.9995	209201	2.475
3	8	48	100	5.596	100	∞	5.719

the underlying Glossy layer (Table 3.7). Further, CRYSTAL achieves a *DC* lower than ORPL, itself the best among the mainstream protocols considered. For instance, for $U = 48$ the improvement is 18% with $N_T = 3$, and 29% with $N_T = 2$. With no data sent ($U = 0$), the *DC* of ORPL is, however, comparable with $N_T = 2$, and even lower than $N_T = 3$.

Note how the CRYSTAL sink is duty cycled, like other nodes; this is an asset in deployments where powering the sink is complicated. In contrast, the results shown throughout the paper for mainstream protocols use an always-on sink; we verified this provides them with highest *PDR* and lowest *DC*. By considering the sink contribution in the CRYSTAL *DC* computation we put our protocol in worst conditions w.r.t. RPL and ORPL. Nonetheless, we maintain this choice does not bias our comparison: the duty cycled CRYSTAL sink typically presents consumption values in line with the ones of the other nodes in the network, therefore not affecting the average *DC* value we report.

3.4.3 Natural Interference: T-HIGH

Next we discuss experiments assessing the same protocols during daytime, which presents higher levels of interference mostly arising from WiFi traffic, as discussed in §3.2.

Concerning the mainstream baseline protocols, a comparison of the left- and right-hand sides of Table 3.4 shows a generalized decrease in *PDR* accompanied by significant increases in *DC*. As in T-LOW, ORPL is the protocol with the best performance. The price to pay, however, is the nearly twofold *DC* increase for both 2 and 8Hz, as a result of longer idle listening and retransmissions induced by interference. Varying the number U of senders (Table 3.5) shows a similar trend of decreasing *PDR* and increasing *DC*.

Instead, CRYSTAL performs quite well (Table 3.6). *PDR* is perfect or near-perfect regardless of the value of U; the occasional (*4 out of total 700k*) packet loss for some values of U is likely due to the unpredictable nature of T-HIGH. Further, *DC* is nearly identical to the T-LOW case. For instance, in the worst-case scenario of $N_T = 3$ and $U = 48$, the increase in T-HIGH w.r.t. T-LOW is a negligible 0.22%. This is partly ascribed to the inherent reliability of the Glossy protocol CRYSTAL builds upon. However, our experiments also show that Glossy *by itself* does not achieve perfect *PDR*. The superior reliability of CRYSTAL is due to its redundancy mechanisms built *atop* Glossy, overcoming daytime noise with little additional overhead. Another way to look at this is to observe that even in the configuration with $N_T = 2$, i.e., less reliability in the Glossy layer, CRYSTAL still achieves essentially the same *PDR* as $N_T = 3$, while of course enjoying better *DC*.

3.4.4 Generated Interference: J-WIFI

We turn our attention to noise patterns we can control via JamLab (§3.2). We first analyze a single J-WIFI jammer next to the sink, then 6 of them fully covering the network. For the mainstream baseline protocols we focus on $U = 1$, only, as this is sufficient to draw interesting observations; for CRYSTAL, instead, given the significantly higher reliability, we inspect the whole range $U \in \{1, 2, 5, 10, 20, 42\}$ concurrent transmitters.

Single jammer next to the sink. We use a single jammer, node 7 in Figure 3.2; its placement is challenging, at only 1 m from the sink. Table 3.8 shows that RPL achieves $PDR = 84\%$, while ORPL yields near-perfect *PDR* with both 2 and 8Hz, and a *DC* comparable to T-HIGH. In the same conditions, CRYSTAL achieves perfect *PDR* and lower *DC* than ORPL (Table 3.9). This remarkable performance is mainly a consequence of the perfect performance of Glossy, reported in Table 3.7.

Six WiFi jammers covering the entire network. We next consider 6 JamLab nodes generating WiFi interference across the entire network like T-HIGH, but with significantly higher noise (§3.2). As RPL showed low performance even with a single jammer, we focus on ORPL, which has significant difficulty overcoming this noise level, regardless of the wake-up interval. In the best case, 2Hz achieves $PDR = 64\%$ (Table 3.8), notably with a much higher *DC* than under natural T-HIGH (3.91% vs. 1.45%).

Glossy instead achieves near-perfect *PDR* (Table 3.7), becoming *perfect* once combined with the CRYSTAL mechanisms built atop, yielding a *DC* only 12% higher than T-LOW (Table 3.9). When $U > 1$ (Table 3.10) CRYSTAL experiences a slight *PDR* decrease, anyway always within 1% even under extreme traffic burst and $N_T = 2$. The reason for the rare losses is that, with very high interference throughout the network, it might seldom happen that no alternate, good paths exist for packets to reliably reach the sink.

Table 3.7: *PDR* of Glossy.

Scenario	N_T, W_T	***PDR*** **(%)**
T-LOW	2, 6	100
	3, 8	100
T-HIGH	2, 6	99.971
	3, 8	99.985
J-WIFI 1 jammer	3, 8	100
J-WIFI 6 jammers	3, 8	99.32
J-MWO 42	3, 8	99.88
J-MWO 13	3, 8	100
J-MWO 7	3, 8	67.90
J-MWO 7	6, 12	83.86
J-MWO 7	10, 17	99.76

Table 3.8: Generated noise: Baseline, $U = 1$.

Interference Type	Interferer ID	Protocol	Wake-up (Hz)	*PDR* (%)	*DC* (%)
J-WIFI:					
	7	RPL	2	89	1.30
			8	84	1.50
	7	ORPL	2	99.9	0.59
			8	99.7	1.31
	7, 12, 20, 31, 37, 49	ORPL	2	64	1.70
			8	60	3.91
J-MWO:					
	42	ORPL	2	98.6	0.844
			8	98.3	2.13
	13	ORPL	2	98.0	0.67
			8	99.7	1.84
	7	ORPL	2	99.8	0.67
			8	99.1	2.23

3.4.5 Generated Interference: J-MWO

We study the impact of a JamLab-emulated microwave oven, causing interference much stronger than WiFi and with different temporal patterns (§3.2). In our experiments, we move the jammer progressively closer to the sink, yielding increasingly challenging scenarios. Given the results in the previous section, our comparison against mainstream protocols considers only ORPL, as RPL yields unacceptable performance.

Jammer *far* from the sink, node 42. We first use a jammer on node 42, far from the

Table 3.9: Generated noise: CRYSTAL, $U = 1$.

Interference Type	Interferer ID	N_T, W_T	R	PDR (%)	DC (%)
J-WIFI:					
	7	2, 6	2	100	0.403
		3, 8	2	100	0.457
	7, 12, 20, 31, 37, 49	2, 6	2	100	0.443
		3, 8	2	100	0.497
J-MWO:					
	42	2, 6	2	99.52	0.430
		3, 8	2	100	0.507
	13	2, 6	2	100	0.405
		3, 8	2	100	0.459
	7	2, 6	2	78.6	0.425
		3, 8	2	78.5	0.453
	7	3, 8	6	100	1.11
	7	6, 12	2	100	0.839

Table 3.10: CRYSTAL under 6 J-WIFI interference, $U > 1$.

N_T	W_T	U	PDR(%)	lost 1 pkt in	DC(%)
2	6	2	100	∞	0.559
2	6	5	99.823	566	0.884
2	6	10	99.872	784	1.411
2	6	20	99.876	806	2.508
2	6	42	99.196	124	5.348
3	8	2	100	∞	0.626
3	8	5	100	∞	0.996
3	8	10	99.969	3249	1.589
3	8	20	99.982	5561	2.852
3	8	42	99.836	609	5.947

sink, in a corner of the network, and amid a dense neighborhood; its noise affects neighboring nodes, but bears limited influence to the rest of the network.

ORPL performs well in this scenario (Table 3.8) although with a *DC* increased w.r.t. lower-noise scenarios. This is due to its buffering and continuous attempts to retransmit packets until it finds the channel free, as we discussed in §3.3.2. Recall that the J-MWO scenario induces periods of strong interference alternated to periods with no interference (§3.1.2). Therefore, the buffering and infinite CCA retries in ORPL effectively delay packets when the microwave oven interference is active, enabling their transmission during no-interference periods. Nevertheless, these retransmissions do increase the *DC*.

Crystal, instead, achieves perfect *PDR* (Table 3.9). Nevertheless, the underlying Glossy layer is affected by interference (Table 3.7); therefore, reliability in Crystal comes at the cost of a higher *DC*. This cost is even higher than with 6 WiFi jammers, although in the latter case the *PDR* of Glossy is worse. The reason is the position of node 42; being in a corner of the network, its strong interference causes the loss of acknowledgments in that neighborhood, triggering retransmissions from the corresponding senders and unnecessarily keeping the *entire* network awake to help forwarding. Instead, in the scenario with 6 WiFi jammers covering the entire network, packet losses are spatially and temporally distributed, and the redundancy brought by both Glossy and Crystal enables packets to more easily find routes "around" the interference.

Jammer *close* to the sink, node 13. We now move the jammer to node 13 at about 4 m from the sink. Intuitively, this is likely to be more disruptive than the far away node 42, but less than an even closer placement, discussed next.

Yet, our results tell a different story. The *PDR* of ORPL is nearly perfect, as shown in Table 3.8, and is achieved with a *DC* that is about 20% lower w.r.t. the previous case with node 42. The same holds for Crystal (Table 3.9), which achieves perfect *PDR* with a *DC* that is about 9% lower w.r.t. node 42, thanks to the perfect reliability of Glossy (Table 3.7). This improved performance arises from jammer position. Node 13 is closer to the sink than 42 and induces stronger interference on it, but it is also more "central", allowing packets to follow routes "around" it. Instead, node 42 is in the network corner, where noise disruption is much harder to compensate via alternative routes.

Jammer *next* to the sink, node 7. When moving the j-mwo jammer on node 7, at 1 m from the sink, the *PDR* of Crystal significantly degrades for the first time, causing a 21.5% packet loss (Table 3.9), mainly due to the fact that, unlike previous scenarios, the underlying Glossy layer loses 32.1% of the packets. The reason is that the interference on node 7 is so strong and so close to the sink that Glossy cannot overcome it. Receiving packets via alternate routes, as with node 13, is no longer an option because *all* routes are jammed by interference, given that the sink is basically at the center of it.

In contrast, ORPL achieves near-perfect *PDR* also in this case (Table 3.8) and with a *DC* only marginally different w.r.t. the interference source on node 13. From the point of view of ORPL, the two situations are virtually the same: *i)* both node 7 and 13 are in the center of the network, unlike the more challenging corner placement of node 42, and *ii)* in both cases, buffering and retransmissions guarantee that a packet not received by the sink due to interference is eventually received in the periods without it.

Instead, Crystal dissemination is designed to be as fast as possible to spare energy, even with the redundancy it builds atop the even shorter one-shot Glossy floods. Consequently, Crystal and Glossy cannot exploit a "wait-and-see" strategy as in ORPL.

3.4.6 Is There a Better Configuration?

We study a configuration yielding perfect *PDR* in the worst scenario for concurrent transmissions, i.e., node 7 as J-MWO jammer. We explore two options: one directly in CRYSTAL, and one in the underlying Glossy.

Crystal: Keeping the network awake. We observed that an asset of ORPL is that it can retransmit until interference ceases. The CRYSTAL analogous comes from increasing *R*, i.e., the number of consecutive silent TA pairs detected before determining that communication is over and it is safe to enter sleep mode until the next epoch (§3.1.1). Increasing *R* keeps the network awake longer, even when the sink reports via its A slot that no packet arrived in the previous T slot. This gives senders more opportunities to attempt retransmission under interference. Indeed, Table 3.9 shows that $R = 6$ enables perfect *PDR*. However, keeping the network awake for three times longer than before causes a nearly three-fold increase in *DC* when the traffic is sparse.

Glossy: Increasing redundancy. An alternative is to make the underlying Glossy layer more reliable. The main knob to achieve this is to increase the number *N* of retransmissions during a flood, and increase the slot duration *W* to ensure the flood has enough time to complete (§3.1.1). We verified that, when pure Glossy is used in isolation, a setting $N = 10$, $W = 17$ yields $PDR = 99.76\%$. However, the reliability provided by CRYSTAL atop Glossy enables the use of a smaller *N*, considerably reducing *DC*. Table 3.9 shows that with $N_T = 6$, $W_T = 12$, CRYSTAL achieves perfect *PDR* (despite Glossy yielding only $PDR = 83.86\%$, see Table 3.7) but nearly doubles *DC*, as each packet is transmitted twice as many times w.r.t. $N_T = 3$.

In summary, a proper *static* configuration of CRYSTAL or Glossy parameters enables perfect reliability but with unacceptable power consumption w.r.t. ORPL (which, however, does *not* achieve perfect *PDR*). Ideally, perfect *PDR* should come without increasing significantly the *DC* observed in the other scenarios in Table 3.9, i.e., at most 0.50%. Further, over-provisioning for the worst case, as these static configurations do, is undesirable. Ideally, CRYSTAL should *dynamically* adapt to interference, bearing extra energy costs only when needed, thus preserving on average its ultra-low power consumption. These observations motivate the techniques we illustrate next.

3.5 Taming Strong Interference

We illustrate a technique to *escape* interference and a complementary one to *fight* it after detecting its presence.

Escaping interference: Channel hopping. Exploiting frequency diversity is a well-known technique for interference resilience [112, 113]. Interference usually affects only some of the 16 channels available in IEEE 802.15.4 (§3.1.2). Therefore, a channel-hopping sequence can be used network-wide to enable subsequent TA pairs to move to different channels, reducing the probability that two consecutive ones both execute on noisy channels. This simple modification does not affect any CRYSTAL parameters.

Channel hopping is driven by the S phase (Figure 3.5); the channels of TA pairs in the epoch depend on the S channel, itself based on a predefined sequence. This mechanism realigns all nodes to the same channel at the epoch start, independent of the number of TA phases they executed in the previous one.

A key decision is which channel to use next. As discussed in §3.1.2, WiFi and microwave ovens are common noise sources, jamming 4 and 7 adjacent channels, respectively. Spacing the current and next channel apart by 4 channels is sufficient to escape WiFi, but not microwave ovens. Therefore, our implementation uses a hopping sequence with 7-channel spacing; alternate hopping sequences can exploit a priori knowledge about interference. Notably, selecting the number of channels to hop over requires little knowledge compared, e.g., to approaches that probe the environment and limit themselves to channels with the least interference [100]. Furthermore, the proposed approach is more resilient to sudden changes in the noise pattern; by constantly sweeping the available channels and skipping over the subsequent ones that are more likely to be concurrently jammed, we reduce the likelihood of remaining on channels with high interference.

Fighting interference: Noise detection. Our next technique relies on the ability to detect abnormally high noise levels. Recall from §3.1.1 that, in Crystal, the distributed termination condition relies on counting *silent pairs* and missed acknowledgements. Under high noise, these *missing-packet* conditions often occur even when a packet was transmitted, but encountered interference. If noise strikes during the T phase close to the sink, the sender will re-transmit the packet in the next T slot. If the sink still does not receive the packet in *R* consecutive T slots, it mistakenly detects termination and puts the whole network to sleep. Instead, noise in the network periphery may cause a node to similarly miss *Z* acknowledgements and go to sleep, likely before the sink. In both cases, data may remain un-delivered because termination was falsely detected.

Adding noise detection and *dynamically* changing termination conditions fights these cases. Noise detection can be easily achieved by periodically checking the CCA pin of CC2420 [110]; in our implementation, all nodes perform the CCA every 64 μs while listening during T or A phases, and define high noise when $RSSI > -60$ dBm is detected at least 80 times, i.e., during the majority of the CCA checks executed within the phase. This threshold is designed to detect only very high noise, e.g., a microwave oven; lower thresholds would unnecessarily trigger the scheduling of

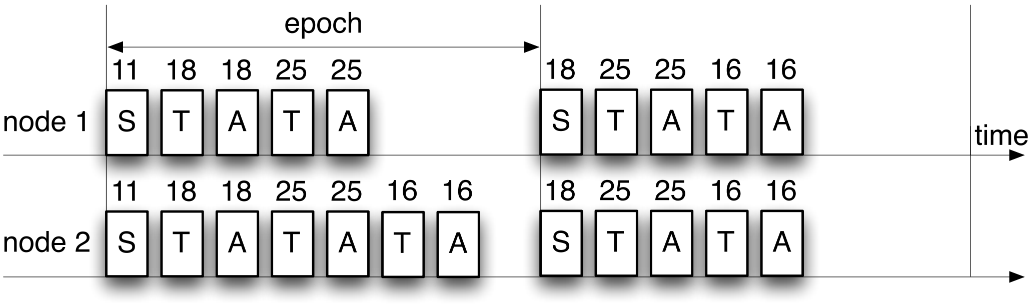


Figure 3.5: Channel hopping in Crystal. The number on each Crystal phase denotes the channel used.

extra TA pairs, e.g., in the WiFi scenarios of §3.4, where even the unmodified Crystal achieves perfect reliability.

As for distributed termination, intuitively, in the presence of noise missing packets do not count towards termination, keeping the network awake and allowing more opportunities for data and acknowledgments to escape the interference. Recall that receiving any packet keeps a node awake to serve as a forwarder. We make the following modifications to Crystal:

1. Define R_{noise} as the maximum number of *consecutive* slots *i)* without a packet *and ii)* with high noise.
2. Change the termination rule at the sink; the network goes to sleep when *either i)* R non-noisy no-data T slots occur since the last received data, *or ii)* $max(R, R_{noise})$ consecutive noisy no-data T slots occur.
3. Change the termination rule elsewhere; a node goes to sleep when *either i)* it receives a sleep command from the sink, *or ii)* it detects Z non-noisy no-data slots since the last packet received in T or A, *or iii)* $max(Z, R_{noise})$ consecutive noisy, silent A slots occur.

In summary, we augment the original termination strategy of Crystal with a mechanism that simultaneously allows to *i)* ignore slots that are silent due to packet loss induced by high noise, overcoming short-term interference, and *ii)* avoid unnecessarily long listening if high noise persists.

Nonetheless, a potential drawback of our technique is that it schedules extra TA pairs upon detecting high noise, regardless of whether a packet is currently being disseminated. If no packet is being disseminated—and there is *no way* to ascertain it in Crystal due to the aperiodic and sparse traffic it targets—our noise detection technique may yield unnecessary energy consumption.

We empirically determined that $R_{noise} = 6$ strikes a good balance between reliability and energy consumption.

Fighting *and* escaping interference. Although both techniques improve performance along some dimension, it is only through their *combination* that very strong interference can be effectively overcome with very low energy consumption, as shown next. Indeed, frequency diversity reduces the probability of the sink to be exposed, in consecutive TA pairs, to high noise levels from the same source, mitigating the above drawback of noise detection. On the other hand, the ability to detect and react to noise is helpful in reducing packet loss when hopping from one bad channel to another one.

3.6 Under Strong Interference

We now evaluate the techniques in §3.5 and show that they not only overcome the interference scenarios considered in §3.4, but also sustain much higher noise levels, detailed next.

Table 3.11: Scenarios with combined interference generated by 2 J-MWO and 4 J-WIFI.

Scenario	# channels jammed		Description
	2 J-MWO	4 J-WIFI	
COMBINED$_{split}$	7	6	Jammers on different *real* channels based on the interference type, mapped on different sets of channels.
COMBINED$_n$	$n \in \{7, 10, 13, 16\}$		All jammers on one *real* channel, itself mapped on n channels.

3.6.1 Experimental Setup

We extend our experimental setup along two dimensions.

Channel mapping. Testing our channel hopping mechanism in principle requires reproducing interference across multiple channels, something JamLab cannot do, as discussed in §3.1.2. We overcome this limitation via a *mapping* between the 16 channels provided by IEEE 802.15.4 and those in the testbed. Whenever our channel hopping mechanism decides to switch to a channel c, a corresponding channel c_{real} is instead used for communication, based on a predefined mapping $c \rightarrow c_{real}$ based on the interference types and channels affected we want to reproduce. For instance, when emulating a microwave oven, we map channels 20–26 to the real one used by J-MWO jammers.

More challenging interference scenarios. As described later, extending CRYSTAL with the techniques in §3.5 allows it to sustain much stronger interference than the one in §3.4, which considered the *separate* effect of generated J-WIFI and J-MWO interference. Therefore, we now focus on the *combined* effect of these two interference types. We combine them in two ways, yielding the scenarios in Table 3.11. The first, COMBINED$_{split}$, combines the two types of interferences by placing each on different real channels. This significantly reduces the chances that channel hopping finds a good channel, and increases the likelihood to hop from one type of interference to the other. The second scenario, COMBINED$_n$, is even more challenging, placing J-MWO and J-WIFI jammers on the *same* real channel, generating noise that is the *sum* of the two. Increasingly challenging scenarios can be generated by determining the number n of channels this strong interference is mapped to. Table 3.11 shows we experiment with n ranging from 7 (i.e., 7 channels where J-MWO and J-WIFI fully overlap) to 16 (i.e., *all* channels jammed by the same combined interference).

Besides combining interference types, we also strengthen J-MWO, the most disruptive one, by using 2 jammers simultaneously, the worst in §3.4: node 7 next to the sink, and node 42 in the corner. As for J-WIFI, using the scenario with 6 jammers would force us to remove 8 nodes in total, further reducing the network size. Therefore, we used 4 J-WIFI jammers that, we verified, yield a noise pattern close to natural T-HIGH.

Table 3.12: ORPL (2Hz) in a 43-node network, $U = 1$.

	TX Power 0 dBm		**TX Power -7 dBm**	
Scenario	*PDR* (%)	*DC* (%)	*PDR* (%)	*DC* (%)
T-LOW	99.6	0.497	97.0	0.454
T-HIGH	98.5	0.776	—	—
4 J-WIFI	61.0	1.35	39.5	6.192
2 J-MWO	97.8	1.19	94.8	1.503
2 J-MWO 4 J-WIFI	65.0	2.14	39.6	5.375

Table 3.13: CRYSTAL$^{\text{CH}}$, under T-HIGH.

N_T	U	*PDR* (%)	*DC* (%)
2	0	—	0.294
2	1	100	0.392
2	2	100	0.486
2	5	100	0.766
2	10	100	1.221
2	20	100	2.122
2	48	100	4.906

As mentioned at the end of §3.2, we use the resulting 43-node network across all scenarios. To re-establish our ORPL baseline, Table 3.12 reports experiments with T-LOW, T-HIGH, 4 J-WIFI, 2 J-MWO, and the combination of 4 J-WIFI and 2 J-MWO jammers over a single channel. ORPL performance is good also with 2 J-MWO, but degrades significantly even with only 4 J-WIFI jammers, instead of the 6 used in §3.4.4.

3.6.2 Channel Hopping

We are now ready to study CRYSTAL extended with channel hopping as discussed in §3.5. We call this variant CRYSTAL$^{\text{CH}}$, to distinguish it from the original single-channel one, and call CRYSTAL$^{\text{CH}}_{\text{ND}}$ the variant that also adds noise detection.

Table 3.13 reports experiments under natural T-HIGH interference, without channel mapping and with CRYSTAL$^{\text{CH}}$ hopping across all 16 channels. A comparison with Table 3.6 shows that CRYSTAL achieves perfect *PDR* (no packets lost of total 150k sent) regardless of U, and does so with $N_T = 2$, which generally yields worse *PDR* w.r.t. $N_T = 3$. Further, *DC* is 1–2% lower than the single-channel version under T-HIGH.

A bigger question lingering from §3.4 is whether CRYSTAL$^{\text{CH}}$ can overcome J-MWO interference next to the sink. We first analyze the performance in the COMBINED$_{split}$ scenario (Table 3.14). Recall this subsumes the scenario with J-MWO on node 7 we discussed at the end of §3.4 by adding a second jammer on node 42, and overall defines a much more challenging setup. Indeed, when hopping out of J-MWO interference, found with a $\frac{7}{16} = 43.75\%$ probability, there is still a 37.5% chance to stumble on J-WIFI interference, and only a 18.75% probability to enjoy T-LOW interference. Nev-

Table 3.14: Crystal$^{\text{CH}}$, under combined$_{split}$.

N_T	U	*PDR* (%)	**Lost 1 pkt in**	*DC* (%)
2	0	—	—	0.335
2	1	100	∞	0.454
2	2	100	∞	0.583
2	5	99.949	1980	0.954
2	10	99.982	5620	1.568
2	20	99.979	4822	2.792
2	42	97.627	42	5.646
3	0	—	—	0.374
3	1	100	∞	0.514
3	2	100	∞	0.661
3	5	99.952	2086	1.069
3	10	99.988	8374	1.750
3	20	99.941	1681	3.138
3	42	100	∞	6.434

ertheless, Crystal$^{\text{CH}}$ achieves perfect *PDR* for $U = 1$ and three-nines reliability for $U > 1$, $N_T = 3$. Further, this is achieved with only a slight increase in *DC* w.r.t. our lowest-interference scenario, t-low: 14.3% and 12.6% for $U = 0$ and $N_T = 2$ and $N_T = 3$, respectively.

The next step is to identify the limit of Crystal$^{\text{CH}}$, which clearly depends on the type of interference applied and number of channels affected. Table 3.15 explores this limit by using the combined$_n$ scenario of Table 3.11. The interference is stronger, as it is the *sum* of 2 j-mwo *and* 4 j-wifi, which in Table 3.14 are instead split on separate sets of channels. Moreover, we apply this strong interference to an increasing number n of channels, progressively reducing the options to hop away from interference. Table 3.15 (left) shows that when $n = 7$, Crystal$^{\text{CH}}$ achieves perfect *PDR* regardless of N_T and number U of senders, with a *DC* marginally smaller than in combined$_{split}$. However, when only 6 channels are free and the others subjected to combined$_{10}$ interference, performance drastically drops. With $U = 1$ sender active, $PDR = 94\%$ is achieved at best, with $N_T = 2$; as U increases, *PDR* plummets. Finally, with only 3 channels free reliability reaches an unacceptable $PDR < 85\%$ with $U = 1$, and at best $PDR = 13.9\%$ with all $U = 42$ senders.

3.6.3 Channel Hopping *and* Noise Detection

These scenarios are very challenging, both in absolute terms and w.r.t. the literature, making the performance of Crystal$^{\text{CH}}$ already remarkable. Nevertheless, we can push reliability even further. When interference affects so many channels that it becomes difficult to *escape* it, the only other choice to improve reliability is to *fight* it with noise detection (§3.5).

Indeed, starting from 10 channels jammed, unlike the channel hopping alone, its

Table 3.15: Crystal$^{\text{CH}}$ vs. Crystal$^{\text{CH}}_{\text{ND}}$, under combined$_n$.

			Crystal$^{\text{CH}}$			Crystal$^{\text{CH}}_{\text{ND}}$		
n	N_T	U	*PDR* (%)	Lost 1 pkt in	*DC* (%)	*PDR* (%)	Lost 1 pkt in	*DC* (%)
7	2	0	—	—	0.328	—	—	0.367
7	2	1	100	∞	0.444	100	∞	0.487
7	2	10	100	∞	1.574	100	∞	1.624
7	2	20	100	∞	2.746	100	∞	2.890
7	2	42	100	∞	5.848	100	∞	5.936
7	3	0	—	—	0.370	—	—	0.444
7	3	1	100	∞	0.501	100	∞	0.576
7	3	10	100	∞	1.749	100	∞	1.826
7	3	20	100	∞	3.142	100	∞	3.256
7	3	42	100	∞	6.494	100	∞	6.566
10	2	0	—	—	0.347	—	—	0.430
10	2	1	94.439	18	0.458	99.919	1237	0.544
10	2	10	71.201	3	1.289	99.962	2637	1.911
10	2	20	46.900	2	1.511	99.788	471	3.414
10	2	42	22.459	1	1.618	99.557	226	7.008
10	3	0	—	—	0.386	—	—	0.521
10	3	1	93.262	15	0.512	99.919	1230	0.660
10	3	10	74.722	4	1.469	99.646	282	2.130
10	3	20	54.497	2	1.878	99.252	134	3.730
10	3	42	30.353	1	2.271	98.402	63	7.464
13	2	0	—	—	0.362	—	—	0.483
13	2	1	84.992	7	0.467	99.748	397	0.621
13	2	10	36.581	2	0.844	99.719	356	2.322
13	2	20	19.497	1	0.817	99.409	169	4.132
13	2	42	9.277	1	0.882	97.577	41	8.456
13	3	0	—	—	0.410	—	—	0.628
13	3	1	85.775	7	0.522	99.919	1237	0.696
13	3	10	47.427	2	1.073	98.608	72	2.484
13	3	20	26.522	1	1.121	96.853	32	4.255
13	3	42	13.936	1	1.230	97.360	38	8.436

combination with the noise detection (Table 3.15, right) achieves two- to three-nines *PDR* with $U = 1$, depending on the choice of N. For $U > 1$ the performance gain is even more visible as Crystal$^{\text{CH}}_{\text{ND}}$ *always* ensures $PDR > 96.8\%$, even with 13 channels jammed and very high traffic loads.

Noise detection becomes more and more important as the number n of jammed channels increases. The extreme case is when *all* channels are jammed by the same

Table 3.16: Crystal$^{\text{CH}}_{\text{ND}}$, under combined$_{16}$.

N_T	U	*PDR* (%)	**Lost 1 pkt in**	*DC* (%)
2	0	—	—	0.543
2	1	99.230	130	0.734
2	2	99.120	114	0.948
2	5	98.777	82	1.622
2	10	98.732	79	2.803
2	20	98.582	71	5.372
2	42	90.056	10	9.308
3	0	—	—	0.739
3	1	99.515	206	0.927
3	2	99.361	157	1.145
3	5	98.914	92	1.693
3	10	98.394	62	2.825
3	20	96.640	30	5.206
3	42	94.397	18	9.952

strong interference (Table 3.16); channel hopping becomes pointless and reliability is provided entirely by noise detection, which performs quite well. Indeed, the *PDR* achieved here is only marginally lower than in combined$_{13}$, with the worst-case $U = 42$ achieving $PDR = 90\%$. To put this value in context, we observe that it is *i) comparable* with what RPL achieves in t-low with $U = 1$ (Table 3.4), and *ii) more* than what ORPL achieves in the natural t-high (no microwave ovens) with $U = 20$ (Table 3.5).

The price to pay for this remarkable reliability is energy consumption. A drawback of noise detection is that high noise keeps the network awake even without packet transmissions (§3.5). This is reflected in the *DC* increase as the number n of jammed channels grows, which increases the likelihood of remaining unnecessarily awake. This is clearly undesirable for $U = 0$; yet, it is key to reliability as U increases, as seen by comparing the two sides of Table 3.15. The actual impact of this increased *DC* on the overall energy consumption depends on the aperiodic traffic at hand, as we analyse in §3.6.5.

On the other hand, we also verified that under t-high, unlike the extremely challenging scenario above, the *DC* of Crystal$^{\text{CH}}_{\text{ND}}$ does not increase w.r.t. Crystal$^{\text{CH}}$ (Table 3.13) since interference is never strong enough to trigger our noise detection mechanism. This is a remarkable and fundamental result; Crystal$^{\text{CH}}_{\text{ND}}$ capability to withstand strong interference *without* entailing unnecessary energy costs under mild noise makes our protocol a suitable solution for a wide spectrum of applications, notably encompassing noise-prone industrial WCSs as well as typically less interference-sensitive general-purpose IoT applications.

Table 3.17: Low power: $\text{Crystal}_{\text{ND}}^{\text{CH}}$.

	COMBINED$_{split}$				COMBINED$_{16}$			
U	N_T	PDR (%)	Lost 1 pkt in	DC (%)	N_T	PDR (%)	Lost 1 pkt in	DC (%)
0	3	—	—	0.541	4	—	—	1.072
1	3	99.992	11825	0.709	4	99.603	252	1.340
2	3	100	∞	0.865	4	99.218	128	1.696
5	3	100	∞	1.335	4	98.736	79	2.646
10	3	99.997	28889	2.125	4	97.640	42	4.194
20	3	99.987	7840	3.652	4	95.082	20	7.784
42	3	100	∞	7.612	4	93.146	15	14.982

3.6.4 A Different Topology: Low Power

We present results with the lower transmission power of -7dBm. This reduces the number of neighbors and increases network diameter (Figure 3.4), yielding a more challenging topology.

To re-establish the ORPL baseline, we repeated experiments in the new topology (Table 3.12, right). ORPL performs close to the high-power setting with only minimal (T-LOW) or J-MWO interference, but shows drastic performance degradation in the presence of J-WIFI, with an almost halved *PDR*.

We ran several Crystal experiments, confirming the trends hitherto observed. However, *DC* increases slightly in all cases, as we must use larger Glossy slots to handle the larger network diameter (Table 3.2). Hereafter, we focus only on $\text{Crystal}_{\text{ND}}^{\text{CH}}$ in the final, most challenging scenarios.

Table 3.17 (left) shows results in the COMBINED$_{split}$ scenario. Comparing against Table 3.14 we see that $N_T = 3$ achieves a *PDR* similar to the high-power case. However, to sustain the most challenging scenario COMBINED$_{16}$ with all channels jammed, the redundancy of the underlying Glossy must be increased to $N_T = 4$ (Table 3.17, right). This enables $\text{Crystal}_{\text{ND}}^{\text{CH}}$ to achieve a *PDR* within 0.14–3% of the high-power case (and notably higher for $U = 1$), above 93% even with all 42 concurrent senders.

These results confirm the effectiveness of our techniques also in the larger-diameter, lower-power setting considered.

3.6.5 Back to Aperiodic Data Collection

We now reconcile the experimental results reported with the original goal of supporting aperiodic, sparse data collection.

We use the traffic profile in Table 3.1 and adapt it for missing values of U in our experiments, replaced by the next higher value available. For instance, the value 606 for $U = 5$ is actually the sum of the epochs with 3, 4, or 5 senders present. This yields worst-case estimates of *PDR* and *DC*, as both increase with U. These are aggregated

Table 3.18: $\text{CRYSTAL}_{\text{ND}}^{\text{CH}}$: performance with the aperiodic, sparse, real-world traffic profile shown in Table 3.1.

	N_T =2		N_T =3	
Scenario	*PDR* (%)	*DC* (%)	*PDR* (%)	*DC* (%)
T-LOW	100	0.105	100	0.119
T-HIGH	100	0.105	—	—
COMBINED$_{16}$	99.487	0.198	99.592	0.263

over the entire dataset as

$$C = \frac{\sum_{u=0}^{N} c(u) \times e(u)}{\sum_{u=0}^{N} e(u)}$$

where $c(u)$ is the value of *PDR* or *DC* for a given number u of concurrent senders (reported in previous sections) and $e(u)$ is the number of epochs in which u concurrent senders are present (from Table 3.1). As the the original dataset uses an epoch $E = 30$s, we re-scaled *DC* accordingly (i.e., $\frac{1}{3}$ of those hitherto shown) to enable a comparison with the performance reported in [31], albeit in a different testbed. In our results (Table 3.18), we consider only the extremes of the interference scenarios we analyzed in the chapter, viz. natural interference and generated interference in the COMBINED$_{16}$ scenario, as these are already sufficient to draw a few interesting observations.

First, in the T-LOW scenario $DC \approx 0.1\%$; this confirms that our results are in line with the per-mille *DC* originally reported in [31]. Interestingly, this is identical to daytime (T-HIGH) when instead the original CRYSTAL behaved erratically, as quoted in the beginning of the chapter. This confirms that the techniques presented in this work effectively combat interference without sacrificing ultra-low power consumption.

Finally, Table 3.18 shows that the *PDR* accrued over the 36-day dataset remains near to 99.5% in COMBINED$_{16}$, which is remarkable given the very challenging nature of this interference scenario. Further, this is achieved with $DC \approx 0.2\%$ depending on N_T. This is twice the baseline established by natural interference, but in absolute terms it is remarkably small w.r.t. the energy consumption commonly reported in the state of the art.

3.7 Related Work

We survey approaches that share our goal of making multi-hop protocols for low-power wireless communication resilient to environmental interference. Notably, interference has also been studied from a security perspective by identifying several types of jamming attacks and related countermeasures. Although not directly related to our contribution, these techniques may inspire alternate resilience mechanisms; we refer the interested reader to [114].

CSMA + channel hopping. Adding channel hopping to combat interference is well accepted in the literature, with recent works modifying standard, CSMA protocol stacks. MiCMAC [98] extends ContikiMAC with channel hopping, resulting in a synchronization-free MAC with high *PDR* under WiFi interference. MiCMAC mechanisms require transmitting and receiving nodes to synchronize in time as well as across channels, increasing latency. Oppcast [99] and MOR [100] offer full-stack alternatives to RPL and MicMAC, combining channel hopping and opportunistic routing to combat high latencies while also escaping high interference.

As MOR code is not available, we offer an informal, numerical comparison with the evaluation in [100], performed on FlockLab [115] with WiFi on one channel and an effective $U = 2.1$. Using this jammed channel plus two free ones, MOR shows the best results: $PDR = 99.35\%$ and $DC = 1.56\%$. We compare to a more challenging scenario with constant, generated WiFi traffic on all channels in our testbed where CRYSTAL shows $PDR = 100\%$ and $DC = 0.559$ for $U = 2$. This *DC* is nearly *three times smaller* w.r.t. MOR, and achieved without any interference avoidance mechanisms. Naturally, with more concurrent packets, the *DC* of CRYSTAL increases, however the same is true for other protocols. Further, in the absence of traffic, a common case in §3.6.5, CRYSTAL maintains $DC < 0.4\%$, levels that duty cycling protocols cannot achieve due to required periodic channel probing. Finally, to manage latency, these protocols hop among few channels, selected during pre-deployment evaluations. In contrast, CRYSTAL$^{\text{CH}}$ can use all channels without affecting its performance, allowing it to adapt to changing interference.

TDMA + channel hopping. TSCH [116] with Orchestra [117] scheduling offers a protocol in which all nodes follow a repeating, slotted schedule, with local and independent slot allocation. The number and type of slots is statically determined, according to expected traffic. Results from Indriya [106] show Orchestra with 47 slots maintains $PDR = 99.99\%$ with an average $DC = 0.4\%$, without interference; in an analogous setting, CRYSTAL consumes twice as much, $DC = 0.8\%$. However, in Orchestra the duty cycle of nodes varies significantly across the network, with nodes closer to the sink reporting much higher values. Further, Orchestra is designed for periodic data, which is critical to statically configure slot parameters. In the aperiodic, dynamic scenarios considered in §3.6.5, Orchestra would over-dimension for the worst case, unable to reduce *DC* under low traffic. Finally, Orchestra has not been evaluated under interference.

Concurrent Transmissions + channel hopping. The combination of channel hopping and concurrent transmissions has also been used to increase parallelism for bulk data dissemination in Splash [118] and Pando [119]. Both protocols also see improvements due to diverse noise levels across channels, but their approaches are not competitive at low data rates.

In the context of the EWSN Dependability Competition [41], three top ranking approaches in 2016 and 2017 [63, 64, 101] perform channel hopping *inside* Glossy, a contrast to the noise resilience mechanisms we designed *on top* of Glossy. However, these solutions were highly specialized for the (single-sender) competition scenario and are not immediately reusable towards our goals. Instead, we evaluated CRYSTAL$^{\text{CH}}_{\text{ND}}$ with concurrent senders and in a wide range of intense interference. Analyzing and exploiting the interplay between Glossy-level channel hopping and our CRYSTAL-level techniques is intriguing, but is beyond the scope of this work, albeit in our research plans.

In the recent years—after the publication of [1] that lays the ground for this chapter—several other protocols leveraging concurrent transmissions in synergy with channel hopping and various additional noise-resistant techniques have been proposed (e.g., [120, 121, 122]). These works repeatedly demonstrated that frequency diversity in combination with concurrent transmissions can unlock at once unprecedented noise resilience and very low latency and energy consumption. As an additional witness to the effectiveness of this synergy, *all the top three winning solutions* of the 2018 and 2019 EWSN Dependability Competition [3, 65, 66, 67, 68] rely on concurrent transmissions enhanced with channel hopping.

3.8 Conclusions and Future Work

This chapter set out to evaluate CRYSTAL's ability to sustain aperiodic, sparse traffic under strong interference, and in turn its applicability to noise-prone applications, e.g., industrial wireless control. As CRYSTAL relies on Glossy, we also offer a noise resilience evaluation for it—effectively providing a first, extensive analysis of CTX performance under interference—along with the two mainstream protocols, RPL and ORPL, we chose as baselines.

Unlike existing works limited to natural WiFi interference, we subjected these protocols also to the stronger noise generated by JamLab-emulated microwave ovens, which exhibit different "on-off" interference patterns where harsh, persistent noise intervals are alternated with inactive periods. In our reproducible and controlled setup we showed that ORPL is very resilient to this type of interference, while CRYSTAL is not. This motivated us to extend it with a combination of channel hopping and noise detection. We demonstrated that our enhanced CRYSTAL$^{\text{CH}}_{\text{ND}}$ protocol achieves unprecedented, near-perfect reliability even against the combination of emulated WiFi and microwave ovens, along with a per-mille radio duty cycle in the aperiodic, sparse traffic targeted by CRYSTAL.

CRYSTAL$^{\text{CH}}_{\text{ND}}$ was later employed as a competing system in the EWSN Dependability Competition [41]; even though not specifically designed to match the peculiar competition scenario, it repeatedly confirmed similar performance, achieving the *second* [3] and *third* [3] place in the 2018 and 2019 edition, respectively.

These results suggest that network wide floods of CTX, if carefully orchestrated, can provide hitherto missing support for the communications demands of aperiodic industrial wireless control systems, which entail *i)* timely and dependable delivery of aperiodic and unpredictable control traffic, *ii)* minimization of the network overhead to ensure ultra-low power consumption, and *iii)* resilience to harsh radio frequency noise. This unlocks intriguing opportunities to co-design new wireless control system architectures that by combining concurrent transmissions in synergy with aperiodic control paradigms, e.g., event-triggered control (ETC), could abate energy costs w.r.t. periodic approaches while retaining similar control performance. In the next chapter we therefore explore this research direction, leveraging concurrent transmissions to co-design the *Wireless Control Bus* (WCB), a long-awaited networking stack for ETC systems.

Regarding future work, a promising avenue is to distill the knowledge gained from the experimental campaigns presented in this chapter into models able to identify the proper Crystal configuration given a known or estimated pattern of interference; this could potentially inform and greatly simplify in-field system configuration. Similarly, these models could be a stepping stone for designing a novel system that, by integrating artificial intelligence strategies in the protocol logic, further enhances Crystal performance, e.g., by fostering a dynamic adaptation of its core parameters to the channel conditions and the application requirements, along the lines of [123].

To contribute in the further development of the CTX research community, we released $\text{Crystal}_{\text{ND}}^{\text{CH}}$ as open source [59], enabling researchers to experiment with it, exploring other ways to enhance its performance or harvest its benefits.

4

The Wireless Control Bus: Multi-hop Event-Triggered Control with Concurrent Transmissions

As discussed in the introduction to Part I, the benefits of wireless communication are so significant that low-power wireless networking is gaining momentum also in traditionally wired domains like industrial control [18, 19, 89, 124], where its adoption is expected to transfigure production processes by enhancing the modularity, scalability, and productivity of industrial sites. Nonetheless, even though low-power wireless technology is extensively employed for *monitoring*, its use for control and automation of plants and processes is still very limited [90]. Key concerns hampering wider adoption are the reliability of communication and the stability and magnitude of its latency. Modern controllers depend on the reliable and timely communication of relatively small data packets containing measurements and commands, generated frequently at the sensors and controller. Guaranteeing these properties is challenging in the large-scale, multi-hop scenarios that are often the main reason for a wireless approach. Moreover, staple applications for wireless control rely on battery-powered sensors, which places energy efficiency in the limelight, as replacing batteries is often costly or impractical. In this respect, it is well-known that radio activity, both listening and transmitting, is the main source of energy consumption.

The development of low-power wireless systems capable of efficiently and reliably harmonise control and network operations, minimising energy costs without impairing control performance, is therefore of utmost importance to foster the adoption of

The contents of this chapter are the basis of the paper submitted to TCPS [75]: M. Trobinger, G. A. Gleizer, T. Istomin, M. Mazo Jr., A. M. Murphy, and G. P. Picco. "The Wireless Control Bus: Enabling Efficient Multi-hop Event-Triggered Control with Concurrent Transmissions". In (2021). arXiv: `2101.10961 [eess.SY]`.

low-power wireless technology for control applications. This is exactly the focus of this chapter, where we explore, for the first time, the synergy of concurrent transmissions with event-triggered control (ETC) to co-design a novel, aperiodic WCS architecture unlocking unprecedented energy savings while preserving desired system performance.

Event-triggered control: A missed opportunity? To facilitate the design of communications and simplify the control performance analysis, most networked control systems (NCS), whether wireless or wired, employ the classical *periodic* sampling of sensor data and update of actuator commands. The choice of sampling period involves a conservative, worst-case analysis of the closed-loop system dynamics. However, this conservative design enters in direct conflict with the objective of reducing energy consumption, enabled by the low-power wireless sensor network (WSN) operation and key to *wireless* NCS (WNCS). To address this limitation, *aperiodic* methods adapting to the dynamic needs of the system have been investigated for a couple of decades (see, e.g., [125]).

A strong surge of interest began in 2007 with the systematic way of designing aperiodic sampling proposed in [43], currently known as *event-triggered control (ETC)*, revolving around the design of a *triggering condition* that only depends on sensor data. While this condition remains unsatisfied, a reference Lyapunov function decays at a certain speed[1]; otherwise, as soon as it is satisfied, sensor data is transmitted and control commands are updated. This procedure guarantees a prescribed decay of the Lyapunov function, serving as a certificate of performance for the control system, while significantly reducing the need for communication and, at least in principle, energy consumption.

Since then, many researchers embraced ETC and contributed to its theoretical foundations [43, 44, 45, 127, 128]. However, its application is still problematic. Although ETC naturally fosters resilience to communication delays [43], this tolerance has its own limitations, and the latency of communication imposes a limit on the achievable performance in terms of convergence rate to an equilibrium. Therefore, minimizing delays remains a critical goal for network stacks supporting ETC. Similar comments hold for reliability, whose crucial role is exacerbated as the entire network must timely and reliably react to the unpredictable violation of triggering conditions for ETC to operate properly.

Guaranteeing these and other properties with a proper network stack is the most significant hampering factor to a wider adoption of ETC. Although wireless implementations of ETC exist [47, 48, 129, 130, 131], these are limited to small-scale, single-hop networks and exhibit poor reliability, high energy consumption, large and unpredictable delays, or a combination thereof, ultimately preventing the overall system to

[1] Lyapunov functions are widely employed in stability and performance analysis and design of control systems. Informally, it can be seen as a mathematical generalization of the energy of a system: it is always positive, it grows with the magnitude of the states, and is zero only at the desired equilibrium point. A decaying Lyapunov function implies that the system is approaching the equilibrium point. For an exposition, see, e.g., [126].

seize the energy savings potentially enabled by ETC. This state of affairs is eloquently summarized in a recent survey on wireless control ([19], p. 22):

> *"While in the control community, many so-called event-triggered estimation and control approaches have been developed in the last two decades, it remains largely unclear whether and how these can be integrated with the communication system and indeed result in demonstrable resource reallocation, savings, or other advantages for wireless systems in practice."*

A wireless control bus for ETC. In this chapter, we answer this question by providing a full-fledged network stack operating in conjunction with ETC, therefore unlocking its remarkable potential for energy savings hitherto hampered by the lack of appropriate communication support.

Our approach exploits concurrent transmissions (CTX), a technique popularized by Glossy [29] that has proven a very effective building block for protocol design, as discussed in Chapter 2 and practically demonstrated in Chapter 3.

CTX-based protocols achieve at once very low latency, high reliability, low energy consumption, and accurate time synchronization, pushing the envelope of what can be achieved by IEEE 802.15.4 and, recently, other low-power wireless radios including BLE [77, 132], UWB [2, 81, 133, 134], and LoRa [78, 79]. Based on efficient network-wide floods, they require neither a MAC nor a routing layer, and their performance is largely unaffected by changes in the topology induced, e.g., by node and link failures. This is a significant departure from conventional techniques (e.g., WirelessHART [6], ISA100.11.a [5], 6TiSCH [7]) that mitigate the packet losses and missed deadlines induced by network vagaries with continuous, high-overhead topology maintenance.

Instead, CTX-based protocols allow for the communication medium to be abstracted into a globally shared bus [30]; application data is broadcast to the entire network and can therefore be read by each node. In our context, this makes centralized control more appealing and even efficient than decentralized and distributed alternatives. Centralized controllers are generally easier to design and provide better performance than controllers accounting for network topology constraints; further, in the specific case of ETC they usually lead to fewer events being triggered. Unfortunately, the use of CTX in control is hitherto largely unexplored, apart from few recent exceptions [91, 92, 124, 135, 136] that however focus on periodic and self-triggered sampling rather than ETC.

Methodology and contributions. To achieve the remarkable potential benefits of CTX-based communications in ETC, *co-design* is fundamental. The control algorithm must work hand-in-hand with the underlying network stack to seize opportunities to reduce the radio active time while ensuring the timeliness and reliability key to control performance. In ETC, control update times are *not* defined a priori; sensors decide on-the-fly whether to send updated readings based on their triggering condition. This *in theory* reduces communication w.r.t. classic control approaches; *in practice*, it must be supported by a network stack capable of *i)* minimizing network overhead during the control idle times, and *ii)* promptly react to triggered events by ensuring timely and reliable collection of sensor readings at the controller and dissemination of updated actuation commands.

We address these challenges with the *Wireless Control Bus (*WCB*)*, a novel protocol that, to the best of our knowledge, is the first supporting multi-hop communication for ETC, and does so efficiently and reliably.

We first summarize the technical foundations of ETC and, motivated by the co-design of control and communication in WCB, put forth a side contribution further reducing communication via rejection of step-disturbances (§4.1). We then illustrate how the design of WCB (§4.2) exploits CTX to meet the above requirements of ETC w.r.t. latency, reliability, and energy efficiency. Moreover, we present a WCB variant that can easily accommodate conventional periodic strategies, endowing them with similarly unprecedented performance and ultimately fostering a holistic approach to control design enabled by a *single* network stack.

We demonstrate the effectiveness of our solutions via a water-irrigation system (WIS) test case, for which we define an ETC-based control strategy (§4.3). A WIS typically extends for kilometers, likely requiring multi-hop communication, in turn demanding complex decentralized or distributed control strategies, as in [137]. In contrast, our combination of WCB and ETC enables a simpler centralized control, as we show experimentally. In this respect, a realistic evaluation is a challenge per se, as we are not aware of large-scale WIS testbeds. Small-scale ones, e.g., the double-tank system [138], are widely adopted but rely on a single-hop, star topology, unsuited to evaluate the multi-hop systems envisioned for industrial wireless control and targeted by this work.

We overcome these limitations with a secondary contribution: the design of a cyber-physical testbed (§4.4) that adopts a real-time, network-in-the-loop approach integrating *i)* a Simulink *model* emulating the physical system, and *ii) real* embedded devices acting as sensors, actuators, forwarders, and controller, executing our control and protocol stack and interacting only wirelessly. We experiment with two distinct networks, where we analyze the sensitivity of WCB to its parameters (§4.5), identify the configuration we use in our extensive experimental campaign, and assess the impact of different scales and topologies on the performance of our ETC system.

The experimental results (§4.6) demonstrate the effectiveness of our approach. The quality of the control achieved by ETC over WCB is virtually the same as periodic sampling. However, it comes at *a fraction* of communication costs; sample count is reduced by $> 89\%$, yielding a $> 62\%$ reduction in radio-on time w.r.t. periodic control—far more than previously observed in the ETC literature [139] in significantly more constrained setups. This confirms that WCB not only provides a network stack, hitherto missing, enabling ETC in multi-hop networks, but also effectively translates the reduction of control traffic enabled by ETC into corresponding savings in energy consumption.

The chapter ends with a summary of related work (§4.7) and brief concluding remarks outlining opportunities for future work (§4.8) on WCB that, similarly to all other contributions presented in this dissertation, we intend to release publicly as open source.

4.1 Event-triggered Control

Event-triggered control (ETC) is a sampling strategy in which the update of sensor data to feedback controllers and of control commands to actuators is determined *on-the-fly* by a *triggering condition*. This is a drastic departure from time-triggered control, which includes the classic periodic control.

In a nutshell, when something relevant happens on the state of a dynamic system, the sensors communicate their most recent values to the controller; otherwise, these values are held constant, and actuators typically also hold their positions. Intuitively, data is sampled *only when needed*, reducing the communication induced by control. In practice, determining when fresh data is needed is somewhat involved and requires control theory to ensure stability and good performance.

We formally describe ETC, including equations for a distributed implementation suited to CTX. In doing so, we also present two contributions: *i)* a generalization of the decentralized ETC strategy in [44] to a broader class of triggering conditions and sensor node arrangements (§4.1.3), and *ii)* an adaptation of unperturbed ETC strategies to the problem of step disturbance rejection (§4.1.4).

4.1.1 Sample-and-hold Control

Hereafter, we consider a linear time-invariant (LTI) system with measurable states of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{w}(t), \tag{4.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the vector of states, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the vector of control inputs, $\boldsymbol{w}(t) \in \mathbb{R}^p$ is the vector of exogenous unmeasured disturbances, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{E}$ are known system matrices of appropriate dimensions. In this work, we assume that all states are measured by sensors. For digital implementation, we consider a state-feedback controller realized in a sample-and-hold fashion:

$$\boldsymbol{u}(t) = \boldsymbol{K}\hat{\boldsymbol{x}}(t), \tag{4.2}$$

where $\boldsymbol{K}$ is a control gain matrix to be designed, and $\hat{\boldsymbol{x}}(t)$ is the sampled state, which satisfies, for a sequence of sampling times $\{t_i\}_{i\in\mathbb{N}}$,

$$\hat{\boldsymbol{x}}(t) = \boldsymbol{x}(t_i), \forall t \in [t_i, t_{i+1}). \tag{4.3}$$

We say that the obtained closed-loop system is *globally exponentially stable* if, for every initial condition $\boldsymbol{x}(0)$, all of its solutions satisfy $|\boldsymbol{x}(t)| \leq M|\boldsymbol{x}(0)|\mathrm{e}^{-\rho t}$ for some $0 \leq M < \infty$ and $\rho > 0$, where ρ is called the *decay rate* of the system.

When using *periodic sampling*, the sequence $\{t_i\}_{i\in\mathbb{N}}$ satisfies $t_i = ih$, for some designed sampling time h. In ETC, the sequence of sampling times is *not* known a priori; instead, it is generated based on some designed *triggering condition* dependent on the states of the system. Although there is a vast literature on ETC, this section focuses on mechanisms enabling two important practical aspects for WNCS:

1. *Triggering conditions can be checked periodically.* This allows for an efficient scheduling of sleep times. Classical ETC requires instead continuous monitoring of triggering conditions, forcing sensors to be always active and preventing energy savings.
2. *Triggering conditions can be checked locally on the sensor nodes.* The alternative of checking them on the controller side would require sensors to send data to it periodically, which would eliminate any communication-related energy savings.

4.1.2 Periodic Event-triggered Control

Using the framework of [45], we define a periodic event-triggered state-feedback system as the one captured by equations (4.1)–(4.3) with the triggering times satisfying

$$t_{i+1} = \inf \left\{ t = kh > t_i, k \in \mathbb{N} \;\middle|\; \begin{bmatrix} \boldsymbol{x}(t) \\ \hat{\boldsymbol{x}}(t) \end{bmatrix}^{\mathsf{T}} \boldsymbol{T} \begin{bmatrix} \boldsymbol{x}(t) \\ \hat{\boldsymbol{x}}(t) \end{bmatrix} > \epsilon^2 \right\}, \tag{4.4}$$

where $\boldsymbol{T}$ is a triggering matrix to be designed and ϵ is a design parameter whose value controls the size of the terminal set to which the system converges. When $\epsilon = 0$, the system converges and stabilizes at the desired equilibrium. A small $\epsilon > 0$ increases the inter-sample times at the expense of stabilizing a set around the equilibrium, of size proportional to ϵ. When persistent external disturbances $\boldsymbol{w}$ are present, one cannot stabilize the origin; setting $\epsilon > 0$ is necessary to prevent excessive sampling precisely when the system is essentially under control, i.e., close to equilibrium.

Several tools are available to verify the stability of the closed-loop system using a given triggering matrix $\boldsymbol{T}$. We recall now one of the results from [45]:

Theorem 1 ([45], Theorem III.4). *With $\epsilon = 0$ and $w(t) \equiv 0$, the periodic ETC (PETC) system* (4.1)–(4.4) *is globally exponentially stable (GES) with decay rate ρ if there exist symmetric matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$, and scalars $\alpha_{ij} \geq 0, \beta_{ij} \geq 0$, and $\kappa_i \geq 0$, $i, j \in \{1, 2\}$, satisfying*[2]

$$\mathrm{e}^{-2\rho h}\boldsymbol{P}_i - \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{P}_j\boldsymbol{A}_i + (-1)^i\alpha_{ij}\boldsymbol{T} + (-1)^j\beta_{ij}\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{T}\boldsymbol{A}_i \succeq \boldsymbol{0}, \quad \forall i, j \in \{1, 2\}, \quad \textit{and}$$

$$\boldsymbol{P}_i + (-1)^i\kappa_i\boldsymbol{T} \succ \boldsymbol{0}, \quad \forall i \in \{1, 2\},$$

$$\textit{where } \boldsymbol{A}_1 := \begin{bmatrix} \boldsymbol{A} + \boldsymbol{BK} & \boldsymbol{0} \\ \mathbf{I} & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{A}_2 := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{BK} \\ \boldsymbol{0} & \mathbf{I} \end{bmatrix}.$$

We use this result in our test case (§4.3) to design appropriate triggering conditions, i.e., a matrix $\boldsymbol{T}$ that guarantees appropriate control performance for a given sampling time h.

4.1.3 Distributed Event-triggered Conditions

The triggering condition in equation (4.4) is, in its most general form, a centralized one, i.e., all states are needed to determine when to sample. However, when sensors

[2] For a symmetric matrix $\boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}}$, we say that $\boldsymbol{A} \succ \boldsymbol{0}$ ($\boldsymbol{A} \succeq \boldsymbol{0}$) if it is positive-(semi)definite.

are remotely located w.r.t. each other, this approach becomes impractical. Fortunately, decentralized triggering conditions exist that address this issue. Here we focus on the strategy proposed in [44], consisting of three key steps posing corresponding requirements on the network stack supporting control:

1. Each sensor has its own triggering condition, which can trigger a controller update *independently* of readings from other sensors.
2. Upon one sensor triggering, *all* others must transmit their up-to-date readings to the controller.
3. Finally, the controller updates its control command and sends it to the actuators.

The following type of triggering condition is used as a starting point in [44]:

$$|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)| > \sigma\,|\boldsymbol{x}(t)|\,, \tag{4.6}$$

where σ is a triggering parameter and $|\cdot|$ is the Euclidean norm. This condition, introduced by the seminal work in [43], essentially compares the *sampling error* $\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ against the state values themselves; if the error is large enough, it is time to update the measurements at the controller.

The main observation in [44] is that by rewriting equation (4.6) one obtains the implication:

$$\sum_{i=1}^{n}(x_i(t) - \hat{x}_i(t))^2 - \sigma^2 x_i^2(t) > 0 \Rightarrow \bigvee_{i=1}^{n}\left((x_i(t) - \hat{x}_i(t))^2 - \sigma^2 x_i^2(t) > \theta_i\right) \tag{4.7}$$

as long as $\sum_{i=1}^{n}\theta_i = 0$ for n state variables. This enables using each of the i-th conditions in the right-hand side of equation (4.7) independently at each sensor. The triggering parameters θ_i can be designed offline or adapted online. Hereafter, we focus on the former; details of their computation are found in [44].

Observe that equation (4.6) can be cast in the form of (4.4) with $\boldsymbol{T} = \begin{bmatrix}(1-\sigma^2)\mathbf{I} & -\mathbf{I}\\ -\mathbf{I} & \mathbf{I}\end{bmatrix}$ and $\epsilon = 0$. Thus, a simple generalization of the approach described above is possible, to include a larger class of triggering conditions of the form of equation (4.4), where more parameters (i.e., all elements of $\boldsymbol{T}$) than simply σ are to be designed. This introduces additional design flexibility for the triggering conditions, which can be used to further reduce the amount of communication triggered by the system.

First, denote the sampling error $\boldsymbol{e} := \hat{\boldsymbol{x}} - \boldsymbol{x}$. Assume $q \leq n$ sensor nodes, each measuring one or more state variables, and denote by $\mathcal{I}_j \subseteq \{1, 2, ..., n\}$ those measured by node j with $\bigcap_{j=1}^{q}\mathcal{I}_j = \emptyset$, i.e., each state variable is measured by only one node. Then, a triggering condition of the form

$$\boldsymbol{e}(t)^{\mathsf{T}}\boldsymbol{M}\boldsymbol{e}(t) - \boldsymbol{x}(t)^{\mathsf{T}}\boldsymbol{N}\boldsymbol{x}(t) > \epsilon^2, \tag{4.8}$$

is decentralizable if the triggering matrices $\boldsymbol{M} = \boldsymbol{M}^{\mathsf{T}}$ and $\boldsymbol{N} = \boldsymbol{N}^{\mathsf{T}}$ have the following structure: an element $M_{ii'}$ ($N_{ii'}$) is nonzero if and only if i and i' belong to the same set $\mathcal{I}_j$ for some sensor node j. Then, denoting by $\boldsymbol{x}_j, \boldsymbol{e}_j, \boldsymbol{M}_j$ and $\boldsymbol{N}_j$ the subvectors and submatrices containing the rows and columns $\mathcal{I}_j$ of $\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{M}$ and $\boldsymbol{N}$, we obtain that equation (4.8) implies:

$$\bigvee_{j=1}^{q} (\boldsymbol{e}_j(t)^{\mathsf{T}} \boldsymbol{M}_j \boldsymbol{e}_j(t) - \boldsymbol{x}_j(t)^{\mathsf{T}} \boldsymbol{N}_j \boldsymbol{x}_j(t) > \theta_j), \qquad \text{with } \sum_{j=1}^{q} \theta_j = \epsilon^2. \tag{4.9}$$

To make triggering as infrequent as possible, during design one may want to maximize some norm of $\boldsymbol{N}$ and minimize $\boldsymbol{M}$, so that the negative term in equation (4.8) dominates the inequality. Note that the triggering condition (4.8) admits the form in equation (4.4) with $\boldsymbol{T} = \left[\begin{smallmatrix} \boldsymbol{M}-\boldsymbol{N} & -\boldsymbol{M} \\ -\boldsymbol{M} & \boldsymbol{M} \end{smallmatrix}\right]$, therefore Theorem 1 can be used to verify global exponential stability. This theorem can also be used to co-design, and optimize for sparse sampling, the matrices $\boldsymbol{P}_i$ and the triggering matrices $\boldsymbol{M}$ and $\boldsymbol{N}$; by fixing the values of κ_i, α_{ij}, and β_{ij}, the problem becomes a linear matrix inequality (LMI) that can be easily solved with existing optimization software. To prevent the triggering condition from being repeatedly violated after the previous sample, when $\boldsymbol{e}(t_i) = \boldsymbol{0}$, $\boldsymbol{N}$ must be positive semidefinite.

4.1.4 The Problem of Disturbance Rejection

The ETC mechanisms presented in this section are associated with the problem of stabilizing the origin, disregarding the effects of disturbances. Still, the presented triggering strategies also give disturbance attenuation properties in the case of linear systems. For example, sufficient conditions to verify a finite $\mathcal{L}_\infty$ gain are also present in [45].

In disturbance rejection problems, like the one we address in the WIS example on which we evaluate our solution (§4.3), there is an important specificity: with the appropriate control design, one can ensure that a set of states (the control outputs $\boldsymbol{y}(t) \in \mathbb{R}^p, \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$) still converge to zero; the remaining states also converge, but to some unknown signal dependent on the disturbances (constant values in the case of step disturbances).

If the objective is to stabilize the system to a given reference $\boldsymbol{x}^*$, the general approach to event design is to perform a change of coordinates $\tilde{\boldsymbol{x}} := \boldsymbol{x} - \boldsymbol{x}^*$, which renders the problem again stabilizing $\tilde{\boldsymbol{x}}$ to the origin. With this change of coordinates, note that the sampling error component does not change, i.e., $\hat{\boldsymbol{e}} = \hat{\tilde{\boldsymbol{x}}} - \tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \boldsymbol{x} = \boldsymbol{e}$. Condition (4.8) becomes

$$\boldsymbol{e}(t)^{\mathsf{T}} \boldsymbol{M} \boldsymbol{e}(t) - (\boldsymbol{x}(t) - \boldsymbol{x}^*)^{\mathsf{T}} \boldsymbol{N} (\boldsymbol{x}(t) - \boldsymbol{x}^*) > \epsilon^2. \tag{4.10}$$

In the case of step disturbance rejection, some of the components of $\boldsymbol{x}^*$ are unknown and vary depending on the disturbance. This makes it impossible to implement equation (4.10) in its most general form. However, if one constrains the elements of $\boldsymbol{N}$ associated with the unknown entries of $\boldsymbol{x}^*$ to be zero, these terms do not appear in the equation, and the triggering condition is implementable regardless of the disturbance levels. Mathematically, the matrix on the second term of equation (4.10) takes the form $(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})\boldsymbol{N}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})$, and the triggering condition can be implemented as

$$\boldsymbol{e}(t)^{\mathsf{T}} \boldsymbol{M} \boldsymbol{e}(t) - \boldsymbol{y}(t)^{\mathsf{T}} \boldsymbol{C}^{\mathsf{T}} \boldsymbol{N} \boldsymbol{C} \boldsymbol{y}(t) > \epsilon^2, \tag{4.11}$$

which can be decentralized to take the form in equation (4.9). To verify stability, one can use Theorem 1 with

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{M} - (\boldsymbol{C}^\mathsf{T}\boldsymbol{C})\boldsymbol{N}(\boldsymbol{C}^\mathsf{T}\boldsymbol{C}) & -\boldsymbol{M} \\ -\boldsymbol{M} & \boldsymbol{M} \end{bmatrix}.$$

4.2 Designing the Wireless Control Bus

The main focus of ETC is to avoid communication during steady-state, while preserving correct and timely control outside of it. From a network standpoint this means that 1) when control traffic is absent, network overhead should be minimized; otherwise, 2) the collection of sensor readings at the controller and consequent dissemination of actuation commands should occur timely and reliably. These requirements, already challenging when taken individually, are even harder to fulfill when combined; a quiescent network, ideal to minimize consumption, is intrinsically at odds with a reactive and reliable one. It is therefore not surprising that a wireless network stack efficiently supporting ETC is still missing, hampering the practical adoption of this control approach.

WCB tackles this challenge by relying on concurrent transmissions (Chapter 2), whose peculiar properties are exploited to efficiently cater for the specific needs of ETC (§4.2.1) and, within the same protocol framework, also of traditional periodic control paradigms (§4.2.2).

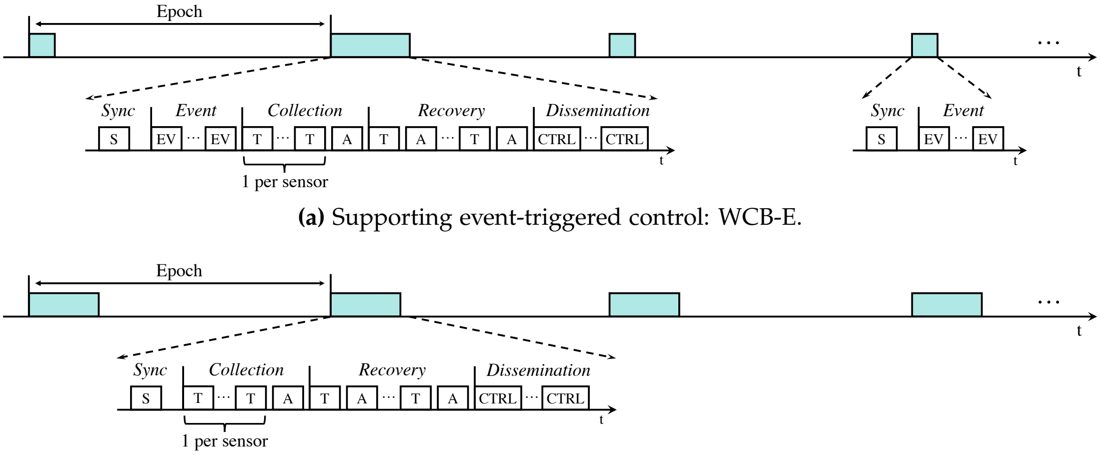
4.2.1 A Network Stack for Event-Triggered Control

A large body of CTX literature leverages Glossy floods as primitive building blocks, composing and scheduling them differently in a distributed fashion, and exploits either or both non-destructive interference and the capture effect (§2.1) depending on the protocol goals at hand. WCB adopts a similar approach, as described next.

Core concepts. Communication in WCB is structured around non-overlapping time *slots*, each containing a separate Glossy flood, potentially initiated by different nodes. The same sequence of time slots repeats at all nodes with a fixed interval called *epoch*, characterized by a very short initial active portion where communication occurs, and a much longer one where nodes turn off their radio and remain in sleep mode to spare energy.

This structure, common to many CTX-based systems, relies on the accurate, network-wide time synchronization enabled by Glossy (§2.2) as part of its operation, and effectively *abstracts the multi-hop wireless network into a shared control bus with time-slotted access*. This simplifies significantly the development of the overall control system by removing all the complexity typically associated with multi-hop networks (e.g., at the MAC and routing layers) and, at the same time, ensuring high determinism in terms of latency and reliability—key for control design and performance.

Time slots can be *i)* dedicated to a single flood by one sender, *ii)* used by multiple senders concurrently flooding the same packet, or *iii)* by multiple senders flooding different packets competing in the same slot. Although in all cases one packet is



(a) Supporting event-triggered control: WCB-E.

(b) Supporting periodic control: WCB-P.

Figure 4.1: The wireless control bus, WCB.

received with high probability, experience with CTX-based systems shows that they typically offer decreasing degrees of reliability [10, 31]. WCB balances the pros and cons of each slot type depending on the target functionality, detailed next.

Protocol Phases. The active portion of a WCB epoch is structured in the following groups of functionally-related slots, or *phases* (Figure 4.1):

1. *Synchronization.* CTX require tight time synchronization, which is also useful to establish a common time reference for control. However, prolonged sleeping periods—the main asset in reducing energy consumption—significantly increase clock drift. Therefore, each WCB epoch begins with a S slot initiated by the controller and exploited by all nodes to realign their schedule.

2. *Event.* This phase is key to efficient ETC support. After synchronization, each sensor node acquires its measurements and evaluates the triggering condition in equation (4.11), §4.1. If this holds, a special and very short event notification packet—the same at all nodes—is flooded in one or more EV slots. Multiple events may be generated simultaneously at different nodes. However, due to the properties of CTX (§2.1), this packet is received with very high reliability at all nodes, informing them *at once* of the need to participate in the subsequent network-wide data collection (left schedule, Figure 4.1a), as the stability of the system might be compromised and a reaction needs to be timely taken. Otherwise, if no event is generated, the system is in steady-state and the nodes can safely enter sleep for the remaining portion of the epoch (right schedule, Figure 4.1a).

3. *Collection.* Sensors report their readings as a sequence of T slots, each reserved to a sensor node performing an isolated flood. At the end, the A slot is dedicated for an acknowledgment flood by the controller, containing a bitmap denoting which sensor packets have been successfully received. Thanks to the reliability of CTX, most of the times all reports are gathered, and all nodes can enter sleep until the dissemination phase (step 5.).

4. *Recovery.* In the rare cases where a sensor node does not receive an acknowledgment or realizes that its packet is not confirmed in the bitmap, the node attempts retransmission in the subsequent T slot. Unlike collection, where each node transmits in a designated slot, during recovery unacknowledged sensors *compete* in the same T slot with concurrent floods for their missed packets. Again due to the properties of CTX (§2.1), one of these packets reaches with high probability the controller, which updates the acknowledgment bitmap and floods it back in the A slot, effectively eliminating one of the competing nodes from the next TA slot pair. This alternating sequence repeats until the controller acknowledges all packets, allowing nodes to safely enter sleep until dissemination, or a pre-defined number R of TA pairs is executed.

5. *Dissemination.* After collecting sensor readings, the controller generates the actuation commands. In the unlikely case where some readings are still missing after recovery, their values from the previous collection are employed by the controller. This is the choice best aligning with the properties of ETC (§4.1), although alternative ones can be easily integrated, if required. Actuation commands are packed in a single packet and disseminated in one or more CTRL slots by a controller-initiated flood; actuators apply the received commands upon their arrival. We *always* include commands for *all* actuators, even when their state is unchanged w.r.t. the previous dissemination, as this provides actuators with multiple chances to receive occasionally missed commands. Dissemination is the last phase of the epoch active portion; upon completion of the last CTRL flood, the network automatically deactivates and all nodes enter sleep mode.

Ensuring Reliability. Each phase exploits different mechanisms to guarantee packet delivery. Recovery exploits an acknowledgment slot A after a T slot, enabling competing nodes to determine whether their packet has been received. This technique has proven very effective (Chapter 3, [31]) when the number of concurrent transmitters is a priori unknown. Nevertheless, in the collection phase it would double the number of slots required and therefore latency and energy consumption. Instead, we exploit a priori knowledge that *all* sensors nodes must transmit, and send a single, cumulative acknowledgment in the A slot at the end of collection, itself triggering recovery only when needed.

The mechanisms above are effective when packets must be delivered to a *single* node—the controller—that can signal their failed receipt. However, they are impractical when packets must reliably reach *multiple* nodes, as in the event and dissemination phases. In these cases, we exploit *redundancy* as a simple yet effective technique to increase reliability, and repeat the EV or CTRL multiple times. The number of repetitions is crucial, as it governs the tradeoffs between reliability and energy consumption; we analyze this parameter experimentally in §4.5.

Finally, we exploit *channel hopping* to further increase resilience to interference, common in industrial scenarios but also in indoor settings (e.g., due to WiFi) like those in our experiments (§4.4). As WCB nodes execute the same schedule in lockstep, even during the dynamic recovery portion, the frequency channel to be used in each slot can change following a globally-known hopping sequence, a technique known to significantly reduce the impact of interference as shown in Chapter 3.

4.2.2 One Wireless Bus to Rule Them All: Periodic Control over WCB

Our stated goal for the design of WCB is to efficiently support ETC. Nevertheless, our protocol can be easily tailored to periodic control by regarding it as a special case of ETC in which the triggering condition is violated during *all* epochs. This renders the dynamic and distributed coordination offered by the event phase superfluous, leading to the schedule in Figure 4.1b. Hereafter, we refer to this specific variant targeting periodic control as WCB-P whenever necessary to distinguish it from the original protocol targeting ETC (Figure 4.1a), itself referred to as WCB-E.

Although the modifications leading to WCB-P are simple, their impact should not be underestimated. On one hand, the dedicated support offered by WCB-E to ETC remains crucial. The active periods in WCB-P are generally longer than in WCB-E, resulting in significantly less energy-efficient communication, as hinted at by the larger active portions of the former in Figure 4.1 and quantitatively shown in our experimental evaluation (§4.6). On the other hand, due to the specific application and control requirements, periodic control may be preferable to ETC. In these cases, the efficiency and performance offered by WCB-P over multi-hop networks is unprecedented, as briefly discussed in §4.7. Further, the ability to use the *same* protocol stack for both flavors of control, ETC and periodic, is a tremendous asset. Not only it greatly reduces the complexity of control design and implementation, but also fosters a holistic approach where the selection of the best control strategy is driven solely by application requirements rather than the lack of a suitable network stack.

4.3 Test Case: A Water Irrigation System

To validate experimentally WCB in a realistic scenario, we use a water irrigation system (WIS) as our test case. A WIS is constituted by a set of pools, often a few kilometers long, connected to one other with controllable gates whose movement regulates the levels of each pool, providing customers with a relatively constant supply. Without communication between neighboring gates, each gate regulates the level of the pool immediately downstream or upstream without knowledge of what happens on the neighboring pools, in what is known as decentralized control. In [137] and [140], it is noted that decentralized control has several limitations that can waste water due to spillovers. These references suggest the use of more interconnected types of control such as centralized and distributed control architectures, in which information from neighboring pools (or all pools in the centralized case) is shared to improve control. With distances on the order of kilometers to be covered and the typical lack of existing wired infrastructure in these areas, WIS are one of the prototypical applications of control over multi-hop wireless networks.

Here we describe our test case, which builds on a real scenario [140]. We then present the periodic event-triggered control (PETC) design that is the basis of our experiments. It is not our intention in this chapter to provide a complete solution to WIS;

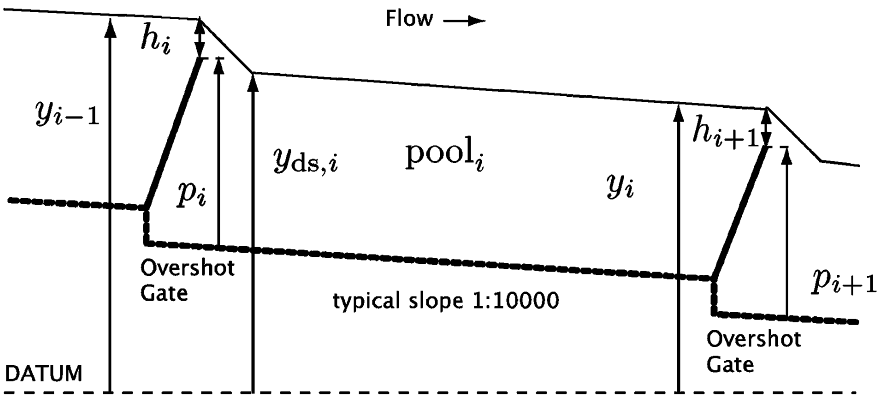


Figure 4.2: A section of an open-water channel with overshot gates (from [137]).

Table 4.1: Parameters of the WIS models (4.13) and (4.14): delay (τ_i), surface area (α_i), and dominant wave frequency (φ_i).

	Pool				
	1	2	3	4	5
τ_i **(min)**	4	2	4	4	6
α_i **(m^2)**	6492	2478	6084	5658	7650
φ_i **(rad/min)**	0.48	1.05	0.48	0.48	0.42

instead, our goal is to use this example as a proof-of-concept for the combination of ETC and WCB presented here. Therefore, we design a simple centralized state-feedback controller that captures the essence of the centralized control problem and allows us to showcase a centralized ETC solution over wireless. Control solutions considering more practical design criteria for WIS are available in, e.g., [137, 141].

4.3.1 System Description and Modeling

In our test case, we consider a WIS composed of multiple pools connected in series; a lateral view is depicted in Figure 4.2. The control problem is to regulate the levels of each pool to their setpoints by adjusting the position of the gates. Opening the gates increases the flow from pool $i-1$ to pool i, contributing to a reduction of level y_{i-1} and an increase of y_i. External off-take disturbances come mostly from end-users, and typically occur downstream in each pool. The control objectives w.r.t. level regulation are [137] *i)* avoiding losses due to spillovers *ii)* keeping levels close to the setpoint to avoid oversupplying, and *iii)* preventing fluctuations that happen when dormant waves are excited.

Accurate models of open water dynamics are very complex. For control design, we can use a simpler one capturing the first modes of wave phenomena via the conservation of mass principle:

$$\pi_i\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)y_i(t) = \gamma_i h_i^{3/2}(t-\tau_i) - \gamma_{i+1}h_{i+1}^{3/2}(t) - d_i(t), \qquad (4.12)$$

where h_i is the relative height above gate i (Figure 4.2), d_i is the total flow of off-take disturbances, τ_i is the time for water to traverse the pool length, and γ_i is a parameter depending on the pool and gate geometry. The model dynamics are captured by a polynomial $\pi_i(\cdot)$: higher orders yield more accurate models. We assume the flow $u_i(t) = \gamma_i h_i^{3/2}(t)$ over gate i can be directly manipulated[3], making equation (4.12) linear. For control design, a first-order polynomial π_i suffices [137, 141]

$$\alpha_i \dot{y}_i(t) = u_i(t - \tau_i) - u_{i+1}(t) - d_i(t), \tag{4.13}$$

where α_i is the pool surface area. However, this model is too simplistic for simulation, an integral part of the experimental setup (§4.4) supporting our combined evaluation of the control and network layers (§4.6). Therefore, as in [137], we use a third-order polynomial $\pi_i(\cdot)$ for the simulated plant:

$$\frac{\alpha_i}{\omega_{n,i}^2}(\dddot{y}_i(t) + 2\zeta_i\omega_{n,i}\ddot{y}(t) + \omega_{n,i}^2\dot{y}(t)) = u_i(t - \tau_i) - u_{i+1}(t) - d_i(t), \tag{4.14}$$

where ζ_i and $\omega_{n,i}$ (satisfying $\varphi_i = \omega_{n,i}\sqrt{1-\zeta_i^2}$, for φ_i the dominant wave frequency), represent the first-mode wave damping ratio, and natural frequency of pool i, respectively. In our test case, we consider a string of five pools representing a section of a water channel in New South Wales, Australia. The characteristics of this setup and related parameters (Table 4.1) are found in [140]. Moreover, we set the additional parameter $\zeta_i = 0.0151$ for all i, as in [143].

4.3.2 Event-triggered Control Design

For ETC design, we apply the principle of separation of concerns between control design and cyber-physical implementation. The controller is designed as a continuous-time controller, for which many methods are available. Then, a sampled-data implementation is devised, which must consider the imperfections of the communication channel to retain some given performance specifications. This prevents changes (e.g., in network technology, topology, nodes, etc.) from requiring a complete redesign of the controller. In our case, this is achieved with the following design procedure:

1. design a centralized state-feedback controller that rejects step disturbances;
2. select the sampling time h for monitoring and event-checking; and
3. design the distributed event-triggering parameters $\boldsymbol{M}_j, \boldsymbol{N}_j, \theta_j$ that achieve similar performance to the continuous-time controller (§4.1).

To design a centralized ETC for the WIS in §4.3.1, we need a state-space description of the system in equation (4.13). To this end, we replace the time-delay by its Padé approximation of order $(1, 1)$, as in [137], and extend the model with states $x_{3,i}$ integrating y_i, to enable rejection of persistent off-take disturbances by the controller. A state-space representation of the resulting model is given by:

$$\dot{x}_{1,i} = -\frac{1}{\tau_i}x_{2,i} - \frac{1}{\alpha_i}(u_i + u_{i+1} + d_i), \qquad \dot{x}_{2,i} = -\frac{2}{\tau_i}x_{2,i} + \frac{4}{\alpha_i}u_i, \qquad \dot{x}_{3,i} = x_{1,i}, \tag{4.15}$$

[3] An example of actuating device in this context is FlumeGate©, by the company Rubicon [142].

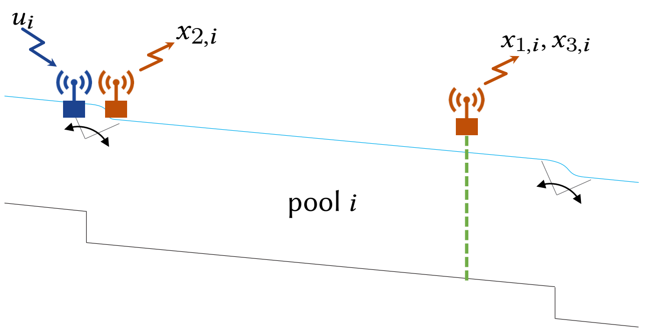


Figure 4.3: Data communicated to/from nodes at pool i. The dashed green line denotes a height measurement sensor, while L-shaped gray elements denote gates with flow control and measurement capabilities.

where $x_{1,i} := y_i$, $x_{2,i}$ can be regarded[4] as a low-pass filter on the flow u_i, and $u_6(t) = 0$, $\forall\, t$, i.e., there is no controlled gate at the downstream side of the last pool. The variables $x_{2,i}$ and $x_{3,i}$ can be locally computed at the flow and height measurement nodes, respectively.

With this model, one can use standard state-space methods for control design. For our test case, we designed a linear-quadratic regulator (LQR) using diagonal weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, with $\boldsymbol{R} = \mathbf{I}$ and $\boldsymbol{Q}$ with diagonal entries (1250, 1250, 2500, 5000, 7500) for $x_{1,i}$, 0 for $x_{2,i}$, and (1.25, 1.25, 2.5, 5, 7.5) for $x_{3,i}$. These values were tuned to achieve a uniform convergence across pools, a trade-off between speed of the state convergence and magnitude of control action, and robustness w.r.t. the natural frequency of oscillation of the pools. The GES decay rate (Theorem 1) of the continuous-time closed loop system is ρ = 0.007 min^{-1}.

Figure 4.3 illustrates how control data is communicated wirelessly. The height sensor node also performs the integration locally to compute $x_{3,i}$. The gate has one node to receive control inputs u_i and one to compute the filtered flow value $x_{2,i}$ and send it to the controller. For the 5-pool system we consider, a total of 10 sensor and 5 actuator nodes are used. The height setpoints are assumed to be locally available to the height device; hereafter, $x_{1,i} = y_i - y_i^*$, i.e., control regulates deviations of height w.r.t. its setpoint, assumed to be set constant throughout the experiment. Figure 4.4 shows a block diagram for the complete control system; note how the controller is a separate node.

We choose the fundamental sampling period $h = 1$ min as in [143], where this value is used for short pools up to 3200 m, as in our setup. As for ETC, we solve iteratively the LMIs in Theorem 1 to find matrices $\boldsymbol{M}_j$ and $\boldsymbol{N}_j$ achieving a high sampling performance (§4.1.3). The triggering parameters θ_j are tuned to further improve the latter in a trade-off with steady-state error, for which a magnitude of 1 cm is deemed acceptable. Figure 4.5 shows the values of $\boldsymbol{M}_j$, $\boldsymbol{N}_j$, θ_j. Nodes 1–5 represent height sensors, with matrices partitioned according to $[x_{1,j}\ x_{3,j}]$, while nodes 6–10 represent filtered flow ($x_{2,j}$) sensors. The resulting decay rate, satisfying Theorem 1, is ρ = 0.006 min^{-1}.

[4] Alternatively, it can be viewed as the Padé approximant of the Smith predictor for the subsystem

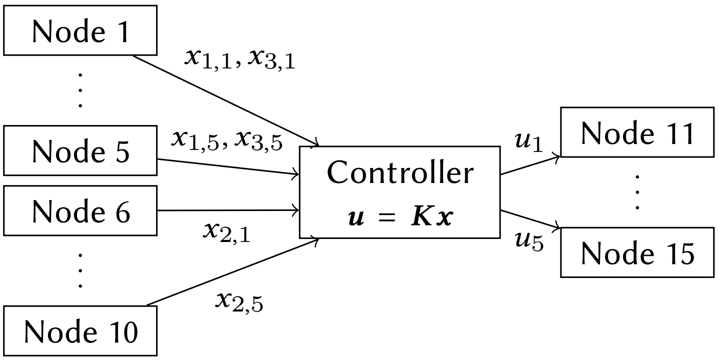


Figure 4.4: Control data diagram for the 5-pool system. Each of nodes 6–10 is co-located with nodes 11–15, respectively; therefore, they can be hosted by the same physical device.

$$M_1 = \begin{bmatrix} 0.621 & 0.0030 \\ 0.003 & 0.0001 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0.414 & 0.003 \\ 0.003 & 0.0002 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1.854 & -0.083 \\ -0.083 & 0.13 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 2.48 & 0.012 \\ 0.012 & 0.001 \end{bmatrix},$$

$$M_5 = \begin{bmatrix} 7.639 & 0.027 \\ 0.027 & 0.006 \end{bmatrix}, \quad M_6 = 0.1147, \quad M_7 = 0.0841, \quad M_8 = 0.2337,$$

$$M_9 = 0.5352, \quad M_{10} = 1.4786, \quad N_1 = \begin{bmatrix} 2.5 \times 10^{-8} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0.0503 & 0 \\ 0 & 0 \end{bmatrix},$$

$$N_3 = \begin{bmatrix} 1.2 \times 10^{-8} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_4 = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_5 = \begin{bmatrix} 0.9497 & 0 \\ 0 & 0 \end{bmatrix}, \quad N_6 = 0,$$

$$N_7 = 0, \quad N_8 = 0, \quad N_9 = 0, \quad N_{10} = 0,$$

$$\theta_1 = 0.415, \quad \theta_2 = 0.24, \quad \theta_3 = 0.987, \quad \theta_4 = 1.18, \quad \theta_5 = 2.15, \quad \theta_j = 9, \forall j \in \{6, ..., 10\},$$

Figure 4.5: Triggering parameters applied in the test case.

4.4 A Cyber-Physical Experimental Testbed

A widely-adopted methodology for evaluating wireless networked control systems relies on small-scale laboratory setups mimicking industrial process control loops, e.g., the double-tank system [47, 139]. This approach tests the ability to control *real* physical processes, but often relies on single-hop networks, neglecting key networking aspects (e.g., packet delays and losses) which WCB instead explicitly addresses.

To overcome this limitation, we designed an experimental setup (Figure 4.6) combining a simulated plant with a real large-scale wireless network. Its architecture is general and can be applied to systems exploring alternate control strategies and/or network stacks supporting them.

Real network, simulated plant. The plant model, implemented in MATLAB/Simulink, emulates the physical system; it receives actuator state changes as input and produces sensor readings as output. We replace the real drivers on the wireless devices with stubs interacting with the plant model, so that *i)* sensor nodes receive values from the model instead of real sensors, and *ii)* actuator nodes send the commands

$\alpha_i \dot{x}_{2i} = u_i(t - \tau_i)$.

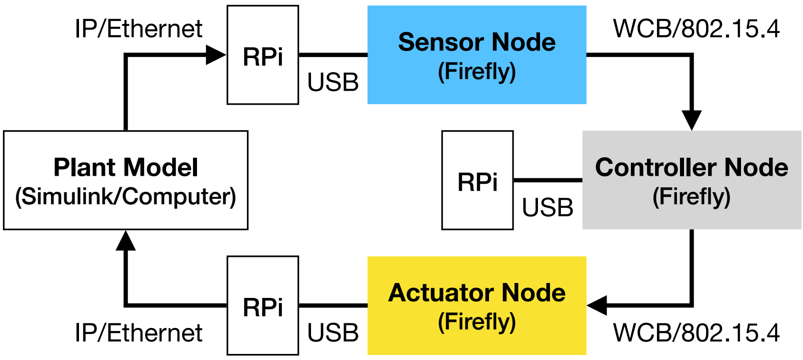


Figure 4.6: Experimental framework: Architecture.

received from the controller to the model instead of the real actuators. Communication between the stubs and the computer running the plant model occurs out-of-band, via TCP/IP over Ethernet, to avoid interfering with the wireless network under study. The latter runs WCB unmodified, providing multi-hop communication among sensor, actuator, and controller nodes distributed across large testbed areas. Each network node consists of a Zolertia Firefly [57], the actual embedded platform under test, connected via USB to a Raspberry Pi (RPi). The Firefly is equipped with a TI CC2538 SoC combining an ARM Cortex-M3 MCU and a 2.4GHz IEEE 802.15.4 radio. Our WCB prototype is built atop a Contiki OS port of Glossy for this SoC [144]. The RPi supports the above out-of-band channel between the Firefly board and the plant model, as well as enables the automation and remote execution of tests.

Dealing with time. For our setup to provide a realistic evaluation, it is crucial that the plant simulator, controller, and wireless network share the same notion of time. The main challenge is to realign the physical time the last two physical components rely on with the synthetic one in the plant simulator. Moreover, the out-of-band Ethernet bridging the real and simulated components is affected by random delays not present in a real system, which must be accounted for.

We address these issues as follows. First, we observe that, thanks to the synchronization inherent in WCB and other Glossy-based protocols, all wireless nodes, notably including the controller, share the same time reference with ms-level accuracy. Therefore, they can timestamp local events and perform their actions at specified instants in *global* time. Second, the joint operation of control and network is *periodic* and *structured*: *i)* (short) active periods where communication occurs are interleaved with (long) periods where the system is quiescent, and *ii)* during active periods, the interleaving of communication and control follows a well-defined pattern known a priori. Third, we leverage the presence of a simulated component to realign the physical and synthetic time references, precisely by exploiting the periodic and structured system nature. During the inactive portion of the schedule, the simulator runs at its own (faster) pace, generating the inputs to be fed to physical components at appropriate (global) times.

Figure 4.7 illustrates our strategy. Sensor acquisition during epoch E occurs at its start time, ${t_S}^E$. The WCB collection schedule unfolds and, after the recovery phase, the controller executes and generates the actuation commands. These are sent during the WCB dissemination phase, and received by each actuator $i \in \{1, \dots, M\}$ at a potentially different time $t^E_{A,i}$. Once dissemination is complete, the WCB network enters sleep. During this inactive period, the actuator stubs send the received commands to the plant model over the out-of-band network, along with the reception times $t^E_{A,i}$ that, like ${t_S}^E$, are precisely timestamped, as per our first observation. These actuator states are collected at the computer running the plant model and input to Simulink, which executes the block diagram shown in Figure 4.8 with a simulation time synchronized with the epoch start, ${t_S}^E$. The timestamps $t^E_{A,i}$ are used to "replay" the arrival of the actuation commands u_i by taking into account the *real* delays $\Delta^E_{A,i} := t^E_{A,i} - t^E_S$. Based on this input vector $u_i(t + \Delta^E_{A,i})$, $i \in \{1, ..., M\}$, the simulator advances the model execution in the time interval $[t^E_S; t^{E+1}_S]$, generating the sensor readings for the acquisition at the beginning of the next epoch. These are sent to the stubs on the sensor nodes via the out-of-band network; when the (physical) time t^{E+1}_S arrives, the sensor nodes wake up and "acquire" these sensor readings. The process repeats in each epoch.

Nevertheless, the inactive period of the wireless network must accommodate the worst-case delays induced by model computation and Ethernet communication. Although we designed our testbed to stop upon detecting a violation of this requirement, this never happened in our experiments, where delays (<2 s) are significantly smaller than the control period (60 s). In cases where the control period is shorter than the delays, execution can be artificially slowed down by increasing the inactive period and removing the extra *empty* time in post processing. The opposite, i.e., shortening the inactive period and adding empty time in post processing, can also be done; we actually adopted this technique to speed up the execution of our experiments.

Wireless testbeds. We rely on two large-scale multi-hop wireless testbeds at our premises, called DEPT and HALL, constituted by 36 and 19 nodes, respectively. DEPT (Figure 4.9a) is deployed along office corridors, yielding a mostly linear topology

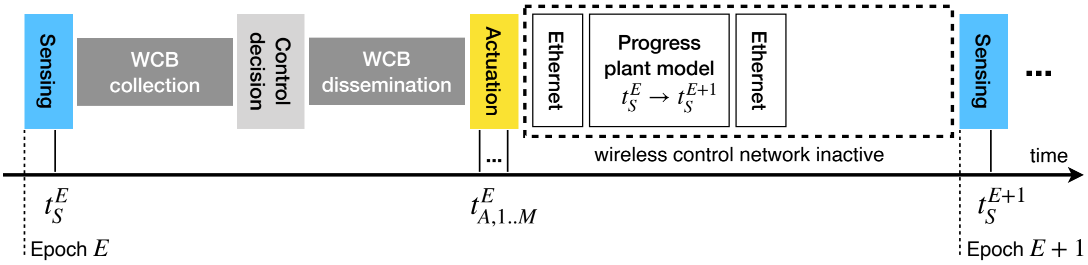


Figure 4.7: Synchronous test execution with real wireless network and simulated plant.

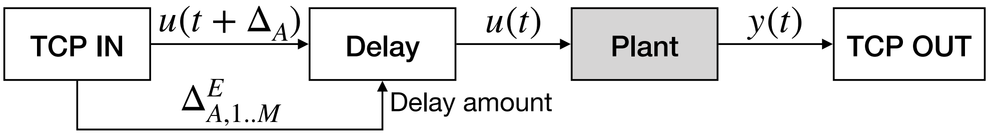
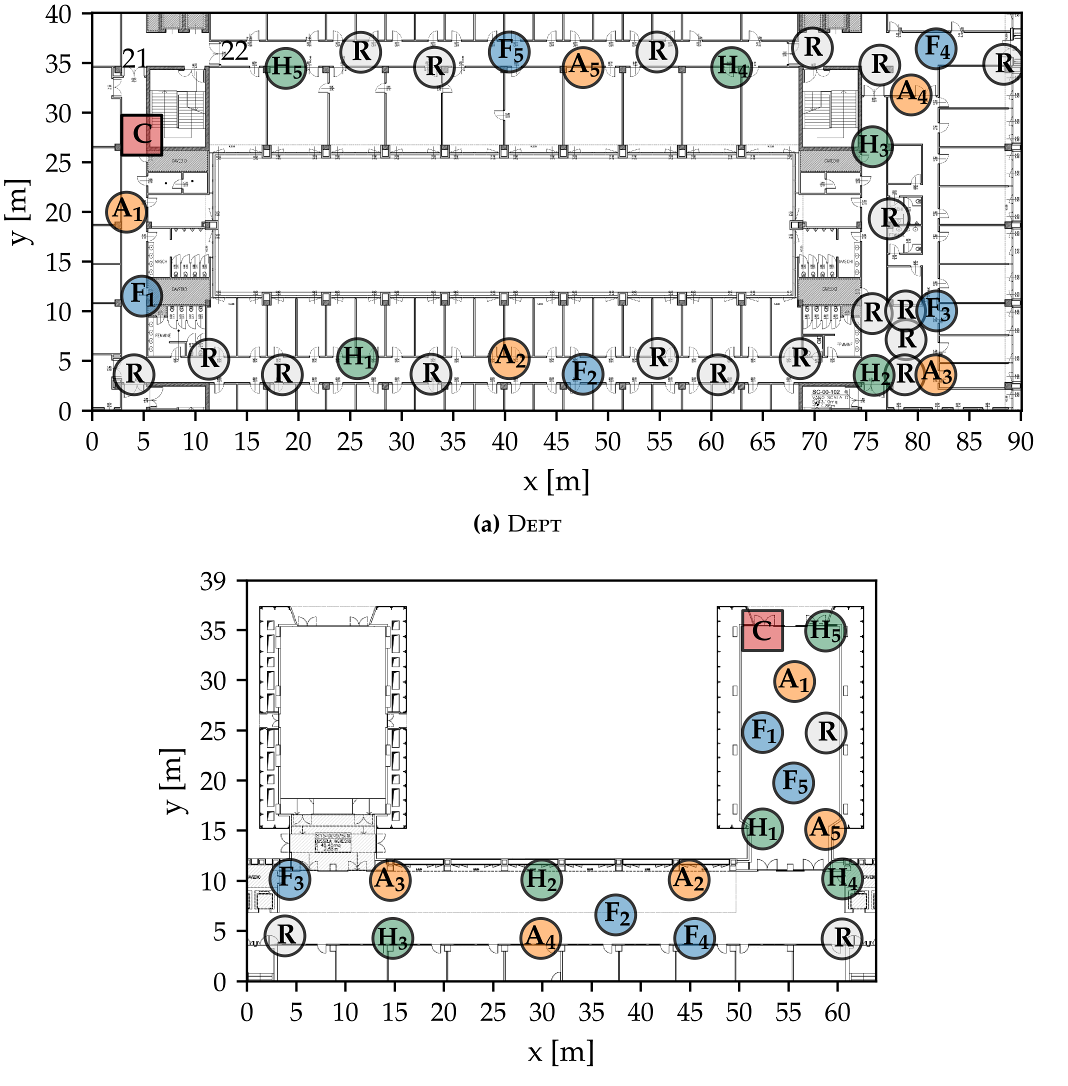


Figure 4.8: Block diagram of the simulation.



(a) Dept

(b) Hall

Figure 4.9: The wireless testbeds used in our experiments. The red square denotes the controller (C), orange circles are actuators (A), while light blue and green circles are flow (F) and height (H) sensors, respectively. Nodes acting as forwarders (R) are in grey. Nodes 21–22 (top figure) are disabled to increase the network diameter.

spanning a 83 × 33 m^2 area; by disabling node 21–22 we enforce a 5-hop network. Hall (Figure 4.9b) is denser and spans a 56 × 30 m^2 L-shaped area; nodes in the same segment are within communication range, yielding a 2-hop network.

The role of each node (Figure 4.9) mimics our WIS test case (Figure 4.3): the actuator and flow sensor nodes of pool i are close to each another, while the height sensor is far from them, at the end of pool i and closer to the actuator and flow sensor of pool $i+1$. Instead, the controller node position maximizes hop distance, creating a challenging topology for our evaluation.

Benefits and applicability. Our experimental setup is a contribution per se, offering several advantages. It is *flexible*, enabling experimentation with control systems exhibiting diverse requirements and time scales by simply developing appropriate

Table 4.2: Protocol parameters.

	Slot parameters, defined for every slot type
W	Slot duration
N	Number of packet retransmissions within the slot
	Epoch parameters
R	Max. number of TA pairs in the recovery phase
C	Number of command dissemination slots CTRL
E	Number of event slots EV (only WCB-E)

Table 4.3: Reliability of the WCB configuration.

	Hall			Dept		
Slot Type	N	W	$\overline{PDR}$	N	W	$\overline{PDR}$
S	3	7	0.99996	3	10	0.99993
T	2	6	0.9994	2	9	0.99914
A	3	8	1.0	3	11	0.99994
CTRL	2	8	0.99987	2	11	0.9998

Simulink models. It is easily *replicable* and *scalable* as it does not require specific hardware components apart from mote-class and RPi-class devices; existing wireless testbeds [41, 106, 115] could easily support it. Finally, and most importantly, it fosters *repeatability*, as the control plant is simulated, hence not subject to the vagaries of a real system.

4.5 Configuring (and Improving) the Wireless Control Bus

We empirically study how the parameters of WCB affect its performance, and identify the configuration used in the evaluation. This is also an opportunity to identify low-level optimizations further improving performance. Table 4.2 summarizes the key parameters, following the protocol description (§4.2). *Slot* parameters govern the behavior of a single Glossy flood, and can be tuned for each slot type. *Epoch* parameters govern the use of these slots inside the active period in each epoch. The table does not consider the number K of data collection slots T, one per sensor node, as this is an *application* parameter and therefore only known at deployment time.

Methodology. We determine the parameter values as inspired by [31]. We analyze the sensitivity of WCB to each parameter value via thousands of floods performed with the same topology, initiating nodes and packet size as in our evaluation (§4.6). An exception is the duration W_x of each slot type $x \in \{\mathsf{S}, \mathsf{T}, \mathsf{A}, \mathsf{EV}, \mathsf{CTRL}\}$, determined analytically based on the corresponding number N_x of packet retransmissions and knowledge of network diameter, packet on-air duration, and Glossy delay between packet RX and TX, plus a small slack accounting for potential collisions.

Slot parameters. Table 4.3 shows the configuration we select along with the corresponding mean packet delivery rate $\overline{PDR}$ for the *whole* network, which we prefer to consider here w.r.t. the *PDR* of sink and actuators to reduce the risk of biasing results due to lucky or unlucky choices of these nodes. EV slots are not reported here as they are used only in WCB-E; they are analyzed at the end of the section.

A value $N \in \{2, 3\}$ ensures very good reliability; higher values increase consumption without much improvement. We select $N = 3$ for S and A slots as these are *i)* crucial to the overall reliability of WCB, and *ii)* scheduled *once* per epoch, bearing a moderate impact on energy consumption w.r.t. $N = 2$. As for T slots, they *i)* are the largest—and thus more energy demanding—component of an epoch active portion, always present in WCB-P and dynamically triggered in WCB-E, and *ii)* benefit from the safety net of acknowledgements and retransmissions scheduled on-demand during the recovery phase. Therefore, we privilege energy consumption over reliability and use $N = 2$. Nevertheless, Table 4.3 shows that this value still achieves a remarkable three-nine reliability of T slots *over the entire network*.

Knowledge of this reliability enables us to estimate analytically the probability to collect at the sink *all* the K sensor readings, assuming packet loss modeled as a series of independent and identically distributed (i.i.d.) Bernoulli trials [88]. In our case (§4.3), this yields a probability to deliver all $K = 10$ sensor readings of 99.3% and 99.6% in HALL and DEPT, respectively. In other words, at least one reading is lost only in 4–7 epochs out of 1000. In these relatively rare cases, the recovery phase is automatically triggered, and the lost packets retrieved when needed, much more efficiently than by increasing the reliability (and consumption) of *every* T flood.

Epoch parameters. In the recovery phase, R is the number of TA pairs enabling nodes to retransmit packets not acknowledged by the sink, if any. This parameter directly affects the reliability of data collection but also the latency of actuation commands, as their dissemination is always scheduled after the maximum duration of the recovery phase (Figure 4.1). Hereafter, we select $R = 3$, as we verified experimentally that the probability to lose more than three packets in the collection phase is below 10^{-7}.

On the other hand, the dissemination phase must also be reliable in addition to timely, as it is crucial to the control operation that actuation commands are correctly received network-wide. Nevertheless, a safety net of acknowledgments and retransmissions, akin to the one supporting many-to-one data collection traffic, would be inefficient for one-to-many dissemination. Fortunately, a simple and effective redundancy strategy where the CTRL slot containing actuation commands is always repeated C times is possible. Table 4.3 shows that $N = 2$ already makes it unlikely that an actuation message is lost network-wide. The probability that the packet is lost *multiple* times in a row is therefore very low; we verified empirically and analytically that the value $C = 2$ used hereafter is sufficient to obtain between 6- and 7-nine reliability in our testbeds.

Event phase. The reliability of the event phase in WCB-E is crucial to the correct and timely operation of ETC. Nevertheless, the EV slots constituting this phase have peculiar characteristics. First, they are *shared*; several sensor nodes may detect at the same time a violation of the triggering condition and decide to signal an event by concurrently transmitting in the same EV slot. Second, their reception triggers a reaction *at the sink and all sensor nodes*, signaling the need to perform a collection phase. Third, as in the case of actuation commands, this traffic pattern is not amenable to acknowledgments, and therefore must rely on alternative reliability mechanisms.

Table 4.4 analyzes the reliability of EV slots, similarly to what reported for the other slots in Table 4.3, this time considering also a number U of randomly-selected sensor nodes transmitting in the same shared slot. Results show that while most of the network, including the sink, enjoys near-perfect reliability, a few nodes instead experience repeated losses. This is exacerbated as U increases, with a minimum network-wide $\overline{PDR} = 97\%$. Unfortunately, losing 3 events out of 100 is unacceptable, as it could severely hamper ETC performance.

A redundant strategy, similar to the one adopted for the dissemination phase, mitigates the problem; repeating the EV slot for $\mathsf{E} = 2$ times improves reliability in all configurations and yields a minimum $\overline{PDR} = 99.3\%$. Increasing E would improve further, but also severely reduce the energy efficiency of the ETC system, as the event phase is scheduled in *every* epoch of WCB-E.

However, an alternative, energy-efficient technique is possible. We observe that event packets do not carry data; their mere reception is what informs nodes that an event has been reported. Consequently, instead of requiring correct reception of event packets, we consider the reception of *any* IEEE 802.15.4 frame (even corrupted ones) in an EV slot as an indication of an event detection.

The impact of this technique is beneficial, as shown in the right-hand side of Table 4.4, reporting the average, network-wide signal detection rate $\overline{SDR}$. Reliability is increased in all configurations, with a minimum $\overline{SDR} = 99.8\%$ with $U = 10$ senders in Dept. Further, reliability rapidly increases as U decreases, achieving or approaching 5 nines. In practice, in our representative test case the number of sensors concurrently detecting events is below 1.2 on average, and always lower than 6; the configuration we chose provides therefore very-high dependability for the WIS test case under study.

On the other hand, relying on corrupted packets in the EV slot may lead nodes to falsely presume an event has been detected, wasting energy by incorrectly triggering data collection. We verified empirically both in our dedicated experiments as well as in the overall evaluation (§4.6) that the rate of these false positives is below 0.003%, bearing a negligible impact on energy consumption.

Table 4.5 summarizes the configuration used in the evaluation.

4.6 ETC over WCB: A Testbed Evaluation

We now ascertain the ability of WCB to efficiently support ETC by fulfilling its peculiar requirements in terms of reliability and latency, necessary to a correct and efficient control, while retaining the energy savings enabled by ETC adaptive sampling. To offer a concrete and complete application of ETC over WCB, we focus on the WIS test case and execute in our cyber-physical testbed (§4.4) the control strategy we out-

Table 4.4: Reliability of the EV phase in WCB-E.

				$\overline{PDR}$		$\overline{SDR}$	
	N	W	U	E=1	E=2	E=1	E=2
Hall:							
	2	4	1	0.9993	0.9999990	1.0	1.0
	2	4	2	0.992	0.99973	0.9986	0.99999
	2	4	3	0.985	0.9988	0.997	0.99994
	2	4	5	0.973	0.995	0.991	0.9995
	2	4	7	0.969	0.993	0.988	0.999
	2	4	10	0.976	0.997	0.989	0.999
Dept:							
	2	6	1	0.9988	0.999997	1.0	1.0
	2	6	2	0.996	0.99996	0.9994	0.999997
	2	6	3	0.993	0.99986	0.9988	0.999993
	2	6	5	0.991	0.9997	0.9984	0.99998
	2	6	7	0.987	0.9991	0.997	0.9998
	2	6	10	0.97	0.995	0.989	0.998

Table 4.5: WCB configuration in §4.6. The values W_x are in ms.

	N_S	W_S	N_{EV}	W_{EV}	N_T	W_T	N_A	W_A	N_{CTRL}	W_{CTRL}	E	R	C
Hall	3	7	2	4	2	6	3	8	2	8	2	3	2
Dept	3	10	2	6	2	9	3	11	2	11	2	3	2

lined (§4.3) atop the WCB-E variant properly configured (§4.5). Each experiment has a duration of one full day (1440 epochs) of simulated time, repeated multiple times.

We compare against periodic control over WCB-P. Although a comparison of the latter against the state-of-the-art in networking for periodic control is outside the scope of this chapter, we argue that WCB-P is likely more performant than the existing CTX-based solutions we survey in §4.7—themselves outperforming conventional ones—due to the different design and reliability mechanisms, whose beneficial impact we show here. In any case, given that WCB-P is essentially a degenerate case of WCB-E (§4.2.2) our choice compares both control strategies against the same protocol framework, elucidating the key differences without the bias a completely different network stack would induce.

4.6.1 Control Performance

Each simulated day starts off the setpoint, with $x_{1,i} = 0.05$ m, $x_{2,i} = x_{3,i} = 0$ m for each pool i and no disturbance. Off-take step disturbances are added at pool 5 as in [140]: 0→16 m^3/min at minute 180, 16→34 m^3/min at 450, and 34→0 m^3/min at 600. As the system has time to settle in between and after disturbances, we observe it both in steady state and during transient, when perturbed.

We consider *i)* an ideal scenario where sensors yield perfect readings, and *ii)* one where independent normally-distributed pseudo-random white noise is added to both level and flow measurements, with zero mean and standard deviation of 0.001 m and 1 m^3/min, respectively. In the ideal scenario, the only source of randomness is the network, allowing us to isolate the impact of the protocol stack on control performance. In the second scenario, the added noise introduces variability (and degradation) of the ETC sampling performance, enabling a more realistic assessment.

Metrics. We focus on the number of samples generated as well as on two metrics based on the *integral average error* (IAE) of a signal $x(t)$ w.r.t. its reference x^*

$$\mathrm{IAE}(x, x^*, T) := \frac{1}{T}\int_0^T |x(t) - x^*| \mathrm{d}t. \tag{4.16}$$

This standard control performance metric measures the accumulated tracking error; the smaller its value, the faster states converge to their references. In our case $T = 1440$ minutes, the duration of the experiments. Since height references are already accounted for in the variables $x_{1,i}$, we set $x^* = 0$, yielding the metrics $\mathrm{IAE}_i := \mathrm{IAE}(x_{1,i}, 0, T)$. For each simulation, we compute the sums and maxima of IAEs over the pools, with the following shortened notations:

$$\mathrm{IAE}_{\Sigma} := \sum_{i=1}^{5} \mathrm{IAE}_i, \quad \mathrm{IAE}_{\max} := \max_{i\in\{1,\dots,5\}} \mathrm{IAE}_i, \tag{4.17}$$

Results. The pool heights follow a similar trajectory under both control strategies (Figure 4.10, top) and with a similar performance in reference tracking (Table 4.6), confirming the desirable property that ETC yields essentially the same control output of periodic control. However, *ETC generates significantly fewer samples than periodic control*, almost 90% less in the ideal scenario and only slightly more, 87% less on average, with measurement noise (Table 4.6). The sample pattern for ETC (Figure 4.10, bottom) highlights that, as expected, sampling is more frequent when transients are stronger, and becomes sporadic as the system approaches steady state.

We observe that the *variation* of sample count across experiments, captured by the standard deviation (Table 4.6), appears in ETC only in the scenario with measurement noise and is completely absent in the ideal one. This is a witness of the consistent performance of WCB-E in terms of reliability and latency, analyzed next: *practical control aspects like measurement noise induce significantly higher variations in ETC sampling than the vagaries of the wireless communication.*

4.6.2 Network Performance

The reliability of event detection, sensor reading collection, and command dissemination, together with the actuation latency, are crucial to the control performance we observed.

Table 4.7 reports the average of these metrics across 16 test runs of one full day of simulated time each, i.e., 1440 × 16 = 23040 epochs in each row. WCB achieves *zero* packet losses regardless of the functionality, protocol variant, and testbed considered,

Table 4.6: Sampling and control performance metrics from experiments: mean (and standard deviation when different from 0) over 8 executions of 1 day of plant operations each.

Scenario	Testbed	Sampling	Sample count	$\mathrm{IAE}_{\sum}$ (m)	$\mathrm{IAE}_{\max}$ (m)
No noise:					
	Hall	ETC	149	0.1084	0.03283
		Periodic	1440	0.1085 ($<10^{-6}$)	0.03293 ($<10^{-6}$)
	Dept	ETC	148	0.1088 ($<10^{-6}$)	0.03286
		Periodic	1440	0.1085 ($<10^{-6}$)	0.03293 ($<10^{-6}$)
With noise:					
	Hall	ETC	186.1 (5.743)	0.1091 (1.21×10^{-4})	0.03311 (6.1×10^{-5})
		Periodic	1440	0.1088 (3.8×10^{-5})	0.033 (2.1×10^{-5})
	Dept	ETC	185.4 (4.984)	0.109 (1.39×10^{-4})	0.03308 (4.7×10^{-5})
		Periodic	1440	0.1088 (3.8×10^{-5})	0.033 (2.1×10^{-5})

confirming the effectiveness of its strategy (§4.2) and configuration (§4.5). Together with the high dependability of the Glossy flooding primitive WCB builds atop, recovery mechanisms are key to achieve this result. Log inspection shows that, for data collection, they are triggered ~1% of the times; while small in absolute terms, this fraction of lost packets, if not recovered, would be enough to impact negatively the control performance.

The latency between the beginning of an epoch and the delivery of the *last* actuation command is also very small, especially if compared to the sampling period (hundreds of ms vs. 60 s). Further, it has minimal jitter, as commands usually reach actuators in the first CTRL slot. Interestingly, the different network diameter of the two testbeds induces an inevitable difference in the latency of actuation commands. Although this

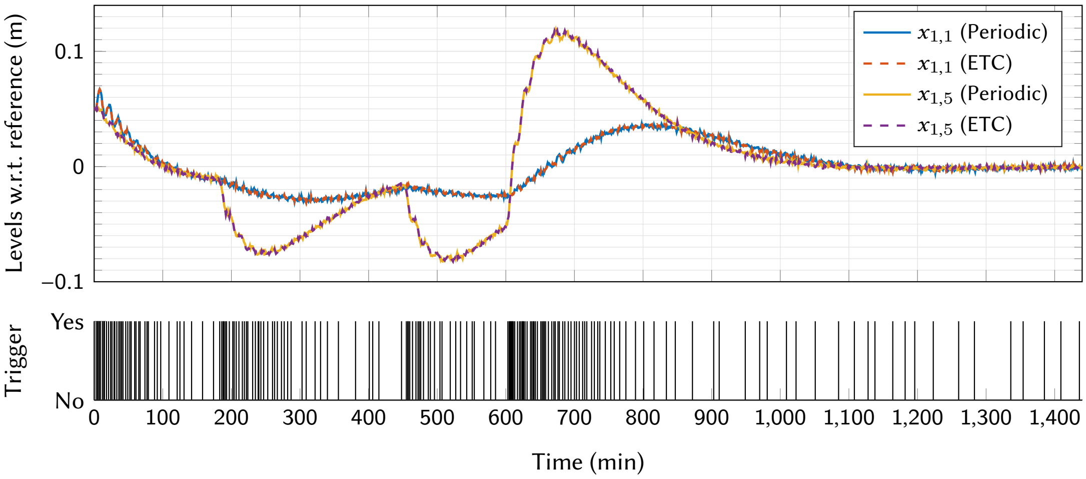


Figure 4.10: ETC vs. periodic control, both over WCB in Dept. Top: Level w.r.t. reference for the 1st and 5th pools, 1-day executions with measurement noise. Bottom: Sampling instants for the ETC case.

Table 4.7: Performance of WCB in hardware-in-the-loop testbed experiments: mean (and standard deviation when non-zero) over 16 executions of 1440 epochs each, i.e., 1 day of plant operation.

		Reliability [%]			**Latency [ms]**
Testbed	Protocol	Event detection	Data collection	Command dissemination	Actuation commands
Hall	WCB-E	100	100	100	192.021 (0.04)
	WCB-P	—	100	100	180.023 (0.02)
Dept	WCB-E	100	100	100	253
	WCB-P	—	100	100	237.017 (0.012)

difference is very small (<61 ms) w.r.t. the system dynamics (hours), ETC is known to be sensitive to small perturbations over the long run; the net effect is however only a small difference in the ETC sample count (Table 4.6).

Finally, as expected, WCB-E is slightly slower (∼6.7%) than WCB-P due to the additional event detection phase; nonetheless, the absolute difference is negligible w.r.t. the sampling period and does not affect the control output (Figure 4.10, Table 4.6), as already discussed in §4.6.1.

4.6.3 Energy Consumption

The wireless transceiver is notoriously the most power-hungry component in networked embedded systems, and the one whose contribution ETC seeks to minimize. Therefore, we compare ETC vs. periodic control in terms of the radio duty-cycle $DC = \frac{T_{on}}{T}$, i.e., the per-node radio-on time over experiment duration, a metric commonly accepted as a reliable proxy for energy consumption.

Key finding. Table 4.8 confirms that our embodiment of ETC consumes significantly less than periodic control—one of our goals. The reason lies precisely in the *interplay* between ETC and the network stack supporting its operation, WCB-E. By design, ETC abates traffic by triggering sensor data transmissions only when needed for control. In our test case, more than 89% of the periodic samples are suppressed in the ideal case, and more than 87% in the noisy one. In general, this traffic suppression does not automatically translate in energy savings, e.g., as reported in [139], where due to an overly conservative design of the networking stack, a tenfold reduction of control communication does not even result in a double life span of the aperiodic system under study w.r.t. a periodic counterpart. Nevertheless, WCB-E minimizes consumption when the system is in steady state while ensuring timely and reliable communication when required to support control. In our case, this yields a DC reduction well above 62%, with marginal differences in the two testbeds due to their different network diameter. Therefore, WCB-E *effectively translates the significant reduction of control traffic achieved by ETC into corresponding savings in energy consumption*. This is a significant leap forward w.r.t. state-of-the-art ETC literature [43, 44, 45, 127, 128] whose energy reduction is hampered by inefficient protocols and limited to

Table 4.8: Sampling and duty-cycle performance of ETC and periodic control vs. presence of measurement noise. Results are average percentages over 8 executions of 1440 epochs each, i.e., 1 day of plant operation.

		No measurement noise			**With measurement noise**		
			ETC reduction [%]			ETC reduction [%]	
Testbed	Control	*DC* [%]	Sampling	*DC*	*DC* [%]	Sampling	*DC*
HALL	ETC Periodic	0.0319 0.0992	89.65	67.84	0.0341 0.0987	87.02	65.45
DEPT	ETC Periodic	0.0413 0.1166	89.72	64.58	0.0438 0.1167	87.13	62.47

Table 4.9: Average per-epoch radio-on time T_{on} and duty-cycle *DC* without measurement noise. Values are the average over 8 executions of 1440 epochs each, i.e., 1 day of plant operation.

		WCB-E					**WCB-P**
Testbed	Metric	No event detected	Event detected	Transient (600–750)	Steady state (1000–1440)	1 day (0–1440)	1 day (0–1440)
HALL	T_{on} [ms] *DC* [%]	13.81 0.0230	65.51 0.1092	29.58 0.0493	14.40 0.0240	19.16 0.0319	59.50 0.0992
DEPT	T_{on} [ms] *DC* [%]	18.82 0.0314	76.93 0.1282	36.58 0.0610	19.60 0.0327	24.79 0.0413	69.98 0.1166

small-scale star topologies.

Dissecting the energy contribution. Figure 4.11 highlights where energy savings arise from, by comparing the average *DC* per epoch of WCB-P and WCB-E across one day of plant operation. The behavior of the periodic controller is invariant w.r.t. system conditions. Therefore, WCB-P must acquire sensors readings and disseminate actuation commands in *every* epoch, resulting in a nearly-constant *DC*; the small spikes correspond to occasional recovery phases. In contrast, the adaptive ETC controller triggers communication via WCB-E only when needed. This results in a pattern similar to Figure 4.10, although here we focus on the ideal case as it simplifies

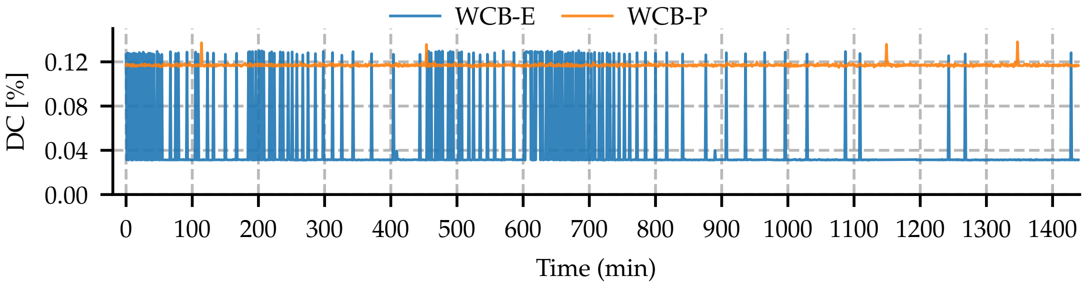
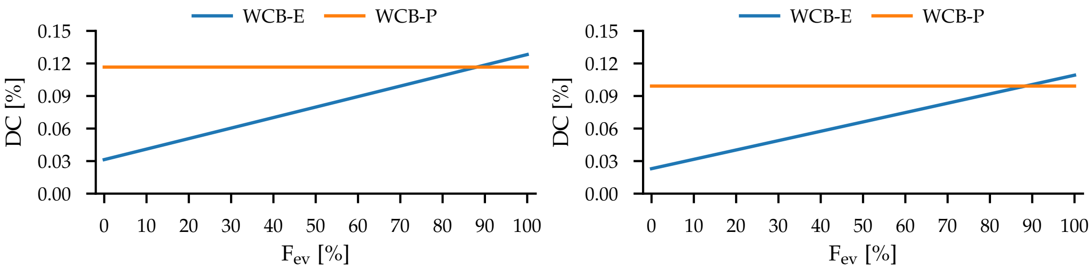


Figure 4.11: Comparison of the average network duty-cycle per-epoch of WCB-E and WCB-P during one day of plant operations in DEPT in absence of measurement noise.



(a) Dept **(b)** Hall

Figure 4.12: Comparison of WCB-P and WCB-E vs. the frequency of epochs with events, in both testbeds.

observations concerned with communication by separating them from measurement noise. After the initial settling phase, Figure 4.11 clearly shows how *DC* increases in conjunction with off-take step disturbances (minute 180, 450, and 600) and reduces when the system approaches stability (1000–1440).

Table 4.9 offers additional insights on T_{on} and *DC*, by comparing the invariant control operation of WCB-P against the various stages of ETC operation over WCB-E. It provides various aggregates: for all epochs with and without events, for the "steady" state (1000-1440 minutes, only 7 epochs with events), for the "disturbed" state (600-750 minutes, 32% epochs with events), and for the whole day of operation. In epochs where no event is detected WCB-E saves 73.1% and 76.8% w.r.t. WCB-P in Dept and Hall, respectively. Energy is minimized by putting the network to sleep right after the EV phase (Figure 4.1). Otherwise, when an event is detected WCB-E is slightly more active ($\leq$ 10.1%) due to the extra EV slots.

Generalizing to other scenarios. These results show how the efficiency of ETC over WCB-E ultimately depends on how frequently the triggering condition is violated. As long as events are relatively *rare*, the energy savings in steady-state outweigh the extra cost of the EV phase.

System designers must ascertain this tradeoff in the early stages of development, to select the most appropriate control strategy and the corresponding network stack supporting it. Luckily, analytical models for the energy consumption of both WCB variants can be easily derived, as all nodes follow the same, global, periodic schedule. Once the average network-wide radio-on time $t_{on,X}$ of each slot type is estimated as in [31] and §4.5, the overall per-epoch radio-on time $T_{on,P}$ of WCB-P is simply the sum of $t_{on,X}$ across slots in each protocol phase, invariant w.r.t. event detection. The one for WCB-E is then derived as:

$$T_{on,E} = F_{ev} \times (T_{on,P} + \mathsf{E} \times t_{on,EV}) + (1 - F_{ev}) \times (t_{on,S} + \mathsf{E} \times t_{on,EV}) \tag{4.18}$$

where F_{ev} is the average frequency of epochs with at least one event and E the number of EV slots (§4.5). *DC* is computed for both cases by dividing the radio-on time by the epoch duration.

Figure 4.12 exemplifies the tradeoffs at stake by reusing the parameters from our evaluation except for the frequency F_{ev}, whose value here is varied to represent, in an abstract setting, the DC resulting from several hypothetical control problems. The charts show how, in these conditions, periodic control over WCB-P becomes preferable vs. ETC over WCB-E only when $F_{ev} \geq 90\%$; notably, the latter enables conspicuous energy savings even when F_{ev} approaches this break-even point. For instance, when $F_{ev} \approx 70\%$, DC is reduced by nearly 15%, which becomes 25% when $F_{ev} \approx 60\%$, still extending system lifetime significantly. Overall, this confirms that ETC over WCB-E supports a wide range of real-world control problems and systems where it unlocks remarkable energy savings, ultimately pushing the envelope of the application of cyber-physical systems to untethered scenarios.

4.7 Related Work

The adaptive control strategy of ETC raised a lot of interest in the last decade, with several researchers tackling the design of new triggering conditions and other strategies to reduce communication further [127, 128], improve applicability on digital platforms [45], and decentralize triggering conditions [44]. An overview of the state of the art in ETC can be found in [40, 46].

However, the benefits unleashed *in theory* by ETC must be confirmed *in practice* by real-world testbed experiments. This is true in general [24, 145] and, as discussed at the beginning of this chapter, even more poignant for ETC, given the peculiar challenges it poses to communication [19]. Unfortunately, only few works investigate ETC performance via prototypes. These use IEEE 802.15.4 [47, 48], WiFi [129], or G5 (IEEE 801.11.p) [131], but always in a single-hop topology with at most 5 nodes, hardly representative of staple real-world use cases for WNCS.

In contrast, the work described here is validated with a realistic setup that combines a model of the system under control with a real, multi-hop low-power wireless network, yielding a significant higher realism of the evaluation. These testbeds are unfortunately rare in the literature. The closest is the one proposed in [91], featuring a similar combination of modeled system and real network. Nevertheless, the concise description does not detail if and how network-induced random delays are mitigated; further, it relies on the PTP protocol for time synchronization, requiring dedicated, expensive hardware. In contrast, our testbed explicitly targets random delays with an architecture (§4.4) that, in addition, provides the extra flexibility to speed up or slow down the real-time execution. Moreover, it uses commonplace devices and is therefore easily replicable by other researchers.

Apart from providing a realistic evaluation, in this chapter we have tackled the crux of the matter by proposing a network stack expressly targeting ETC. Indeed, those commonly used in industrial control, e.g., WirelessHART [6], ISA100.11.a [5], 6TiSCH [7], do not offer the necessary guarantees [146], especially in multi-hop configurations, and lack the degree of flexibility and adaptability pivotal to efficiently support ETC, motivating this work and specifically the use of CTX.

In this respect, the design of WCB is inspired by two systems: the Low-power Wireless Bus (LWB) [30] and Crystal [31]. The former was the first to make explicit the potential of CTX for abstracting communication into a network-wide bus, generating several follow-up variants. For instance, Blink [146] targets hard real-time communications by equipping LWB with a real-time scheduler based on earliest deadline first. eLWB [147] extends LWB with the ability to handle events, as a side contribution of a more general architecture targeting an acoustic emission monitoring system. In eLWB, the reaction to the event is centralized at the controller, while in WCB it is decentralized at sensor nodes, yielding lower latency. Further, eLWB focuses on monitoring rather than control, without dedicated reliability mechanisms, crucial in ETC and discussed later.

LWB has been exploited also specifically for control. The system in [92] supports feedback control, stability guarantees, and mode changes over multi-hop wireless networks for systems with fast dynamics (tens of ms). Latency is therefore the main focus rather than reliability, for which dedicated mechanisms are not provided. The paper exploits a periodic controller. Another work by the same group explores instead self-triggered control [135] where, contrary to ETC, nodes *predict* when they expect to trigger an event; this information is exploited to reserve the required communication slots with LWB. Self-triggered control is also studied in [91], and compared against rate adaptation; in both control strategies, the necessary communication is provided by a variant of LWB.

The aperiodic, unpredictable communication patterns of ETC are significantly more challenging than the pre-defined or predictable ones induced by periodic and self-triggered control. ETC in principle enables minimal network overhead during quiescent, steady-state periods, but also demands both timely and reliable communication otherwise, to guarantee correctness and performance. None of the LWB-based stacks above support these requirements; further, none of them provides dedicated mechanisms *expressly* targeting a reliability enhancement.

Instead, these conflicting requirements have been reconciled in Crystal [1, 31], notably even under harsh RF noise as discussed and practically demonstrated in the previous chapter. Aperiodic communication "makes each packet count", as it is transmitted unpredictably and sporadically, implicitly carrying more information. Crystal focuses on data collection and exploits the capture effect to support concurrent, reliable transmission of sensor readings, each *individually* acknowledged by a Glossy flood. This pattern directly inspires the T and A slots in WCB, where they are combined differently. In Crystal, concurrent senders are a priori unknown; in the worst case where all U nodes transmit, at least $2U$ Glossy floods are required. In WCB, instead, data collection occurs only if and when an event signaling a violation of the ETC triggering condition is disseminated, and *always* implies the collection of readings from *every* sensor node. As event detection occurs reliably and in a distributed fashion, it eliminates contention and enables the dependable collection of all U measurements using only $U + 1$ floods. The recovery phase, reminiscent of the TA pairs of Crystal, must therefore retrieve only an occasional missed packet, rather than all competing ones, limiting overhead and bounding the recovery duration, crucial for predictable control operation.

4.8 Conclusions and Future Work

We presented the Wireless Control Bus (WCB), the first network stack efficiently supporting the peculiar communication requirements induced by ETC. Unlike the few prototypes reported in the literature, WCB expressly targets multi-hop, low-power wireless networks, and advances the state of the art by significantly reducing the gap between communication savings and energy savings—a well-known issue hampering ETC adoption. We design a centralized state feedback controller using a novel, modified decentralized periodic ETC suited for step disturbance rejection, combine it with WCB, and evaluate its performance in network-in-the-loop setups emulating a 15-state water irrigation system over a real-world multi-hop network. Our results show that w.r.t. periodic control, also implemented over WCB: *i)* ETC reduces samples by more than 87%, translated by WCB into energy savings above 62%, and *ii)* the control performance is essentially equivalent in the two strategies and consistent across experiments, witnessing the extreme dependability of the WCB network layer.

We intend to release publicly WCB as open source. We believe that the availability and performance of WCB, unlocking the full potential of ETC, may fuel new research on this topic. Our own agenda includes exploring the combination of WCB with other decentralized ETC frameworks [45], implementing theta-adaptation [44], and using traffic models [148] to further reduce energy consumption by scheduling longer periods of sensor node sleep, along the lines of [149]. Furthermore, given the potentially higher impact of packet loss for ETC performance w.r.t. periodic control, evaluate WCB under harsh radio-frequency noise—as we did for Crystal in Chapter 3—is also of interest. We already demonstrated that WCB perfectly handles the natural interference present in office buildings; nonetheless, co-design additional noise-resilient countermeasures both at the control and network level could further boost the dependability of our solution, and in turn its widespread adoption.

Concerning our test case of water irrigation systems, we are working on alternate control architectures, like the robust output-feedback controllers in [140], and developing a testbed using a scaled-down irrigation channel, to investigate other practical aspects of wireless ETC.

Finally, as discussed in Part II and Chapter 8, we believe that the exploitation of concurrent transmissions in PHY layers other than IEEE 802.15.4 lays the ground for intriguing research opportunities. Specifically, we are actively working with ultra-wideband (UWB) radios, the reference technology of Part II. Its peculiar features, notably including a data rate more than one order of magnitude higher than IEEE 802.15.4 and a remarkable interference resilience, can unlock additional performance improvements for latency- and noise-sensitive applications like ETC. The contributions presented in Chapter 6 and 7, although not specifically tied to ETC, are a stepping stone towards that direction. First, they practically demonstrate the potential of UWB technology for communication, and the applicability of CTX to this alternate PHY layer. Second, they provide so far missing fast, reliable, and energy efficient UWB network stacks for data dissemination and collection, which can directly foster the co-design of novel low-power wireless ETC systems.

Exploring the synergy of CTX-based UWB protocols with ETC sampling strategies is on top of our research agenda.

Part II

Concurrent Transmissions on Ultra-wideband Radios: Challenges & Opportunities

For a decade and more, IEEE 802.15.4 narrowband was the de facto reference technology for low-power wireless applications. Proposed for the first time in 2003 [150], the IEEE 802.15.4 physical layer rapidly led undisputed the low-power wireless research scenery, up to an extent that relying on IEEE 802.15.4 narrowband radios was somehow an implicit choice for researchers working in the field. The enormous body of literature revolving around the CC2420 radio chip [110] is a witness to this.

Things have rapidly changed in the last few years, with the proliferation of novel technologies, standards, and radio chips revolutionising and broadening the landscape of low-power wireless communications at large, and inspiring a new wave of research interest and enthusiasm in the field. Bluetooth Low energy (BLE), ultra-wideband (UWB), LoRa, SigFox, wake-up radios, and ambient backscattering, are only some of the new radio technologies that are rapidly gaining momentum in academia and industry and are envisioned to (re)shape and lead—together with IEEE 802.15.4 narrowband—the IoT of the future. The very different tradeoffs they offer among energy consumption, data rate, range of communication, and working frequency open up the door for a plethora of new applications that can now enjoy the benefits low-power wireless, and in turn new research opportunities. Among this multiplicity of radio layers, in this second part of the thesis we focus on UWB technology. Specifically, we rely on the DecaWave DW1000 chip [54, 55], arguably the most popular and widely available UWB transceiver today, whose peculiar features and rich set of primitives we exploit to develop fast, reliable, and energy efficient low-power wireless communication protocols.

Why UWB? The reason that led us to focus on UWB is twofold. First, even though still not as pervasive as other radios, UWB is increasingly popular. After a decade of oblivion, the recent availability of a new breed of energy-savvy, cheap, tiny, and standard compliant UWB chips—a breakthrough w.r.t. their bulky, expensive and energy-hungry predecessors, e.g., [151]—has sparked a new research and market interest in this technology. The potential of combining in a single transceiver decimetre-level distance estimation and high-rate communication has rapidly attracted a considerable attention from academia, and an ever-increasing number of prominent players from industry. Bosch, Apple, NXP, Samsung, Xiaomi, Hyundai, Qorvo are only a small subset of the many companies currently working and investing in this technology, and which have recently formalized their common aim to foster the growth of the UWB ecosystem by establishing the UWB Alliance [152] and the FiRa consortium [153]. The inclusion of UWB chips in smartphones by Apple [49] and Samsung [50, 51], and in laptops by Lenovo [154] is an additional, clear indicator of how UWB is rapidly evolving towards a mainstream consumer technology, with the potential to massively enter in our houses and pockets, and become a pillar technology for the Internet of Things. Second, staple network stacks implementations for UWB are surprisingly missing, in stark contrast with the many available for IEEE 802.15.4 narrowband after almost two decades of research on wireless sensor networks (WSN). UWB precise ranging capability has indeed catalysed a great deal of research effort; designing ever more accurate and sophisticated ranging and localization services has been the cornerstone of UWB research [52, 53], somehow pushing to the background the communication angle. This latter is instead precisely the focus of this part of the thesis.

UWB-based communication: A necessity and an opportunity. As a matter of fact, the lack of fast, reliable, and energy efficient UWB networking protocols could rapidly become a show-stopper for the adoption of UWB technology in different fields. For instance, the efficient support of multi-hop communication is of paramount importance to enable infrastructureless large-scale UWB localization systems [155]—one of the key target applications of UWB technology.

Furthermore, the unique physical layer features UWB radios rely on, open up intriguing research opportunities also for sensing and communication, besides ranging and localization [156, 157]. By encoding information via short ($\sim$ 1 ns) pulses, UWB radios spread the transmitted signal power over a very wide bandwidth ($\geq$ 500 MHz), with a very low power spectral density. Large bandwidth, in addition to excellent time resolution that underpins UWB accurate localization and tracking, results in noteworthy multipath immunity and high data throughput—more than one order of magnitude larger than IEEE 802.15.4 narrowband (6.8 Mbps vs. 250 kbps for DW1000 and IEEE 802.15.4 narrowband, respectively), unlocking substantial latency improvements. Low power spectral density, instead, limits co-existence issues, making UWB signals essentially harmless for co-located RF services, while enhancing robustness to jamming attacks. Last but not least, by operating outside the notoriously crowded 2.4 GHz ISM band, UWB technology is not affected by many of the interference sources commonly present in office and industrial environments [158], which highly hinder narrowband transmissions [26, 27], as analysed in Chapter 3. These properties, remarkable per se, make UWB technology appealing for a myriad of applications, notably including industrial wireless control systems, the focus of Part I. Latency and noise-sensitive applications like WCSs could indeed tremendously benefit from the high interference immunity and the drastic reduction of end-to-end latency that UWB communication entails. Nonetheless, to the best of our knowledge, research in this field is still in its early stages; the few existing works (e.g., [156, 158, 159, 160]) mainly evaluated UWB performance in single-hop industrial settings, only scratching the surface of what can be achieved by UWB communication at large.

The development of network stacks capable of unleashing and practically demonstrating the UWB potential is pivotal for bringing these benefits from theory to practice, fostering the adoption of UWB technology for communication and sensing. This is precisely the contribution of Chapter 6 and 7. Motivated by the overwhelming impact concurrent transmissions had on IEEE 802.15.4 narrowband, we *i)* explore the applicability of this technique in combination with UWB radios to enable fast, dependable, and energy efficient multi-hop communication, and *ii)* leverage this synergy to design and develop WEAVER, a novel ultra-fast protocol stack for aperiodic collection over UWB, respectively.

Before delving into such contributions, in the following chapter we provide a concise summary with the necessary background on UWB technology.

5 Ultra-wideband in a Nutshell

Modern ultra-wideband radios are impulse-based (IR-UWB), using carrier-less signals where information is encoded through short pulses ($\sim$ 1 ns). This translates into a bandwidth ($\geq$ 500 MHz) much wider than traditional narrowband solutions, and a reduced power spectral density. As a result, UWB technology offers several advantages over narrowband communication, e.g., *i)* larger channel capacity (up to 27 Mbps according to the IEEE 802.15.4-2011 standard [8]), *ii)* limited interception probability and co-existence issues, *iii)* enhanced penetration performance [161] and multipath immunity, and *iv)* outstanding time resolution. In practice, UWB radios provide unprecedented potentials both for communication and localization purposes, making this technology extremely appealing for several IoT applications, and motivating the recently renewed interest in this technology.

The IEEE 802.15.4 standard defines the structure of a UWB frame, the embedded error correction mechanisms, and the decoding procedure. Here, we briefly introduce the standard UWB physical layer, and we present the most important details on the DecaWave DW1000 chip [54] we rely on in the following chapters.

UWB PHY layer. The UWB frame (Figure 5.1, top) is divided in two parts with different encodings: *i)* the synchronization header (SHR), and *ii)* the modulated portion embedding the data payload.

The SHR, encoded in single pulses, is composed of a preamble and a start-of-frame delimiter (SFD). The former serves for signal detection and synchronization, the latter indicates the beginning of the data portion. The SHR preamble is constructed from standard-defined preamble codes of 31 and 127 elements drawing from a ternary alphabet $\{+1,\ 0, -1\}$ corresponding to a positive, absent, or negative pulse. Preamble codes, interleaved with zeros according to a spreading factor, form preamble symbols. The spreading operation yields a (mean) *pulse repetition frequency* (*PRF*) of nominally 16 MHz and 64 MHz for 31 and 127 elements long preamble codes, respectively. The number of repetitions of a predefined preamble symbol determines the duration of

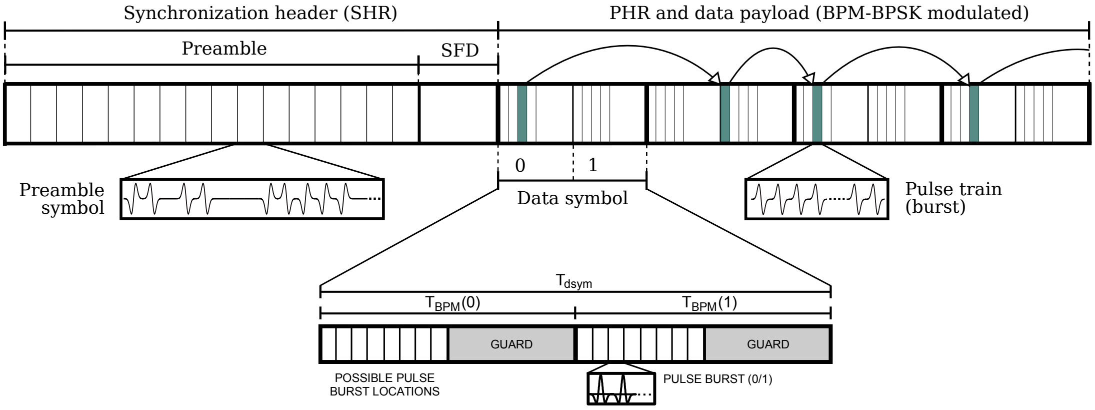


Figure 5.1: The UWB frame structure. Image inspired from [81] and adapted for the purpose of this thesis.

the UWB preamble; similarly to the *PRF*, the *preamble symbol repetition* (*PSR*) is a tunable parameter. The length of the SFD, instead, depends on the data rate of the payload: 64 preamble symbols are recommended when data is sent at 110 kbps, for any other data rate the SFD lasts 8 preamble symbols.

The data portion, consisting of a physical header (PHR) and the data payload, is modulated via a combination of burst position modulation (BPM) and binary phase-shift keying (BPSK). Each BPM-BPSK data symbol (Figure 5.1, bottom) carries two bits of information, but only represents one input bit due to convolution coding. The data symbol, of duration T_{dsym}, is partitioned in two halves of duration T_{BPM}, where only one of the two halves is meant to host a burst (or pulse train). The number of pulses in the burst depends on the configured data rate. For the 6.8 Mbps rate used in the following works, a burst is made of two pulses. If the radio transmits the pulse burst in the first half, it is interpreted as a 0 bit, 1 otherwise. The second bit is encoded by the phase (polarity) of said burst. Guard intervals, in which pulses are never transmitted, are placed between possible burst positions to serve as protection from high-energy multipath signal components, preventing inter-symbol interference. The combination of BPM and BPSK modulation schemes supports both coherent and noncoherent receivers; the latter are unable to extract polarity information but can still decode based on the burst positions. To allow multi-user uncoordinated access, the location of pulses within a T_{BPM} duration is defined by a time-hopping sequence, derived from the preamble code.

DecaWave DW1000 and EVB1000. The DW1000 is a standard-compliant fully-coherent UWB transceiver, now commercialized by Qorvo, and included in the EVB1000 development platform [58] we use in our experiments. The DW1000 supports frequency channels $\{1–5, 7\}$, each with both 16 MHz and 64 MHz *PRF*. Channels $\{4, 7\}$ have a bandwidth of 900 MHz, the others are limited to 499.2 MHz. Three data rates are available: 110 kbps, 850 kbps, and 6.8 Mbps; the latter is our default choice in the next chapters given the benefits high data throughput provides in terms of communication latency. In terms of energy consumption, the DW1000 is considerably less energy hungry than previous UWB transceivers (e.g., [151]); nominal current

Table 5.1: DW1000 current consumption. Note that consumption depends on radio configuration.

Radio state	DW1000
Deep Sleep	50 nA
Sleep	1 µA
Idle	12–18 mA
TX	35–85 mA
RX	57–126 mA

consumption of this chip are summarised in Table 5.1. Unlike narrowband radios (e.g., [110, 162]), the DW1000 presents a significant energy imbalance between TX and RX costs, with the latter operation being substantially more expensive. In the next chapters, we will analyse how this peculiarity impacts UWB protocols' performance.

The DW1000 requires an external 38.4 MHz oscillator, with a tolerance of ±20 ppm [54]. This reference clock is used as phase-locked loop input to obtain a frequency of 125 MHz, allowing packets to be scheduled for delayed transmission with a precision of 8 ns. The programmer can also trim the crystal frequency with a step that depends on the platform capacitors; in our case, the step is $\sim$ 1.45 ppm. Trimming might be used to dynamically tune the node's clock frequency for tighter synchronization, and to enhance the robustness of demodulation, e.g., in presence of concurrent transmissions. The capability to accurately timestamp received frames with sub-nanosecond precision, and schedule TX/RX operations at specified times with a 8 ns granularity drastically simplifies protocols' implementation, as discussed in the next chapters. This is particularly poignant for CTX-based solutions, given the tight timing requirements this technique entails (Chapter 2).

Reception errors. UWB transmissions employ forward error correction in the form of convolution coding, combined with Reed-Solomon (RS) and SECDED bits (single error correction, double error detection) in the data payload and the PHR, respectively. A 6-bit parity check SECDEC filed is included in the PHR, whilst a RS encoder adds 48 parity bits every 330 bits of data. Additionally, a 2-bytes CRC sequence is appended at the end of the frame. Uncorrectable bit errors are reported in the status register of the DW1000. The radio also signals SFD timeouts, which occur if the preamble of a frame is detected, but the presence of the SFD cannot be ascertained within the configured time interval.

Channel codes and error correction mechanisms enhance the dependability of UWB transmissions, e.g., favouring the correct reception of messages despite concurrent transmissions. RX errors, instead, provide useful insights about the nature of packet loss; such notion turned out to be key in guiding us throughout the implementation and debugging of the protocols proposed in the next chapters, and can be leveraged to improve system performance.

6

Concurrent Transmissions for Multi-hop Communication on UWB Radios

Ultra-wideband (UWB) radios are rapidly emerging as a prominent player in the ever-changing landscape of Internet of Things (IoT) enabling technologies. Their reliance on very short impulses yields *i)* distance estimation (ranging) with significantly higher accuracy (centimeters vs. meters) than competing RF-based technologies, and *ii)* high-rate wireless communication, therefore reuniting in a single radio transceiver two key functions of many IoT scenarios.

Motivation. Nevertheless, a staple network stack for UWB is still missing. This is partly explained by the fact that the interest in UWB, at its peak more than a decade ago and largely forgotten thereafter, renewed only recently, fueled by new chips (e.g., the popular DecaWave DW1000 [55]) that yield high ranging accuracy and data throughput and yet are small, cheap, energy-savvy, and standard-compliant. In contrast, during the same decade, research in academia and industry generated numerous IEEE 802.15.4 narrowband protocols, systems, and real-world deployments targeting a variety of traffic patterns, operating conditions, and stack layers. Among these, the approaches based on *concurrent transmissions* (CTX) have proven a very effective building block for protocol design, as discussed in Chapter 2. Pioneered by Glossy [29], several protocols [10, 28] embraced this technique and its variants, ultimately pushing the envelope of IEEE 802.15.4 communication by achieving low latency, high reliability, low energy consumption—all at once.

This chapter revises the publication [2]: D. Lobba, M. Trobinger, D. Vecchia, T. Istomin, and G. P. Picco. "Concurrent Transmissions for Multi-hop Communication on Ultra-wideband Radios". In *Proc. of the* 17th *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2020.

This chapter therefore sets out to investigate whether, and to which extent, CTX are applicable to UWB radios where, if they ensure performance akin to narrowband, this technique could turn into an asset for rapidly bridging the current lack of fast, dependable, and energy efficient UWB network stacks, finally unleashing and concretely showcasing UWB communication potential.

Goals. As we discuss in the context of related work (§6.2), a recent study [81] elicited the conditions for successful UWB concurrent transmissions, both on the same channel (as in our case) and on different channels (not of interest here). Moreover, in the UWB localization system in [133], the use of Glossy-like concurrent transmissions is reported as a means to coordinate ranging exchanges.

In this chapter we exploit some of the findings in these works, but significantly differ from them in its grander goal to:

1. determine whether *different flavors* of UWB concurrent transmissions can be embodied in *full-fledged protocols and systems* and, in the process,
2. highlight *similarities and differences* w.r.t. their narrowband counterpart in terms of both *implementation complexity and system performance*, and ultimately
3. provide a *reference implementation* of CTX protocols that can be *directly* used and improved by the research community at large, fostering the adoption of this technique on UWB radios and, in turn, of UWB technology for communication purposes.

Which type of concurrent transmissions? As mentioned, several "flavors" of concurrent transmissions exist. Glossy was originally designed to support a single network-wide flood, triggered at an *initiator* node; all nodes disseminate the *same* packet via a tightly synchronized schedule of alternating RX and TX slots, until the desired number N of packet (re)transmissions are performed.

In this chapter, we consider other two variants, representative of the state of the art. On one hand, in the last editions of the EWSN Dependability Competition [41] several systems [63, 67] achieved very high performance by changing Glossy to exploit only the single initial RX slot necessary to receive the packet, followed by N consecutive TX slots catering for its re-transmission. Given that the RX energy costs of the popular UWB platform we use are almost twice than TX ones ([54], Chapter 5), this TX-centric operation is definitely worth investigating. On the other hand, various works (e.g., [31, 32]) observed that CTX floods are pretty reliable even when *different* packets are concurrently transmitted by different initiators, offering another dimension to our study.

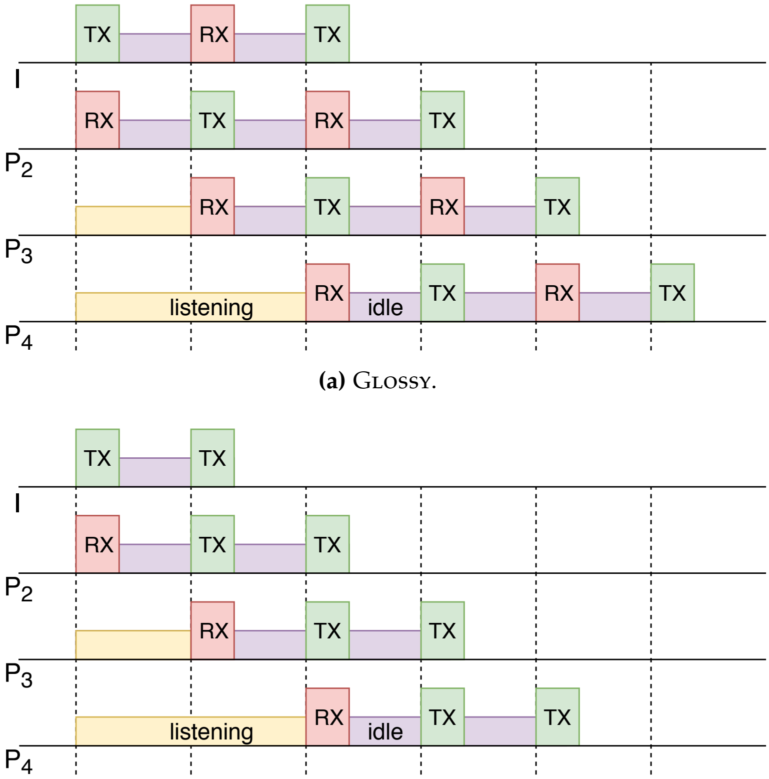
Methodology and contribution. Our investigation is system-driven, and relies on *complete protocol implementations* of the variants above, briefly introduced in §6.1, as well as *testbed experiments* on multi-hop topologies. This methodology is in contrast with [81], whose results rely on micro-benchmarks with few nodes in the same single-hop neighbourhood, and also with [133], whose very short description of their Glossy-like component is insufficient to ascertain its actual performance or guide further developments, which is instead a *desired outcome* of our work. As our target UWB platform we use the popular DW1000 UWB transceiver, and specifically the EVB1000 board; we develop software atop ContikiOS, exploiting the availability of drivers for the DW1000 [163].

This system-driven emphasis enables us to directly face the opportunities and challenges in exploiting UWB concurrent transmissions, as well as to highlight key differences w.r.t. the corresponding narrowband implementations—e.g., due to the ability of the DW1000 chip to precisely schedule transmissions, which greatly simplifies implementation as anticipated in Chapter 5. Further, it also allows us to confirm, and sometimes disprove, some of the findings in [81, 133], ultimately contributing to a better system-level understanding of UWB concurrent transmissions.

We re-implemented Glossy and its TX-based variant from scratch, motivated by key differences in narrowband and UWB radio operation and configuration, detailed in §6.3.1. However, we also used the publicly-available codebase of Crystal [1, 59] that, by combining in a single communication stack both classic, single-initiator, same-packet floods as well as multiple-initiators, different-packet transmissions, serves as a sort of "catch-all" protocol enabling us to experiment with different types of concurrent transmissions in a single, structured system. Further, the fact that we *reuse* the original Crystal codebase allows us to ascertain the extent to which this higher-level protocol built atop a narrowband Glossy layer can work when the latter is replaced with our UWB-based one. Our analysis shows that only minimal changes are required, suggesting that existing Contiki implementations of other higher-level protocols (e.g., [30, 32, 164]) may be similarly reused for UWB radios, with relative ease.

We illustrate the salient details of our implementations of the two Glossy variants (§6.3) and of Crystal (§6.4) hand-in-hand with their evaluation in a 23-node indoor testbed at our premises, which enables us to experiment at scale on multi-hop topologies. Results show that UWB concurrent transmissions yield benefits similar to narrowband, achieving near-perfect reliability and very low energy consumption across the 4 hops in our testbed. Moreover, underpinned by the high clock resolution and data rate provided by the DW1000 radio, CTX on UWB unleash remarkable latency reductions, up to 80% lower than narrowband, along with order-of-magnitude improvements in network-wide time synchronization. Nevertheless, as our experimental results are inevitably biased by the peculiarities of our testbed, we also manipulate artificially our setup to investigate the conditions under which UWB concurrent transmissions may fail, validating or disproving earlier findings [81, 133].

The chapter ends by distilling findings and lessons learned (§6.5) that will hopefully inspire further work on the topic—as they actually did for our next contribution in Chapter 7—before ending with brief concluding remarks (§6.6). We argue that our results pave the way for the exploitation of concurrent transmissions in UWB, which we foster by releasing our systems as open source [60], enabling their immediate use and improvement by researchers and practitioners.



(a) Glossy.

(b) GlossyTX.

Figure 6.1: The two Glossy variants in a 3-hop network with $N = 2$ (re)transmissions.

6.1 Background

We provide a concise primer on the three CTX-based systems focus of this chapter: Glossy, its TX-only variant, and Crystal, analysing similarities and differences in how they leverage and organise CTX floods. We refer interested readers to Chapter 2 for a more comprehensive background on concurrent transmissions; Chapter 5, instead, offers an overview on UWB technology.

Glossy. Originally designed for multi-hop time synchronization, the Glossy protocol ([29], §2.2) embraces non-destructive interference and the capture effect (§2.1) to achieve fast, energy-efficient, and reliable network floods. Figure 6.1a illustrates the concept. The *initiator* begins a flood by broadcasting a packet. As the rest of the network is assumed to be already listening on the channel, the packet is received and immediately rebroadcast by neighbors, yielding concurrent transmissions. After (re)transmitting, the nodes go back to receiving, thus repeating the RX/TX sequence up to N times; the value of N is the main knob to control the balance between flood reliability and energy efficiency. Another important factor affecting energy consumption is the duration of the slots, which must be long enough to accommodate either a packet TX or RX, including some guard times and software delays; nevertheless, when a packet is *not* received, a node listens for the entire slot, potentially wasting energy.

GlossyTX. A Glossy[1] flood unfolds by alternating RX and TX slots (Figure 6.1a); actually, a node is allowed to TX a packet *only* after a successful RX. This choice was originally motivated [29] by the use of CC2420 radio events as a means to enforce tight synchronization. However, it has drawbacks; a node that receives a packet in a RX slot and loses it in the next one is forbidden from rebroadcasting the (*same!*) packet in the subsequent TX slots of the flood until a new successful reception (if any) occurs, wasting time and energy, and possibly decreasing reliability. Glossy partially mitigates these problems by allowing only the initiator—i.e., the synchronization source—to transmit in a TX slot regardless of the outcome of RX ones, improving the flood progress in some unlucky situations.

Figure 6.1b shows an alternative scheme in which each node, after the initial successful RX, performs its N retransmissions in consecutive TX slots. This approach, hereafter called GlossyTX to distinguish it from the original, was first introduced by the winners of the 2^{nd} EWSN Dependability Competition [63], and exploited by other teams in following editions. A major drawback of GlossyTX is that its implementation, relying solely on timeouts, makes it more challenging to ensure tight synchronization of concurrent senders on TelosB-like devices. Further, more nodes transmit concurrently, increasing contention and possibly the probability of collisions [10, 28, 32]. On the other hand, GlossyTX unlocks several advantages by: *i)* solving the problem above induced by the original Glossy scheme, therefore potentially improving latency and/or reliability, e.g., when a flood gets stuck due to external interference, *ii)* enabling significant energy savings, by shortening the radio-on time by removing the unnecessary RX slots or, dually, *iii)* enabling reliability improvements, by replacing them instead with up to $N-1$ TX slots.

The fact that Glossy and GlossyTX strike different tradeoffs would already be enough motivation to consider them both in this chapter. However, an even more compelling reason is the fact that, in the popular DW1000 UWB radio we use, the RX current draw is almost twice than the TX one, making GlossyTX preferable, at least in principle.

Higher-level abstractions: Crystal. The effectiveness of Glossy gave rise to several protocols that are built directly atop the original implementation [30, 31] or slight modifications thereof [32, 84, 164]. Among these, Crystal ([1, 31], Chapter §3.1.1) is particularly suited for the study presented in this chapter because *i)* it does not require modifications to Glossy, therefore allowing us to explore the extent to which the original narrowband can be replaced by our UWB implementations, and *ii)* it exploits concurrent Glossy floods containing *different* packets along with conventional, isolated ones, as described next.

A Crystal schedule is composed of three phases, each corresponding to a Glossy flood:

1. The initial S phase, a *dedicated flood* initiated by the sink to ensure time synchronization.

[1] Hereafter, we use "Glossy" to refer generically to the system in [29], and "Glossy" to refer to the specific scheme derived from it (Figure 6.1) and implemented in this chapter.

2. The T phase, used by concurrent senders to disseminate their data. It is therefore the crucial phase, where *different packets* compete within concurrent Glossy floods originating from *different initiators*.

3. The A phase, directed from the sink to all the nodes. It is performed *in isolation* and provides a network-wide acknowledgement of sorts, enabling each sender to determine whether a retransmission—another Glossy flood in the next T phase—is needed or not.

Crystal dynamically adapts the length of the TA sequence to unpredictable application demands; termination occurs at each node when an empty T phase followed by an A phase containing no acknowledgement are observed for a number R of times.

6.2 Related Work

Concurrent transmissions, as popularized by Glossy, have been a breakthrough in narrowband low-power networking (Chapter 2), showing unprecedented performance and leading to numerous follow-up works [10, 28]. It is therefore not surprising that researchers have begun investigating their applicability to other radio technologies, e.g., Bluetooth Low Energy (BLE) [77, 132]. However, bringing techniques and results from narrowband to the impulse-radio UWB is non-trivial, due to the significantly different characteristics of the PHY layers.

A recent work [81], based on single-hop micro-benchmarks, experimentally verified that concurrent transmissions are possible on UWB links. This holds with *identical* frames, but it was observed that also *different* frames can be supported under certain conditions, namely de-synchronization and signal strength disparity. These findings lay the foundation for our study, where we exploit CTX in actual full-fledged systems for multi-hop data dissemination and collection. Moreover, to further investigate the limitations of the schemes we employ, we analyze the synchronization requirements for correct reception on the time scale of a single data symbol, unlike [81].

Another study [133] reported the use of a Glossy-like protocol to support a UWB localization system, i.e., the main contribution. The work described two necessary conditions for the correct operation of Glossy: preventing data symbol collisions and ensuring signal coherency of concurrent transmissions. As CTX were not the main focus, however, the authors provide very few implementation details and no performance evaluation.

UWB concurrent transmissions have also been recently applied for concurrent ranging [165, 166], in which all receivers of a single ranging request reply together. The authors show that the channel impulse response (CIR) available on the DW1000 can be exploited to collect multiple time-of-flight measurements at once. However, the system is not designed for communication, and the reliability of reception is only tested for the purpose of ranging in a single-hop scenario.

6.3 Glossy on UWB

We first illustrate our implementation of Glossy and GlossyTX on UWB (§6.3.1), focusing on how we exploit the opportunities offered by the DW1000 chip. We then quantitatively evaluate the performance of both variants (§6.3.2), drawing parallels with their narrowband counterparts. Finally, we analyze potential threats to the correct operation of our implementations and critically revisit some of the findings reported in the literature (§6.3.3).

6.3.1 Implementation Highlights

The original implementation of Glossy [29] targeted the TelosB motes (CC2420 radio [110] and TI MSP430F1611 MCU [167]) and was technically complex due to the lack of hardware support for precise timestamping of received packets and scheduling retransmissions. The clocks of the radio and the MCU run asynchronously, which causes a random jitter in the transfer of digital signals between these two components. Due to the non-deterministic time that elapses between a detected radio event and the invocation of the corresponding interrupt service routine, it is difficult to guarantee that the MCU issues the TX command to the radio at a designated time precisely enough to meet the stringent synchronization requirements CTX, and even more poignantly non-destructive interference, entail (§2.1). Moreover, the original Glossy avoided using the platform timers to schedule transmissions, because the stable 32kHz clock does not provide a sufficient resolution and the 4MHz DCO clock is not stable enough. All actions of the protocol are triggered solely by radio events (e.g., end-of-RX, end-of-TX, SFD), further complicating the implementation.

The implementations described in this chapter are for the DecaWave EVB1000 board, equipped with the DW1000 UWB radio and STM32F105 ARM Cortex M3 MCU. Other MCUs can be easily supported; however, their clock speed and the data rate of the SPI bus connecting MCU and radio can affect the timing of Glossy floods.

The DW1000 simplifies the Glossy implementation on many accounts. First and foremost, the DW1000 gives access to its internal clock (Chaper 5); this can be used to *i)* timestamp received frames with sub-ns precision, and *ii)* schedule delayed frame TX with an 8 ns granularity. Both opportunities simplify the implementation tremendously. Random delays in ISR execution are no longer a problem, as the radio can be instructed to begin TX at an *exact* time in the future. Further, there is no jitter or non-determinism, because *a single component*—the radio—both timestamps the RX and triggers the TX using the *same* built-in clock.

Two variants of Glossy. As mentioned in §6.1, our Glossy and GlossyTX implementations have different purposes. Glossy is a faithful re-implementation of the original system in [29], where we exploit the precise timestamping and TX scheduling of the DW1000. Notably, we retain the original scheme in which a TX can be performed *only* for the packet received in the immediately preceding RX slot, except at the initiator (§6.1). This constraint was motivated in [29] by the need to obtain accurate timing information, and is made superfluous by the DW1000 features. Nevertheless, we pre-

Table 6.1: Operation durations for UWB packets (μs).

	SHR	PHR & payload	SPI read & write	Other
15 B	73	45	∼36	∼250
127 B	73	178	∼304	∼250

Table 6.2: Slot durations for UWB and narrowband (NB) in μs.

	UWB	NB
15 B	404	887
127 B	806	4471

serve it to avoid changing the protocol too much, with the intent to have a yardstick enabling *direct* comparison with the body of literature on narrowband Glossy.

Indeed, if one were to allow a TX of a received packet regardless of the outcome of the preceding RX slot, the purpose of the latter would become unclear. A more efficient protocol would be one where, after the first successful RX, the packet is transmitted N times without other RX slots. This is exactly what the GlossyTX variant does, for which our implementation takes full advantage of the DW1000 features. Direct access to the stable clock of the radio greatly simplifies implementation. The latter was actually the major hurdle pointed out by the literature [63, 67], which however lacks in-depth evaluations comparing GlossyTX vs. Glossy.

Anatomy and duration of a slot. A Glossy slot must account for the time necessary to: *i)* read/write the frame payload from/to the radio via SPI, *ii)* transmit/receive the frame synchronization header (SHR), *iii)* transmit/receive the physical layer header (PHR) and the payload, *iv)* perform various software and hardware operations required for packet processing and radio configuration. Table 6.1 shows approximate durations of these steps in the EVB1000, for the two packet lengths we experiment with. By summing up the duration of all the steps, we obtain the actual slot sizes. Their comparison with corresponding slot sizes of narrowband Glossy (Table 6.2) shows a key advantage of UWB: the higher data rate (6.8Mbps vs. 250kbps on the CC2420) allows for slots that are 2.1x and 5.5x smaller, with evident benefits in latency.

On the other hand, concerning the first step above, the DW1000 does not support writing/reading the frame payload during its TX/RX, a feature of the CC2420 which increases parallelism. The DW1000 does allow uploading the payload in parallel with transmitting the preamble; however, we could not exploit this feature because, for the preamble setting we used, the former is slower than the latter.

Dynamic clock frequency calibration. The radio clock of our platform is very stable. Even though DW1000 tolerates up to ±20 ppm [54] frequency drift (Chapter 5), the EVB1000 platform we use integrates a ±10 ppm oscillator, individually calibrated (trimmed) by the manufacturer to achieve ±3 ppm in normal conditions. Nevertheless, temperature and voltage variations may cause its frequency to drift within the

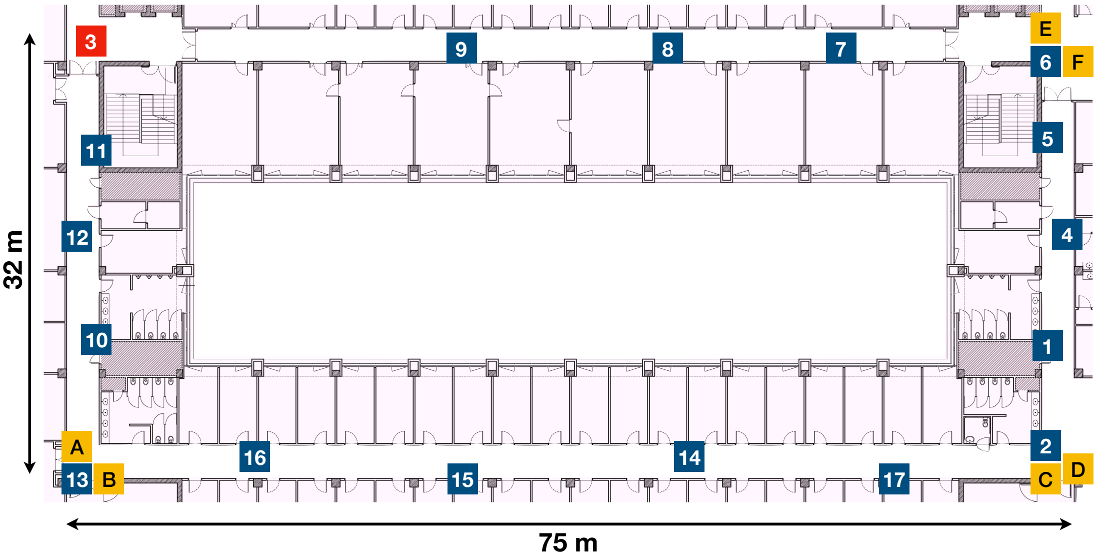


Figure 6.2: Experimental testbed. Out of the 23 nodes available, 22 were running the protocols under study; node 3 served as a sniffer.

full ±10 ppm range.

The authors of [133] report that this drift may undermine the reliability of concurrent transmissions. Therefore, we implement a dynamic frequency calibration of the radio clock of the receivers, relative to that of the flood initiator. Inspired by [133], the calibration is achieved by observing the time offset between the expected and actual arrival of consecutive floods, and adjusting the radio oscillator frequency of every receiver appropriately. This is done by trimming the oscillator with a hardware-defined step of 1.45 ppm. By choosing the value closest to the desired frequency, we ensure that the frequency offset of any device w.r.t. the flood initiator is within ±0.725 ppm. For any pair of non-initiator devices, their relative frequency offset stays within 1.45 ppm.

This dynamic calibration requires the radio clock to remain active in between floods, with the radio in idle mode. This has relatively high power drain w.r.t. sleep and deep sleep states (Table 6.5, [54]); however, the calibration is in general infrequent, hence typically bearing a limited impact on the overall energy consumption. In §6.3.3 we further elaborate on the effect of this technique via experiments that provide additional insights beyond what reported in [133], whose results are based on a custom hardware design achieving higher synchronization accuracy.

6.3.2 Evaluation

We evaluate several aspects of Glossy and GlossyTX, highlighting similarities and differences between them and w.r.t. their narrowband counterparts.

Experimental setup and radio configuration. We tested our implementation of Glossy in a 23-node testbed deployed in the corridors of an office building (Figure 6.2). The communication range normally extends through the entire length of each straight segment, with at least one link with *PRR* $\geq$ 90% for every pair of ad-

Table 6.3: Glossy vs. GlossyTX: reliability.

		Avg. flood reliability [%]				Min. node reliability [%]			
Protocol	**Frame size**	N=1	N=2	N=4	N=8	N=1	N=2	N=4	N=8
Glossy	15 B	100	100	100	100	100	100	100	100
	127 B	99.91	99.997	99.9992	100	99.5	99.95	99.991	100
GlossyTX	15 B	100	100	100	100	100	100	100	100
	127 B	99.91	99.97	99.95	99.997	99.5	99.8	99.0	99.95

Table 6.4: Glossy vs. GlossyTX: latency.

		Mean first relay count			
Protocol	**Frame size**	N=1	N=2	N=4	N=8
Glossy	15 B	1.32	1.32	1.32	1.32
	127 B	1.49	1.37	1.46	1.49
GlossyTX	15 B	1.32	1.32	1.32	1.32
	127 B	1.49	1.48	1.56	1.46

jacent corners. However, exceptions exist where shorter links are less reliable, e.g., node 17 cannot communicate directly with 13. Further, we verified that node 11 is not in communication range with 9; therefore, we do not use node 3 in between them, and set node 9 to be the initiator, achieving a network diameter of 4 hops. The resulting topology is particularly challenging for Glossy and derivatives, as it limits the spatial diversity of their floods by forcing all packets to proceed along a *single* direction.

As for the radio configuration, we use channel 4, 6.8 Mbps data rate, 64 MHz *PRF*, 64 μs preamble, and the maximum transmission power of `0x9A9A9A9A` recommended [55] for the combination of channel and *PRF* we use.

Flood reliability. One of the main benefits of concurrent transmissions is their ability to achieve near-perfect reliability. The latter strongly depends on the number N of retransmissions (§6.1). However, long packets are also known to be detrimental to dependability in narrowband [29, 168]. For these reasons, we experiment with $N \in \{1, 2, 4, 8\}$ and *i)* short packets of 15 bytes, allowing for 8 bytes of payload as commonly used in the literature, and *ii)* long packets of 127 bytes, the maximum allowed by the standard. For every combination of these values, we report results aggregated from 12,000 floods.

Table 6.3 shows that, in our experiments, both variants *always* achieved perfect reliability with short packets, even with $N = 1$. Instead, with long packets this level of dependability is ensured only by Glossy and only with $N = 8$; further, GlossyTX is systematically less reliable for $N \geq 2$, although it achieves a reliability $\geq$ 99% in all cases and for all nodes. Interestingly, the dependability of Glossy increases with N, as expected, while this is not always true for GlossyTX. For $N \geq 2$, in GlossyTX the number of TX slots at each node is higher than RX slots, increasing the number of nodes transmitting simultaneously and therefore the chance of occasional collisions, especially among the long packets.

Latency. These trends are mirrored by the *first relay count* (Table 6.4), i.e., the number of slots elapsed at a node before the first successful RX slot, effectively an indirect measure of latency. The values for this metric are identical for the two variants in the case of short packets or $N = 1$, but are slightly higher in the other cases, meaning that the flood is slightly delayed due to lost packets and consequent retransmissions.

To investigate the *maximum* latency, Figure 6.3 focuses on node 11, the farthest from the initiator. In most cases, the flood reaches this node exactly after 4 hops, with sporadic outliers in case of short packets. With long packets, the maximum latency is still very stable, though bigger, due to larger Glossy slots needed (Table 6.2). However, the 99th percentile shows increase in latency corresponding to 1–2 slots. Overall, there is a weak tendency for latency to grow when N increases—again, more marked with long packets—because of a higher chance of collision at the increase of contention, as discussed before.

Compared to narrowband [29], our UWB implementation provides smaller latency due to the shorter slots used, *with a 52% reduction for short packets and 82% for long packets.*

Energy consumption. Unlike narrowband radios like the CC2420, for which TX and RX have similar energy costs, the RX current draw of the DW1000 chip is almost twice than the TX one (Table 6.5). This motivates investigating the energy consumption of the two Glossy variants, as they exploit very differently these two radio states (§6.1). However, this energy unbalance prevents us from using radio-on time as an energy metric, as commonly done by the narrowband literature. We therefore resort to *modeling* directly the energy costs, as the structure of Glossy protocols is simple and largely deterministic. Specifically, we study the energy cost of Glossy and GlossyTX as a function of the hop distance from the initiator; however, we neglect the contribution of collisions, as these are generally rare and in any case dependent on the specific target environment and network topology.

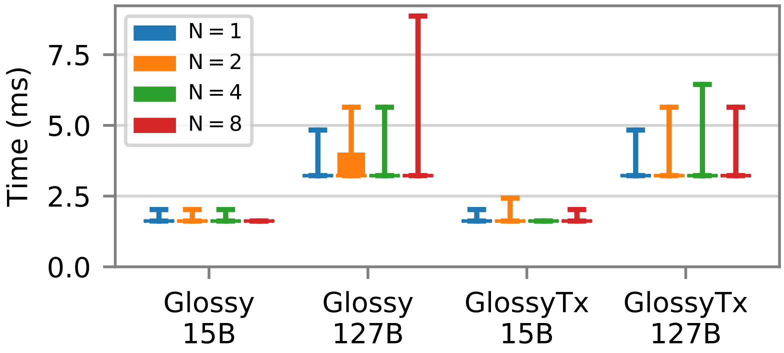


Figure 6.3: Latency of node 11, the farthest from the initiator (4 hops). Bars denote minimum/maximum values; boxes denote the 25–75% percentile.

The drain of electric charge of a node during a flood is:

$$Q = T_{\text{TX}} I_{\text{TX}} + T_{\text{RX}} I_{\text{RX}} + T_{\text{LISTEN}} I_{\text{LISTEN}} + T_{\text{IDLE}} I_{\text{IDLE}} \tag{6.1}$$

where T_i and I_i are, respectively, the time spent and corresponding current draw in a given radio state i (Figure 6.1). From this, energy can be computed easily as $E = Q \cdot V$, with knowledge of the voltage supply (Table 6.5).

The current draw for each state is shown in Table 6.5 for both the DW1000 and the CC2420 used in the original implementation of Glossy. In the latter case, the values are retrieved from the datasheet (p. 13–14, [110]). For the DW1000, the datasheet does not contain information specific to the radio configuration of our experiments; therefore, we use the current draw reported for the most similar one, i.e., the one with the same parameters but channel 2, which has the same central frequency of channel 4 but smaller bandwidth. Note that radio states with a lower power than the idle one cannot be exploited within a flood, due to the large time required by the radio to exit from them (e.g., up to ∼3 ms for the DW1000).

Interestingly, equation (6.1) also models the consumption of *narrowband* Glossy and GlossyTX, albeit with a few caveats. Indeed, the original implementation for CC2420 reads and writes frame data via SPI directly during RX and TX, respectively, avoiding inter-slot processing delays and the need for putting the radio to the idle state between RX/TX slots. Therefore, to apply our energy model in equation (6.1) to narrowband, we *i)* consider as T_{IDLE} the time (192 μs) needed by the radio to switch from RX to TX and the software delay (23.3 μs) required by the MCU to trigger a TX, and *ii)* account for it as if the radio were in RX. This, along with the fact that $I_{\text{LISTEN}} = I_{\text{RX}}$ for narrowband (Table 6.5) leads to the simpler expression for *narrowband* variants:

$$Q = T_{\text{TX}} I_{\text{TX}} + (T_{\text{RX}} + T_{\text{LISTEN}} + T_{\text{IDLE}}) I_{\text{RX}} \tag{6.2}$$

The values of T_i can be determined as a function of N, of the slot duration (T_{SLOT}), of the radio time to TX or RX a frame (T_{FRAME}), and of the first relay counter (C):

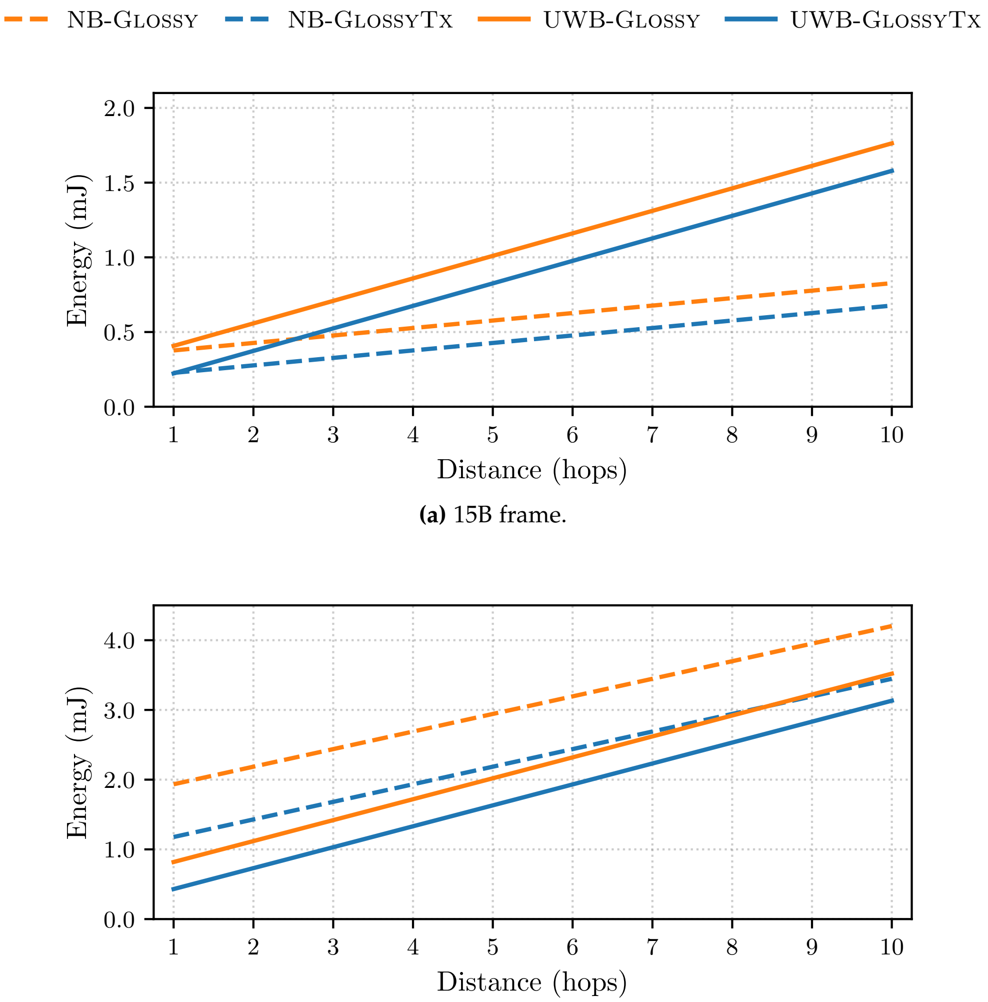
$$\begin{aligned} T_{\text{TX}} &= N \cdot T_{\text{FRAME}} \\ T_{\text{RX}} &= N_{\text{RX}} \cdot T_{\text{FRAME}} \\ T_{\text{LISTEN}} &= C \cdot T_{\text{SLOT}} \\ T_{\text{IDLE}} &= (N_{\text{RX}} + N - 1) \cdot (T_{\text{SLOT}} - T_{\text{FRAME}}) \end{aligned} \tag{6.3}$$

where $N_{\text{RX}} = N$ for Glossy, and $N_{\text{RX}} = 1$ for GlossyTX.

Figure 6.4 shows the resulting energy estimates for $N = 4$; different N values exhibit similar trends. As expected, GlossyTX is more energy-efficient than Glossy both in narrowband and UWB. By scheduling only TX slots after the first successful RX, GlossyTX reduces the flood duration, sparing energy. This difference increases with N. As for the tradeoffs between narrowband and UWB, with short packets the former clearly outperform the latter. Notably, roles are reversed when long packets are transmitted. Despite the higher energy cost of both TX and RX for the DW1000 chip (Table 6.5), UWB GlossyTX is the most efficient solution. At the first hop it consumes nearly one third of its narrowband counterpart, and 4.5x less than narrowband Glossy. However, the gap decreases with hop distance.

Table 6.5: Nominal current draw and voltage supply.

		Current draw (mA)				
	Frame size (B)	I_{RX}	I_{LISTEN}	I_{TX}	I_{IDLE}	**Voltage (V)**
CC2420	any	18.8	18.8	17.4	0.426	3.0
DW1000	15	114.9	113.0	71.5	18.0	3.3
	127	116.5	113.0	61.1	18.0	



(a) 15B frame.

(b) 127B frame.

Figure 6.4: Energy consumption: Glossy vs. GlossyTX for narrowband (NB) and ultra-wideband UWB. Note the difference in scale between the y-axes.

The reason behind the higher energy efficiency of UWB-based solutions with larger payloads is twofold. First, the data rate of CC2420 is 27 times smaller than DW1000 (250 kbps vs 6.8 Mbps); to TX (or RX) 127 bytes of data, narrowband radios stay active about 18 times longer than UWB. Second, the processing delays required by the UWB implementation (§6.3.1) do not significantly affect the overall energy consumption,

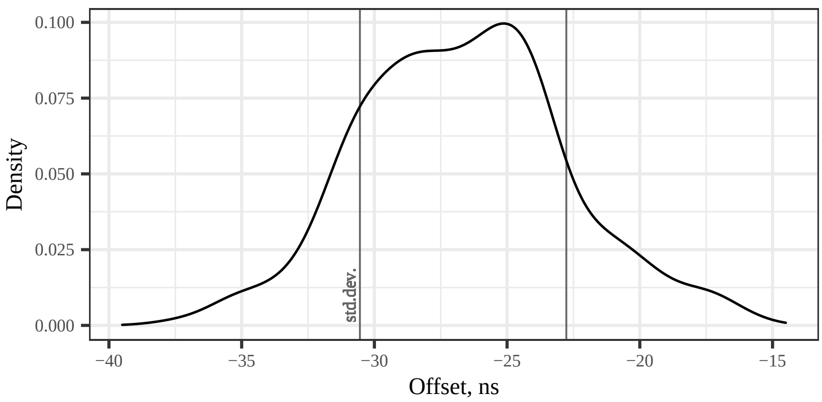


Figure 6.5: Time synchronization error.

as the radio remains idle, therefore draining little power (Table 6.5). After the first successful RX, nodes spend in idle $> 60\%$ of the time, saving considerable energy.

These results are of note; the higher energy cost of UWB radio operation w.r.t. narrowband has been largely considered a key hampering factor for the widespread adoption of UWB technology, especially for communication purposes. Interestingly, Figure 6.4b suggests that this is not necessarily the case; if communications are wisely orchestrated within a properly designed networking stack, UWB protocols can ensure consumption akin—if not even lower than—staple narrowband solutions.

Moreover, energy savvier UWB chips (e.g., from SPARK [169]) are rapidly entering the market; similarly, new generation chips from Qorvo (e.g., the DW3000) have been optimized for low-power battery-based operation [170], further reducing energy costs w.r.t. their predecessors used in this work, and effectively making the energy gap between narrowband and UWB chips less marked. We expect this leap towards increasingly more energy efficient UWB platforms to unleash additional interest in this technology, and further broaden its research and commercial impact.

Accuracy of time synchronization. Finally, we recall that Glossy was originally proposed for time synchronization [29]. It is therefore interesting to investigate this aspect, especially given that the UWB platform we use provides access to its high-resolution clock. To study the synchronization accuracy, we *i)* rely on the privileged position of node 3 (Figure 6.2) and use it as a "sniffer", capable to hear and timestamp the RX of packets sent by node 9, the initiator, and node 11, the farthest from it, and *ii)* analyze their difference w.r.t. the first TX of the initiator (relay count of 0) and the first TX of node 11 (relay counter of 4).

As Glossy was, by design, unaware of the distance among nodes, the reference time at the receivers is always biased by the signal propagation delay, about 333 ns every 100 m. Knowing the overall distance the signal travels in our setup, we subtract that bias and determine the error distribution. This yields a setup similar to the original in [29], where nodes were all on the same desk, making the signal propagation time essentially negligible. Results show an underestimation of $\sim$6.5 ns per hop, resulting in an average offset of -26 ns at 4 hops (Figure 6.5). We attribute this bias to an imprecise antenna delay calibration. Overall, the error distribution covers an interval of 22 ns, essentially due to the 8-ns precision in the DW1000 TX scheduling, accumulating over 4 hops. In the worst case, the TX scheduling error is always exactly 8 ns, yielding a theoretical maximum error of 32 ns for our setup; in practice, the random variations of TX times often cancel each other. In any case, the standard deviation of the error is 3.89 ns, i.e., *almost three orders of magnitude smaller than in narrowband*, reported in [29] to be 2.5 μs over 4 hops.

6.3.3 Exploring the Limits

As we observe instances of frame interference in our testbed setup, we investigate the conditions that can hamper concurrent transmissions in our UWB implementations of Glossy. To this end, we collect empirical evidence in a different, smaller-scale setup where we can precisely control the overlapping of signals. We position a receiver in between 2 synchronized transmitters, at 1 m distance from each, and evaluate the effect of data symbol misalignment and crystal accuracy. This placement is particularly challenging because the strength of concurrent signals is similar at the receiver, reducing the likelihood of the capture effect to take place (§2.1). However, it allows us to derive stronger conclusions regarding the limitations and the ideal conditions of concurrent transmissions, as well as provide evidence that the signal strength and the number of available receivers play a role in the robustness of Glossy.

Payload collisions. As described in Chapter 5, UWB data symbols are divided in two halves for binary burst position modulation (BPM). In principle, a node may fail to receive correctly when two concurrent transmissions are shifted by more than half a symbol duration, and hence one of the pulse bursts occupies the wrong BPM location. The ability to ensure that different transmissions occupy the same BPM locations has been reported as a *necessary* condition to prevent collisions [133]. Specifically, pulse bursts should remain within the T_{BPM} duration (Chapter 5), i.e., 64.105 ns for the 6.8 Mbps data rate used in our setup, resulting in severe limitations on the position of nodes.

However, another related work [81] and our earlier experiments—neither of which mentioned this constraint—hint at the fact that this is actually not at all crucial. To verify this hypothesis, we apply a correction to schedule TXs with a precision of ~ 1 ns (instead of 8 ns) and delay one of the transmitters to cause different degrees of signal overlapping along the symbol duration. Figure 6.6 shows that the receiver enjoys a $PRR \geq 98\%$ even when the delay we artificially introduce is > 64.105 ns, causing pulses to occupy the opposite side of the data symbol. This proves that the DW1000 radio is able to decode the packet correctly even in the presence of pulse bursts in erroneous locations. We speculate that this is due to the coherency mechanisms built in the DW1000 receiver, that allow correct decoding based on the phase of the signal alone. On the other hand, Figure 6.6 also shows that the decoder is affected by concurrency when pulses are really close to ~ 64 ns, i.e., when they occupy a location matching the time-hopping sequence on the opposite side of the BPM-BPSK symbol. However, this constraint is unlikely to happen and even less likely to disrupt a Glossy flood, thanks to the spatial diversity and temporal redundancy of Glossy, along with the inherent variability in the timing of consecutive concurrent TX that are effectively

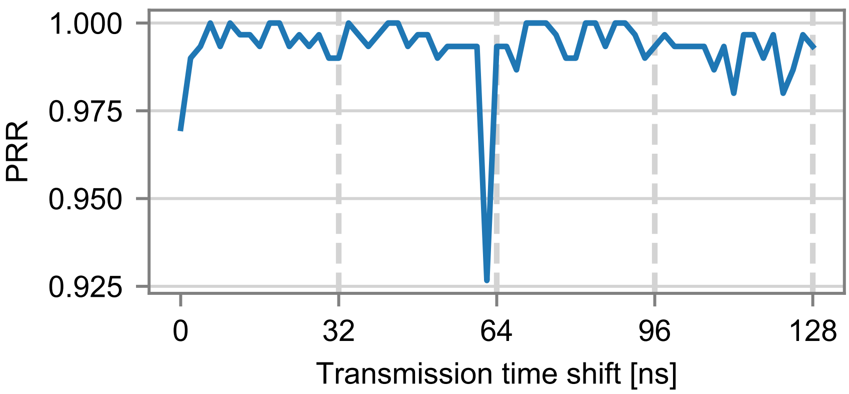


Figure 6.6: *PRR* when a transmission is shifted over the symbol duration (short 15 bytes frame).

scheduled with a 8 ns granularity.

Our findings significantly relax previously reported requirements [133] that, by indirectly affecting the physical location of network nodes, would otherwise hamper the practical applicability of concurrent transmissions over UWB.

Frequency offset. Another necessary requirement reported in [133] is about coherency of two overlapping signals throughout the whole frame. In other words, the phase drift caused by the oscillator frequency difference of two concurrent transmitters should never go beyond half the oscillation period within the frame transmission. In [133], this is translated into a maximum clock frequency offset of 1.39 ppm for 33 bytes frames. The same calculation, applied to the maximum-length packets of 127 bytes allowed by the standard and considered in §6.3.2, yields $0.5/3.4944 \times 10^9/252 \times 10^{-6} = 0.57$ ppm, where 252 μs is the packet TX time with the 64 μs preamble we use. This value is quite far from the 1.45 ppm we achieve with frequency calibration (§6.3.1). Indeed, in our controlled (and challenging) small-scale experiments above, we observe a *PRR* ranging from 0 to 98%, and remaining $\leq$ 10% in many of them, regardless of whether the calibration is used or not. Unfortunately, further improvements are possible only with a custom hardware platform, different from the popular EVB1000 we use in this paper.

On the other hand, to elicit further insights about this constraint as well as assess its impact on short packets, we consider a new small-scale setup where we concurrently transmit 12200 rounds of short packets and simulate the presence of a transmitter with poor crystal accuracy by artificially altering the oscillator frequency, albeit within the ±20 ppm tolerance required by the DW1000 [54]. First of all, we confirm that frequency offsets is not a problem in the case of *isolated* transmissions. In our experiments, a single transmitter with the artificial frequency offset of 10 ppm yields *PRR* $\geq$ 99%, as expected, given that this offset is within the DW1000 tolerance. However, when *multiple concurrent* transmitters are present and one is configured with the same artificial 10 ppm offset, we obtain *PRR* $\geq$ 81.17%, while dynamic frequency calibration (§6.3.1) yields *PRR* $\geq$ 96.74%. This confirms that *i)* the frequency offset matters even for short packets, and *ii)* dynamic frequency calibration is effective in

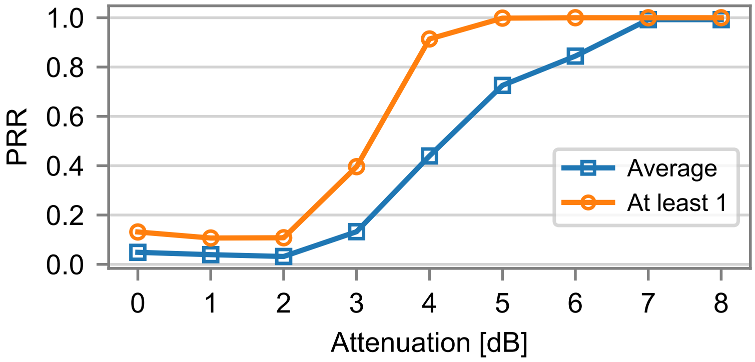


Figure 6.7: Reliability with 4 receivers and 2 transmitters with relative TX attenuation (long packets): average and at-least-one *PRR*.

improving reliability.

On the other hand, we observed these drops in reliability only when *artificially* introducing a frequency offset, as witnessed by the perfect reliability shown in §6.3.2 for both Glossy variants. In practice, the EVB1000 platform we use, factory-trimmed at 3 ppm, guarantees perfect reliability even without dynamic frequency calibration. However, the latter may play a role in deployment environments harsher than the indoor one where we performed our experiments.

Receiver redundancy and TX power. The number of available receivers at a given hop and the relative TX power of transmitters are important factors in the reliability of our UWB Glossy variants, similarly to the narrowband ones [10, 28]. This aspect, largely neglected by related work [81, 133], would deserve a more exhaustive analysis than what possible here. Nevertheless, we offer empirical evidence about it in our small-scale setup with 2 co-located nodes transmitting long packets—the most unreliable—and 4 receivers, again in the most challenging placement where they are at essentially the same distance from transmitters. Further, we configure the transmitters with a relative TX attenuation of 0–8 dB. Figure 6.7 shows that when the transmitters use the same TX power, the average *PRR* is very low; nevertheless, it nearly doubles if computed by considering a reception successful when it occurs on *at least one* of the 4 receivers. Further, it also shows that the *PRR* rapidly grows with the difference in TX power. An attenuation of 7 dB ensures that *all* receivers get the packet; 5 dB are sufficient to ensure reception by at least one of them. These considerations are important, as *one* successful receiver is enough to enable a Glossy flood to progress. On the other hand, the dual also holds; a topology in which progress is ensured by a single forwarder is obviously very brittle. Incidentally, this is the reason why we placed nodes $A - F$ in the corners of our testbed, that is, to eliminate these single-receiver bottlenecks that should anyway be avoided in real deployments.

6.4 Crystal on UWB

The previous section confirmed that our implementations of Glossy for UWB radios provide benefits comparable to those known from the narrowband literature, along with outstanding advantages in latency and network-wide time synchronization. We now turn our attention to a different research question, namely, whether the results from higher-level abstractions and protocols built atop the Glossy layer also transfer to UWB. To provide an answer, we focus on the Crystal protocol [1, 31] briefly introduced in §6.1 and analysed deeper in detail in Chapter 3. We discuss the few changes our Crystal implementation for UWB required w.r.t. the original one, followed by the results of its evaluation in our testbed.

6.4.1 Implementation Highlights

We used the publicly-available code for Crystal [59], and kept the overall protocol logic unchanged; we disabled channel hopping (Chapter 3, [1]), as it is not the focus of this chapter. However, a few minor modifications were necessary, motivated by the different operation of the underlying radios.

Crystal detects termination based on the absence of received packets; it is therefore crucial to tell apart absent transmissions from failed receptions. In narrowband, noise detection enabled Crystal to defer termination if no packet is received but strong noise is detected. However, this mechanism relied on clear-channel assessment (CCA), not present in UWB. Further, it did not provide direct evidence of a failed RX, but only of the *possibility* of one, due to noise.

On the other hand, the DW1000 offers rich information about RX errors, which we exploit in our implementation. This information is signalled when the radio detects a preamble but fails to decode either the SFD or the data portion of the packet, due to Reed-Solomon, SECDED or CRC errors (Chapter 5). A "spontaneous" preamble detection may still happen without any TX, but is highly unlikely in practice [171]. Therefore, the mere presence of a preamble signals with very high probability that one or more nodes are sending packets but their data cannot be decoded, likely because of collisions; we leverage this notion to postpone nodes termination. We verified that this technique significantly improve the reliability of our UWB implementation of Crystal.

Finally, likewise the IEEE 802.15.4 narrowband version, our UWB implementation relies on the MCU 32 kHz timer of the EVB1000 board to schedule its activities; data TX in the shared T slots may hence overlap within 30 μs. Relying on the more accurate radio clock would require significant changes in the code, in contrast with our desire to minimize them. Further, it would likely bring little to no benefits, given that a slight de-synchronisation of transmitters is reported in [81] to increase the reliability of concurrent transmissions of *different* packets. Following another recommendation of [81], we also increase the SFD timeout by 32 μs, to account for possible frame offsets caused by the timer resolution.

6.4.2 Evaluation

We evaluate Crystal in our testbed, with the same configuration of §6.3.2; further, node 9 is the sink, maximizing the network diameter (Figure 6.2). We are interested in the overall reliability in delivering packets at the sink, but also in ascertaining the underlying raw reliability of the floods disseminating *different* packets and competing in the same shared Crystal slot. In doing so, we experiment with both variants of the underlying Glossy, as well as with short and long packets.

We consider two key parameters influencing performance: the number U of concurrent updates and the number N of TX in the Glossy flood inside a shared Crystal slot. U determines how many nodes have to transmit a packet within each Crystal epoch, defining the degree of concurrency. We explore $U \in \{1, 2, 5, 7, 10\}$, i.e., up to half of the network, chosen at random among non-sink nodes; we also consider the extreme case where all the $U = 21$ non-sink nodes transmit concurrently. N defines the degree of redundancy at the Glossy layer; we explore $N \in \{1, 2, 4, 8\}$ as in §6.3.2. We set a default of $U = 5$ and $N = 2$ when exploring the other parameter.

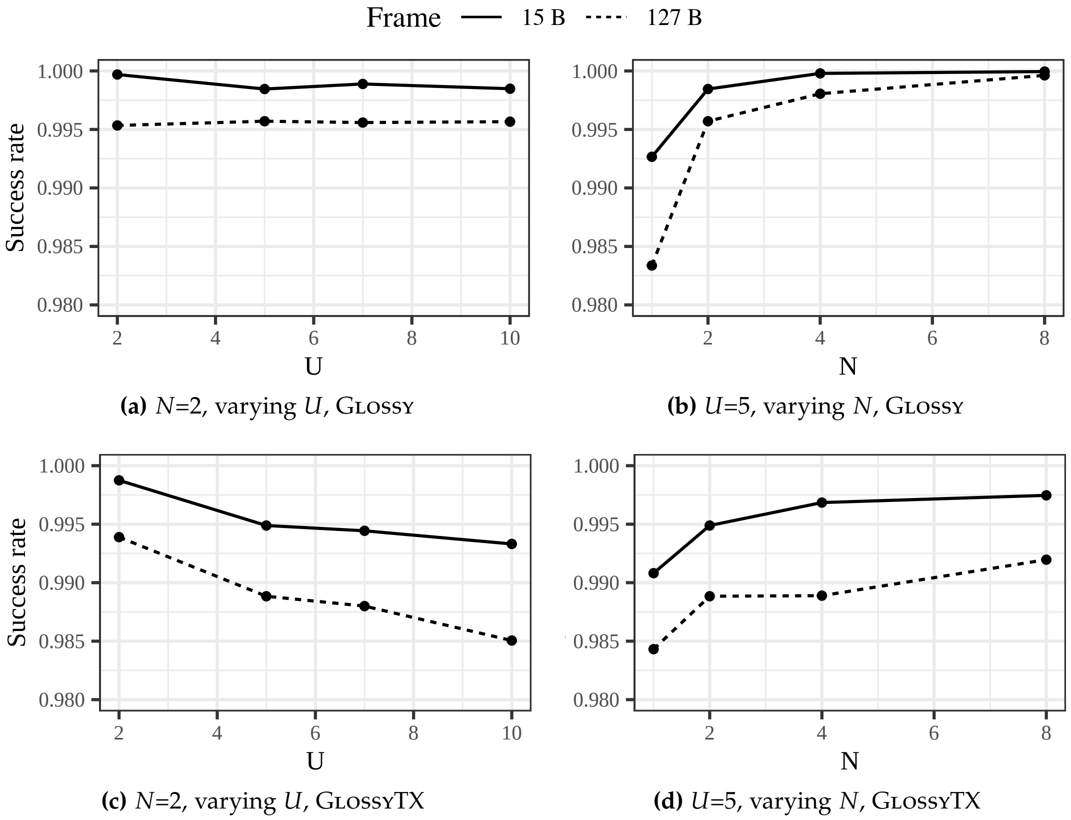
For each combination in this space we collect traces of 1000 Crystal epochs, i.e., $1000 \times U$ packets transmitted.

Overall reliability. Our UWB implementation of Crystal ensures remarkable reliability with both short and long packets, regardless of the Glossy variant it builds atop. With the short 15 bytes packets, Crystal correctly delivered *all* the transmitted messages, more than 150000 in total, independently from the parameter configuration, and notably even when $U = 10$ nodes—i.e., half of the network—were concurrently transmitting. Moreover, even with the longest 127 bytes packets Crystal achieves more than 99.9% reliability. This near-perfect dependability is fully in line with the results originally reported in [31], therefore confirming that the performance Crystal achieves in narrowband can be harvested also in UWB.

Reliability of shared slots. This outstanding reliability is achieved in Crystal via mechanisms that *mask* the packet losses in the underlying concurrent Glossy floods. Therefore, in the light of ascertaining the extent to which concurrent transmissions of *different* packets can be used as a building block for other protocols [30, 32, 84], we now focus on the performance of shared slots in isolation.

Specifically, we look at the first T slot of each Crystal epoch, when there are exactly U nodes transmitting simultaneously. For each node, we define the success rate metric as the ratio between the number of floods when the node received *any* packet over the total number of floods when the node was listening in the first T slot, as in [31]. Figure 6.8 shows the average success rate, and Figure 6.9–6.10 the distribution of this metric across nodes via the complementary empirical cumulative distribution function (CCDF).

The charts exhibit clear trends. First of all, Glossy outperforms GlossyTX systematically, across all configurations. In particular, the reliability of Glossy remains nearly constant w.r.t. the increase in the number U of concurrent senders, while GlossyTX shows a marked decline. This is a consequence of the fact that the density of transmissions in GlossyTX is much higher than in Glossy due to its more aggressive re-transmission policy, as already pointed out in §6.3.2. Things obviously exacerbate as U increases, given the higher number of *different* packets circulating in the network. Long frames degrade reliability of both variants, again consistently with §6.3.2, and with a more marked effect on GlossyTX, as per the observations above. In any case,



(a) N=2, varying U, Glossy

(b) U=5, varying N, Glossy

(c) N=2, varying U, GlossyTX

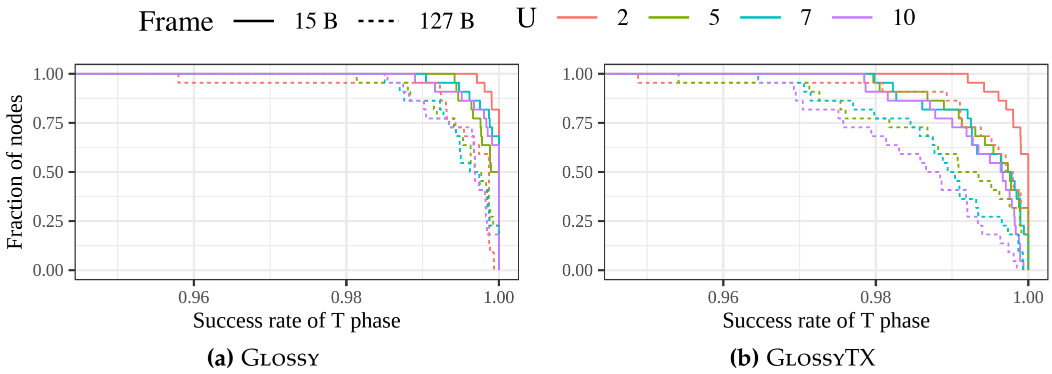
(d) U=5, varying N, GlossyTX

Figure 6.8: Average success rate in the first T shared slot, for different values of N and U.

the absolute worse success rate recorded across all these many experiments was 95%, which is still very good.

Finally, reliability increases with the degree of redundancy induced by N. At the highest value tested, N = 8, the average reliability of Glossy reaches 99.994% with short packets and 99.96% with long ones, caused by packet losses at a single node. As for GlossyTX, the average reaches a plateau of 99.7% for short packets and 99.0% with long ones. These figures are remarkable, considering that *i)* they are achieved with concurrent transmissions of *different* packets, and *ii) without* the reliability mechanisms of Crystal, which are nonetheless key to spare the steep energy costs induced by a high value of N.

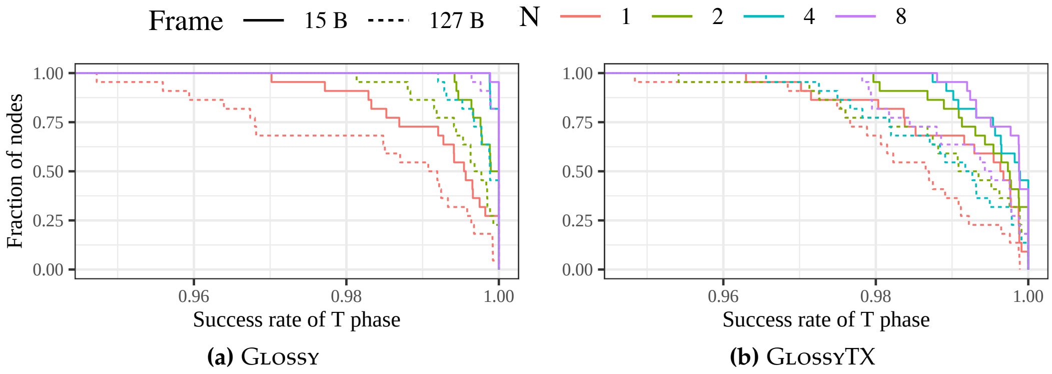
Extreme case. We also tested Crystal in the extreme scenario where *all* 21 non-sink nodes transmitted in all epochs (U = 21). Performance with both Glossy variants were very similar; therefore, in the following we not analyse them separately, but provide a general overview of the results. With N = 2, the success rate of the T phase drops as low as 80% even with short packets; nevertheless, the overall reliability at the sink remains at 99%. As expected, long packets exhibit worse reliability, with a success rate of the T phase at ∼75% and an overall reliability of ∼95%. At the other extreme, N = 8 yields a near-perfect overall reliability for short packets, with several runs at 100%, and ∼98% for long packets, despite an underlying success rate at ∼88% and ∼79%, respectively. Overall, these results confirm once again the effectiveness



(a) Glossy

(b) GlossyTX

Figure 6.9: CCDF for T success rate vs. U (N=2).



(a) Glossy

(b) GlossyTX

Figure 6.10: CCDF for T success rate vs. N (U=5).

of the “safety net” provided by Crystal reliability mechanisms, as already observed for narrowband in similar extreme scenarios [1, 31].

6.5 Discussion

We distill salient findings from our results and offer some considerations that may inspire future work on the topic.

Similarities vs. differences. First and foremost, our experiments demonstrate that Glossy-like mechanisms achieve in UWB benefits similar to narrowband, i.e., low latency, high reliability, low energy consumption——all at once. Energy consumption has slightly different tradeoffs than in narrowband, due to the significant imbalance between RX and TX in the DW1000. Nonetheless, our model suggests that UWB protocols are not necessarily more energy expensive than traditional narrowband ones, as someone could instead expect given the notoriously higher TX and RX costs of such transceivers. At the increase of packet size, the higher UWB TX/RX consumption is gradually compensated by the drastic reduction in radio active time unlocked by the larger data throughput, making UWB solutions particularly suited for high-rate communication, where they actually yield lower energy expenditure w.r.t. their narrowband counterparts. The latency achievable in UWB is instead significantly lower than in narrowband, as a consequence of the high-precision clock and the higher data rate. The former also enables a three order of magnitude improvement in the accuracy of network-wide time synchronization, which was actually the original motivation for Glossy.

Interestingly, our in-depth analysis in §6.3.3 reveals that the DW1000 is capable of decoding concurrent pulses—and therefore packet TX—even when severely misaligned. On the other hand, due to the encoding based on short pulses rather than long waves, the crystal frequency offset is potentially more of an issue for UWB than it is for narrowband, with a stronger impact on the trade-off between the packet length of the concurrent transmissions and their reliability. However, as for narrowband, differences in the signal energy of the concurrently received messages, along with spatial and receivers' diversity, improve performance considerably, also for long frames.

Glossy or GlossyTX? A first observation is that the question is actually an open one for narrowband. Indeed, the potential superiority of GlossyTX is rather anecdotal, as it mainly derives from the ad hoc setup of the EWSN Dependability Competition and has never been rigorously analyzed across different system parameters.

For UWB, our study shows that GlossyTX is more energy-efficient than Glossy; for long packets, it achieves a consumption even lower than narrowband. Therefore, it would seem obvious to always use it in place of Glossy. Nevertheless, our results demonstrate that this is not always necessarily the case. Indeed, the other side of the coin is that the aggressive re-transmission policy of GlossyTX is prone to increasing the number of collisions, affecting reliability. This behaviour is notable both in same-packet and different-packet floods. However, the actual impact ultimately depends on how the Glossy layer is used in the specific traffic profile and/or higher-level system (e.g., Crystal, in our case).

The dual argument is that Glossy appears slightly more reliable, due to the alternating pattern of TX and RX slots that reduces the "density" of concurrent senders and thus the probability of collisions. An opportunity for future work is to find a scheme striking the right balance between the back-to-back transmissions of GlossyTX and the sparser transmissions, yet rigidly alternating with receptions, of Glossy. The contribution presented in the next chapter is directly inspired by this research curiosity, although it tackles the problem from a different perspective, focusing on data collection and exploring alternate CTX design strategies, outside the perimeter of what enabled by the Glossy flooding primitive.

Transferring results from narrowband to UWB. As repeatedly mentioned, there is a substantial literature on concurrent transmissions for narrowband, including systems that built atop the original Glossy to support alternate network functionality [10, 28].

Our experience with "porting" Crystal to UWB and the related evaluation (§6.4) shows that the effort required is relatively small while, on the other hand, the benefits that can be attained are entirely in line with those shown for narrowband. Of course, it would be a leap of faith to claim that the same can be done for all other higher-level abstractions in the literature. However, our experience hints at the fact that this may actually be the case for several of them, especially those that run atop an unmodified Glossy layer, e.g., notably including LWB [30].

We argue that pursuing this question is actually important. First, to amplify the impact (and awareness) of the body of literature on concurrent transmissions on other radio technologies and hence research communities, and concretely demonstrate that it applies to a far more general scope than the hardware niche it was originally developed for. Second, to let the UWB ecosystem benefit from more than a decade of research in low-power wireless networking that could potentially *i)* translate into a plethora of (hitherto missing) fast, dependable, energy efficient, and ”ready-to-use” UWB protocol stacks, and similarly *ii)* lay a solid foundation for new research on the topic.

6.6 Conclusions

We explored the extent to which concurrent transmissions, made popular by Glossy for IEEE 802.15.4 narrowband, can be exploited in UWB via full-fledged, readily-available systems, and ascertained what is the corresponding performance. Overall, the answer is very positive: both variants of Glossy we consider, as well as CRYSTAL, a higher-level abstraction building atop of it, yield in UWB a reliability similar to the one observed in narrowband. Further, the higher clock resolution and data rate in UWB unlock significant latency improvements, which become order-of-magnitude ones for time synchronization, the original motivation of Glossy. We provided a detailed account of the opportunities this UWB platform enables for an efficient implementation of concurrent transmissions, an analysis of the threats to performance, as well as investigated the effort required to exploit CRYSTAL atop the Glossy layer.

Beyond the qualitative lessons learned and quantitative results reported here, we also release the systems we described as open source [60], enabling their immediate use and improvement by researchers and practitioners, and generally inspiring future work and research interest.

Motivated by the remarkable performance achieved by CTX in UWB, in the next chapter we explore further this synergy, and design WEAVER, a novel ultra-fast networking primitive for UWB. Nevertheless, differently from the study proposed here, we *i)* focus on aperiodic data collection, given the importance that efficiently supporting this type of communication has for the wireless control systems targeted in Part I, and *ii)* devise an alternate, fine-grained CTX design paradigm unleashesing the potential of concurrent transmissions without the overhead and the rigidity that relying on predefined, fixed-structure Glossy floods entails.

7

One Flood to Route Them All: Ultra-fast Convergecast of Concurrent Flows over UWB

Our previous contribution showcases how the powerful features of ultra-wideband (UWB) radios can be exploited to effectively implement and support Glossy-based communications, which achieve in UWB a reliability and energy efficiency akin to narrowband, while unlocking remarkable improvements in latency and network-wide time synchronization. In this chapter, we move one step further. Specifically, we observe that a staple convergecast stack for UWB is currently missing, a witness to the limited attention the low-power wireless networking community has hitherto devoted to this physical layer. Leveraging CTX, we aim at filling this gap by providing an ultra-fast, dependable, and efficient data collection primitive for UWB, concretely demonstrating the effectiveness and applicability of this technology for sensing and communication, and thus fostering and accelerating its adoption beyond localization-centric applications.

As analysed in §7.1, most existing CTX-based solutions take advantage of concurrent transmissions via the Glossy system, by simply scheduling its network-wide floods in different ways depending on the traffic pattern and goals at hand. In other words, Glossy is used to a large extent as a *monolithic blackbox*, with little or no modifications by higher-layer protocols. This is a reasonable design decision enabling faster and reliable prototyping, given the system-level complexity of Glossy, which we acknowledge has significantly fuelled CTX research. On the other hand, this decision stifles

This chapter revises the publication [134]: M. Trobinger, D. Vecchia, D. Lobba, T. Istomin, and G.P. Picco. "One Flood to Route Them All: Ultra-fast Convergecast of Concurrent Flows over UWB". in *Proc. of the* 18th *ACM Conference on Embedded Networked Sensor Systems*. SenSys. 2020. DOI: 10.1145/3384419.3430715.

the exploration of alternate, finer-grained designs that, by *directly and individually* exploiting CTX, could further expand the breadth and depth of CTX research, and in turn the impact and applicability of this technique.

This second approach is precisely the one we follow in this chapter, in which we retain network-wide CTX flooding as the main communication mechanism, but *fundamentally change* its purpose and operation.

Exploiting the full potential of CTX: Weaver. A Glossy flood is *entirely* dedicated to disseminating a *single* packet across the entire network. A global scheduling of transmission (TX) and reception (RX) slots, whose redundancy is governed by the user-defined parameter N, achieves the aforementioned excellent performance, repeatedly proven in the literature. Nevertheless, flooding is still a wasteful, network-wide operation, exacerbated by the fact that reliability is directly tied to the redundancy factor N; the higher its value, the higher the number of times a packet is (concurrently) retransmitted and, therefore, the higher the (network-wide) energy consumption. Crucially, the value of N, and hence the duration of the flood, is *fixed before execution*, therefore intrinsically prone to over- or under-provisioning, hampering lifetime and reliability, respectively.

In contrast, the communication primitive we describe here, called Weaver, is expressly designed to *concurrently disseminate towards a receiver different packets from multiple senders in a single, self-terminating, network-wide flood*, significantly improving on latency, reliability, and energy consumption w.r.t. Glossy-based systems.

Weaver achieves these goals with several mechanisms, each built directly atop individual CTX. As in all Glossy-based systems, Weaver alternates (short) periods executing its network-wide flood with (long) periods of inactivity, all implicitly time-synchronized by system operation. Each node, including the sink, executes a time-slotted sequence formed by a TX slot followed by *two* RX slots, that repeats until the flood self-terminates. Adding an extra RX slot to the TX-RX scheme of Glossy may seem a minor change; yet, it is crucial to unlock significant performance gains. Indeed, the resulting 3-slot structure, combined with the propagation of an initial message from the sink, staggers the (concurrent) TX and RX of nodes at different hop counts from the sink, enabling multiple flows to *co-exist* within the same flood without disrupting each other. Further, it enables Weaver to exploit a combination of *local*, 1-hop broadcast acknowledgments and *global*, sink-initiated ones that, together, adaptively *i)* suppress unnecessary packet propagation or, on the contrary *ii)* trigger retransmission of packets that have been lost, therefore decreasing energy consumption and increasing reliability w.r.t. the fixed redundancy of Glossy-based approaches.

Goals, methodology, contributions. We discuss the design rationale and goals for Weaver (§7.2) and offer an analytical model confirming the intrinsically superior performance achieved w.r.t. Glossy-based state-of-the-art representatives, before delving into a more in-depth illustration of protocol details (§7.3).

Systems based on CTX are notoriously complex. This, however, is to a large extent a relic of a past when the lack of proper hardware primitives required complex designs yielding timing guarantees. Nowadays, many radio transceivers offer rich primitives, notably including the ability to schedule transmissions accurately, that drastically simplify CTX-based system implementation. The Decawave DW1000 UWB radio [54] we focus on in this part of the thesis is one of these powerful "new-generation" transceivers.

Once the leap is made from coarse-grained, rigid Glossy floods to finer-grained alternatives based on individual CTX, many solutions are possible, catering for different requirements. Our *modular* implementation (§7.4) sharply decouples the mechanics of accurately scheduling TX and RX slots and ensuring time synchronization, delegated to a Time Slot Manager (TSM) component, from their higher-level orchestration in Weaver, which can be easily replaced by alternate designs. This approach simplifies and speeds up the development of highly customizable and dynamic CTX protocols by hiding much of their low-level complexity to the protocol developer—an asset, which can further fuel CTX research. A component estimating energy consumption, currently missing for UWB, is also provided. We release these reusable components as open source [60], contributing to further developments in the fast-growing UWB (and CTX) research community.

We evaluate Weaver in a 36-node testbed at our premises (§7.5). We first analyze the impact of key design decisions with dedicated experiments. Next, we compare directly the performance of Weaver against Crystal [1, 31], a state-of-the-art Glossy-based convergecast protocol whose publicly-available UWB implementation we adopt as baseline to compare Weaver against is presented and evaluated in Chapter 6. Our results confirm the trends observed in the analytical model, e.g., showing that Weaver can deliver at the sink 30 concurrent flows in about 100 ms, achieving a reduction of ∼70% in both latency and energy consumption w.r.t. Crystal while achieving near-perfect reliability due to the lower contention induced by the finer-grained, adaptive use of CTX. Moreover, the ultra-fast dissemination achieved by Weaver, along with the inherent redundancy offered by CTX, makes our system resilient to topology changes, e.g., induced by mobility.

Finally, although our Weaver prototype targets UWB, its protocol design does not rely on features specific of this PHY layer. The main contribution of this chapter lies indeed in the novel idea of merging multiple packets flows in a single network-wide flood. This notion, not tied to a specific platform or PHY-layer, can be applied to other radios supporting CTX, amplifying the impact and contribution outlined here, and pushing the envelope of what CTX can achieve in low-power wireless communication at large. We concisely discuss these and other follow-up opportunities in §7.6, before ending the chapter (§7.7) with brief concluding remarks.

7.1 Background and Related Work

The work presented in this chapter embraces two main research topics: concurrent transmissions—the common denominator of all the contributions presented in this thesis—and UWB communication, the focus of this part. An introduction to CTX and an in-depth analysis on the underlying PHY-level phenomena enabling the widespread adoption of this technique is presented in Chapter 2. Chapter 5, instead, offers the necessary background on UWB technology, while Chapter 6 demonstrates that CTX are exploitable for multi-hop UWB communication, and overviews the CTX-based UWB literature. We refer the interested readers to these chapters to avoid unnecessary repetitions. In the following, instead, we *i)* clarify the motivations that inspired our work by analysing the impact, influence, and implications brought by Glossy on CTX research, and *ii)* discuss similarities and differences between WEAVER and the few existing non Glossy-centric CTX protocols.

Glossy as a reusable building block. Glossy [29] was the first to exploit CTX into a reliable, efficient, and publicly-available system providing network flooding and time synchronization. Several others leveraged the low-latency, high-reliability, low-consumption, network-wide flooding of Glossy as a primitive building block for higher-level abstractions. LWB [30] supports different traffic flows (many-to-one, one-to-many, one-to-one) by properly scheduling them as individual Glossy floods from a single initiator. CRYSTAL [1, 31] ensures dependable and energy efficient many-to-one convergecast via phases in which Glossy floods from multiple initiators compete, followed by others in which the sink alone has the opportunity to acknowledge the received packet. Other systems [102, 146, 147, 172] explore variants of these concepts.

Reusability vs. degrees of freedom. Interestingly, in all these systems (and many others) Glossy is reused as a *monolithic blackbox*, with little or no modification. Indeed, only few protocols make relatively small modifications to Glossy that, however, are not geared to change its core functionality, rather to improve its performance, e.g., increasing reliability via channel hopping [64, 84] and/or reducing latency [63, 67, 121].

The direct reusability of Glossy actually fueled research on CTX, enabling researchers to experiment with new solutions while avoiding the intricacies of the CTX implementation. Nevertheless, at the same time it also fossilized research on CTX to a large extent.

Indeed, a Glossy flood is entirely dedicated to disseminate a *single* packet from a *single* initiator, with the intent to exploit non-destructive interference among forwarding nodes. LWB and others (e.g., [102, 146]) rely directly on this feature. CRYSTAL and others (e.g., [147]) push Glossy further by having *multiple* initiators compete within the same flood, relying on the capture effect; however, the final outcome is still a single packet from only one of the initiators.

Weaver: Back to individual CTX. In contrast, we take a significantly finer-grained perspective and free ourselves from the mechanics of Glossy, exploiting CTX directly and individually.

Only few systems hitherto resorted to a similar approach, and always to support many-to-many communication. Chaos [32] realizes network-wide aggregation of data from multiple initiators via competing floods. Mixer [85] and CodeCast [164] exploit network coding to improve performance and reliability. In all of them, the TX-RX scheme of Glossy is replaced by a sequence of indistinct slots in which a node dynamically decides whether to TX or RX; in Chaos, a TX happens deterministically when the node observes new information affecting the global aggregate, while in Mixer and CodeCast the decision includes also a probabilistic component.

Our research endeavour differs from the above on two accounts. First, it explores a strategy in which slots are not indistinct, rather they have a preassigned role, as in Glossy. However, differently from Glossy, our scheme is capable of deterministically intertwining multiple flows from different initiators whose dissemination and termination we govern with a novel, adaptive strategy as described in §7.2 and §7.3. Second, instead of many-to-many, we tackle convergecast traffic, and provide efficient support for aperiodic collection. This communication pattern is arguably more popular, thanks to its use in monitoring and data collection applications, yet hitherto dominated by systems relying on monolithic Glossy floods. By "breaking" this unit of communication and achieving better performance via individual CTX we exemplify the power of this alternate design mindset, possibly paving the way to exploration of alternate schemes catering for this and other traffic patterns.

Finally, to facilitate this exploration by others, and simplify our own system development, we follow a recent trend [84, 85, 173] and sharply separate the fine-grained CTX engine from the Weaver protocol built atop it, while taking advantage of features provided by modern transceivers that greatly simplify programming.

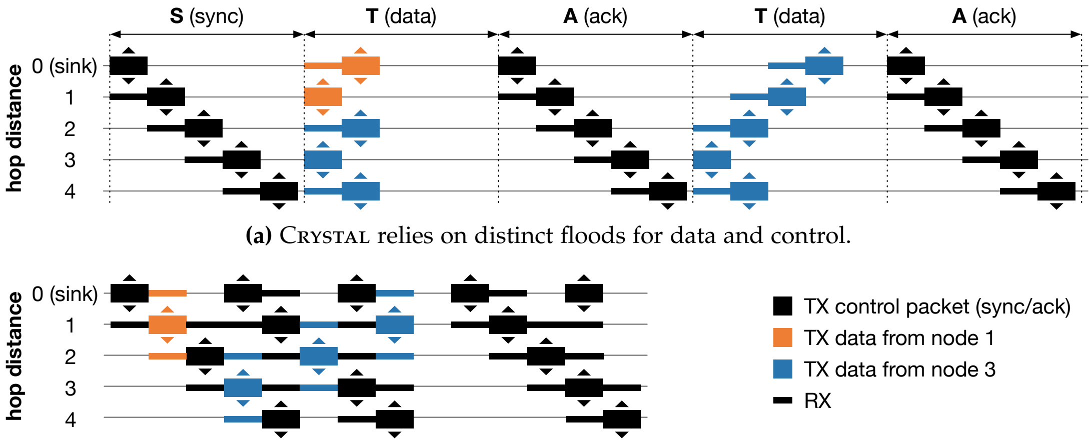
7.2 Design Goals and Principles

We designed Weaver to tackle inherent inefficiencies of data collection protocols that rely on Glossy floods as the only communication primitive. To better understand the crux of the matter, we focus on Crystal and analyze critically its operation, in particular its reliance on Glossy and the related shortcomings (§7.2.1). Motivated by this analysis, we then provide a concise overview of the key design decisions and goals at the core of Weaver (§7.2.2). We conclude the section by providing a quantitative argument, supported by analytical models, showing that our fine-grained CTX-based design is intrinsically superior to Glossy-based ones (§7.2.3).

7.2.1 The Drawbacks of a Glossy Legacy

Crystal in a Nutshell. Crystal [31] targets scenarios with aperiodic data collection and sparse traffic, using a schedule (Figure 7.1a) composed of three phases, each executing a Glossy flood: *i)* the initial S phase, originating at the sink, provides network-wide time synchronization; *ii)* in the T phase, concurrent floods initiating at nodes with data packets compete; due to the capture effect (§2.1), one is received at the sink with high probability, e.g., the orange one in Figure 7.1a; *iii)* the A phase originates at the sink, which exploits it as a network-wide acknowledgment informing senders of whether their packet has been received; in Figure 7.1a, this enables retransmission of the blue packet in the next T phase. The alternation of T and A phases (TA pair in Crystal jargon) continues until all pending packets are received and acknowledged, and a TA pair without data is observed for a pre-defined number of times.

Crystal: A critical look. Although Crystal already achieves remarkable, state-of-



(a) Crystal relies on distinct floods for data and control.

(b) Weaver weaves data and control into the same flood.

Figure 7.1: Sample executions of Crystal and Weaver; nodes at hop distance 1 and 3 from the sink transmit a packet. The termination phase is not shown.

the-art performance, it is inherently limited by its direct reliance on Glossy, as many others (§7.1).

A first problem is that *each* Crystal *phase is a Glossy flood that must complete before a new one is started*. The schedule on each node must allocate enough time for flood propagation, determined with knowledge of the network diameter and number of retransmissions.

This problem is exacerbated by the fact that *the number N of retransmissions is fixed*, yielding other inefficiencies: *i)* retransmissions are performed regardless of whether a packet has already successfully propagated, therefore hampering latency, lifetime, and possibly reliability due to unnecessary contention, and *ii)* the fixed value of N cannot dynamically cater for transient sources of unreliability, common in wireless communications; either the worst case is assumed, hampering lifetime in the normal case, or the latter is assumed, hampering reliability when disturbances occur.

These problems are shared by all Glossy-based approaches (§7.1). Specific to Crystal is the sink-initiated, network-wide acknowledgment in the A phase. The latter has been repeatedly shown (e.g., in [1, 31] and in the context of the EWSN Dependability Competitions [3, 42]) crucial to achieve near-perfect reliability even with aperiodic, bursty traffic and heavy interference. This superior reliability motivates our use of Crystal as a baseline instead of, e.g., LWB or derivatives, besides the lack of UWB implementations. However, the asset brought by A phases also bears drawbacks, again inherited from Glossy. One directly descends from the problems above: successful propagation of a packet requires (at least) *two* phases, transmission (T) and acknowledgment (A), both *fixed-length* and *strictly separated*, wasting energy and time.

A less obvious problem is that *the A phase is oblivious of the reason why packets are not received*. In Crystal, the common case is that transmissions from U initiators compete in the same T phase; the A phase is crucial to inform senders of whether their packet should be re-sent. Nevertheless, the A phase also counters packet losses due to collisions and/or external interference. These are often concentrated in network "pockets" where packet transmissions violate the constraints for successful CTX, yielding a collision. This situation stems from a combination of neighbor density, relative signal power, and environmental conditions, extremely hard to predict yet likely to repeat due to the periodic operation of the protocol. Unfortunately, in Crystal the only option is to inform the network about the missed packet, and hope that somehow the problem solves itself.

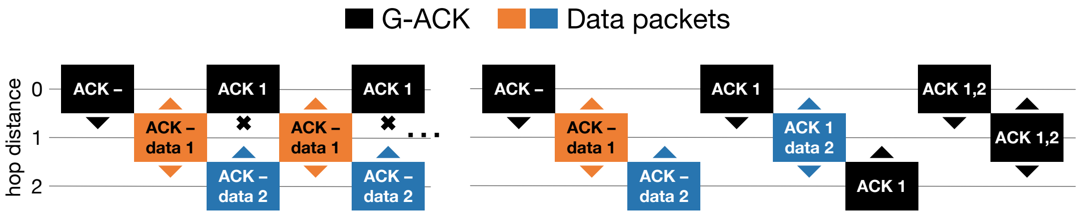
7.2.2 Weaver: The Power of Fine-grained CTX

We tackle the limitations above at their core by removing the dependency on Glossy, therefore regaining the full degrees of freedom available once the unit of communication becomes an individual CTX rather than a monolithic Glossy flood. This finer-grained design mindset enables us to bring to the table several techniques that, together, improve significantly the already remarkable performance of Crystal, in its role of representative of Glossy-based approaches. The most significant point of departure is that Weaver collects and acknowledges multiple packets *within a single flood*, where the different flows coexist without disrupting each other. This is achieved by wisely combining flooding with topology information, acquired within data collection, that are typically disregarded in Glossy-based systems.

On the surface, Weaver resembles other CTX-based convergecast protocols, e.g., Crystal or LWB. Time is divided in rounds (*epochs*) of fixed length, each containing a time-slotted communication schedule. The *sink*, i.e., the node collecting data, periodically starts a new epoch by broadcasting a synchronization packet; nodes re-propagate it concurrently, exploiting CTX, and align their slots to the one of the sink, beginning execution of the global schedule.

Epoch bootstrap: Acquiring one-shot topology information. Differences begin with this first step, which in Weaver is exploited not only to enable nodes to time-synchronize, but also to *learn their hop distance from the sink*. This topology information is exploited by a node to relay only packets from nodes at the same, or higher, hop distance from the sink, i.e., favoring packets in need to make progress and quenching those already ahead, reducing contention.

This *topology information is not explicitly maintained*, as in conventional route-based approaches, *rather passively learned* during the initial dissemination, hereafter termed as *epoch bootstrap*. In this respect, the ultra-fast dissemination achieved by Weaver doubles as a fundamental asset for its operation. As we show later (§7.5), Weaver disseminates 30 flows over 6 hops in only ~100 ms. Therefore, *during this very short*



(a) TX-RX schedule (Glossy). **(b)** TX-RX-RX schedule (Weaver).

Figure 7.2: Weaving flows of data and acknowledgments.

time span the network can be effectively considered as static, and the topology learned during bootstrap safely assumed to persist throughout the entire flood, even in scenarios with node and/or sink mobility, as we investigate in §7.5.4.

Weaving packet flows. WEAVER merges flows from multiple senders (initiators) into a single flood where data *implicitly* follows the upward gradient towards the sink established by the epoch bootstrap, and acknowledgments flow downwards towards initiators.

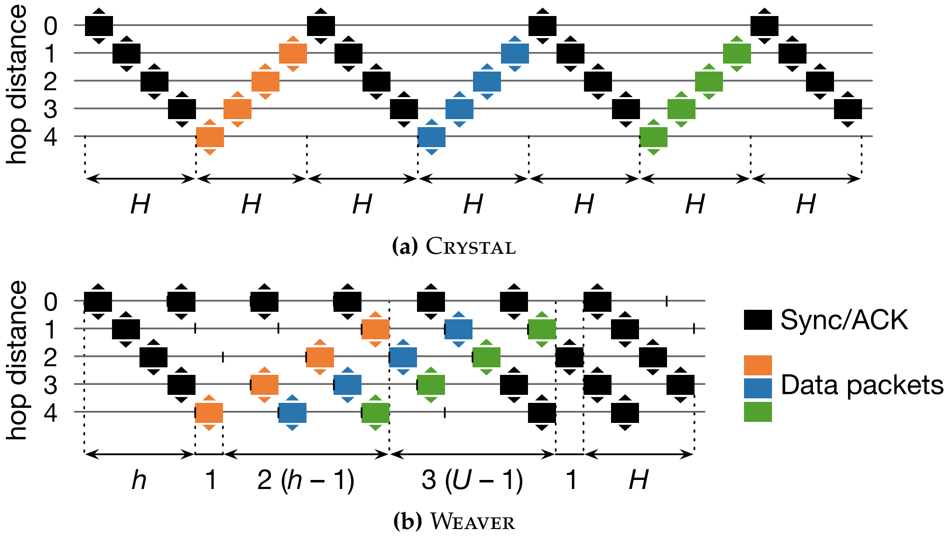
This goal is intrinsically at odds with the classic Glossy schedule alternating a TX slot with a RX one, which causes floods two hops apart to *systematically* compete. A node in RX mode hears TX from both nodes on the next and previous hop towards the sink, with the latter potentially halting propagation of data upstream. For instance, in Figure 7.2a, the TX of the ACK from the sink for the orange packet is performed concurrently with the TX of the blue packet. If the latter prevails, the ACK is lost and the orange packet must be retransmitted, as shown. Otherwise, the ACK prevails and delays the TX of the blue packet. Which one occurs depends on the vagaries of CTX and is therefore unpredictable, ultimately making it impossible to distinguish between data and acknowledgments.

WEAVER replaces the 2-slot TX-RX structure of Glossy with a 3-slot TX-RX-RX one (Figure 7.2b). This simple addition, combined with information gathered during the epoch bootstrap, decouples the TX from nodes at different hop distances, enabling each node to consistently receive *i)* in the first RX slot, data flowing upwards, i.e., from nodes *farther* from the sink and to be forwarded towards it, and *ii)* in the second RX slot, acknowledgments flowing downwards from nodes *closer* to the sink, to be forwarded to initiators.

The effect is clearly visible in Figure 7.2b: *CTX from nodes at different hop distances no longer interfere*. During the first RX slot, a node at hop h can receive only from senders at $h + 1$, nodes at $h + 3$ from senders at $h + 4$, and so on. Receivers then relay data concurrently in their next TX slot, providing forwarding progress. Therefore, different data flows coexists within a single flood and proceed towards the sink in an orderly fashion. If these non-overlapping flows carry packets from different initiators, collection speed increases dramatically, as shown by comparing Figure 7.1b with Figure 7.1a.

The role of (different) acknowledgments. An obstacle remains on the path to sink: packets from same-hop initiators *compete* towards the next hop. The use of CTX ensures that one is decoded at the receiver with high probability, but what happens to the others?

WEAVER solves the problem with two types of ACKs, both piggybacked on data packets whenever possible. A *local acknowledgment* (*L*-ACK) is sent by a receiver to its 1-hop neighbors to (temporarily) suppress their TX for a packet that has already made progress upwards, therefore leaving room for the propagation of other packets still behind. A *global acknowledgment* (*G*-ACK) is instead sent by the sink upon receiving a packet, and re-propagated by each node, informing the whole network that retransmissions are no longer needed for this packet, and it can be discarded.



(a) Crystal

(b) Weaver

Figure 7.3: Latency models for Crystal and Weaver with $U = 3$ initiators (orange, blue, and green) all at maximum hop distance H. Termination phase is not shown.

The two ACKs are implicitly related. A node whose TX is suppressed by a L-ACK should eventually receive a G-ACK; otherwise, the packet has been lost on the way to the sink, and its TX should be resumed. The crucial question then becomes: How long should the node wait before resuming TX? The answer comes, again, from the topology knowledge accrued during the epoch bootstrap that, by informing the node of its hop distance from the sink, enables an accurate estimation of the number of slots expected to elapse between a L-ACK and the corresponding G-ACK for the same packet.

7.2.3 Is It Worth? An Analytical Model

A full understanding of Weaver entails several details (§7.3) whose treatment we postpone to first offer evidence that our strategy achieves significant improvements w.r.t. the state of the art.

We achieve this goal with simplified models for Crystal and Weaver, where we assume that *i)* all data packets originate from U initiators placed at the same hop distance h, whose value $h = H$ is the worst-case maximum distance from the sink *ii)* packet collisions never occur, i.e., one of the packets concurrently transmitted is always received, and *iii)* we do not consider the overhead induced by protocol termination, present in both protocols.

Our models compute the number of slots required to disseminate U data packets under these assumptions, offering a proxy for latency and energy consumption. This allows us to *directly* compare Crystal and Weaver in an abstract setting, eliciting their *intrinsic* differences, independent of the PHY layer or other system factors.

Assuming the most energy-efficient—but least reliable—configuration with a single Glossy retransmission ($N = 1$), Crystal requires

$$L_C = H(2U + 1) \tag{7.1}$$

CTX slots to deliver and acknowledge all U packets. H slots are required for the initial, synchronisation phase (S), followed by one TA pair with $2H$ slots for each of the U initiators.

This is exemplified in Figure 7.3, which also shows how WEAVER significantly increases the parallelism of the U data flows and their ACKs. In the worst-case scenario we consider, the nodes $h = H$ hops away from the sink must wait h slots before they can TX data, to first receive the epoch bootstrap packet from the sink. On the other hand, differently from CRYSTAL, data TX can begin *immediately* after the first successful reception, as in WEAVER CTX occur free from the many constraints of Glossy floods, and dissemination and collection flows are effectively interwoven. The first packet reaches the sink after $1 + 2(h - 1)$ slots, as it takes 2 slots to relay a packet one hop upwards. The remaining $U - 1$ packets reach the sink once every 3 slots, completing after $3(U - 1)$ slots. Finally, the network-wide G-ACKs triggered by the sink upon receipt of each packet account for $1 + H$ slots each, yielding

$$L_W = 3(h + U - 1) + H \tag{7.2}$$

as the total number of slots utilized by WEAVER.

Figure 7.4 compares the protocol performance based on our simplified models, for several values of network diameter H and initiators U. CRYSTAL is more efficient in the (degenerate) case of 1-hop networks, as it uses a 2-slot schedule instead of the 3-slot one adopted by WEAVER and, for the same reason, latency is marginally better with a single initiator ($U = 1$) at $h = H$. However, in a multi-hop network, the number of slots required by CRYSTAL is directly proportional to the network diameter H. This is not the case for WEAVER, which is also faster and more scalable as U increases due to its ability to parallelize flows: up to ~2 times faster for a 4-hop diameter and ~4 times for a 7-hop one, with $U = 30$ (Figure 7.4).

Although the magnitude of the performance gap between the protocols is evident, and sufficient to confirm the validity of our design choices, there are obviously several aspects that are not captured by our simplistic models. Specifically, they do not cater for system and environmental factors affecting reliability and, in turn, latency and energy consumption. These can be ascertained only with real-world experiments, which we present in §7.5 after further detailing our protocol and its implementation.

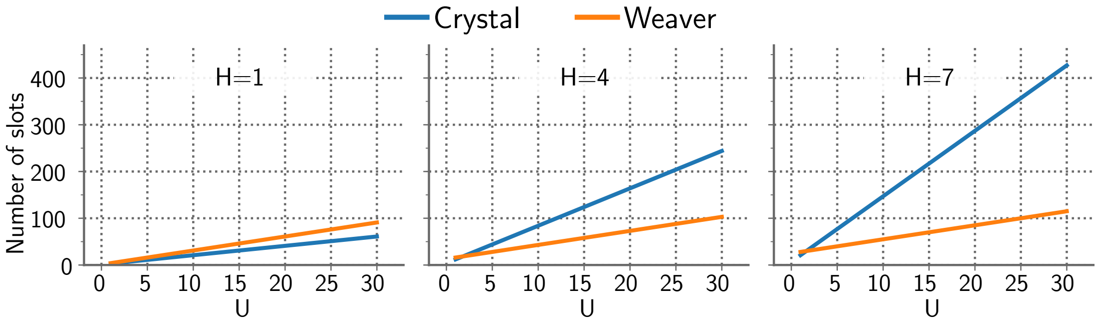


Figure 7.4: Estimated number of slots required to deliver U data packets in a network of H hops.

7.3 Protocol Details

We now complete the description of the Weaver protocol with additional, important details.

Epoch bootstrap. The bootstrap packet sent by the sink and re-propagated throughout the network at the beginning of each epoch is key to provide nodes with a common time reference and topology information. Initiators can *immediately* transmit data inside the re-propagated bootstrap packet; unlike Glossy-based systems, there is no need to separate data collection from the initial synchronization.

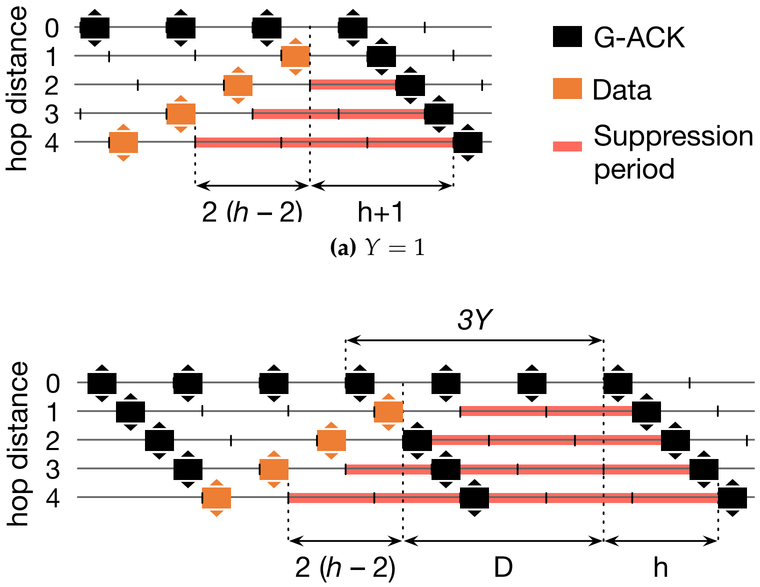
In theory, one bootstrap network-wide flood is enough; in practice, it may not reach all nodes due to collisions. When this happens, functionality is impaired; nodes that missed the bootstrap packet are unaware of their hop distance from the sink and do not know how to realign their schedule, preventing reliable operation. This is less of a problem with several initiators, as the hop distance included in all packets gives nodes that missed the bootstrap multiple chances to realign; however, it is crucial with few initiators.

The nodes that missed the bootstrap packet can reuse the information learned during the previous epoch. Often, this information is unchanged and can be refreshed in the next epoch, if collisions are rare. However, this may be not enough to accommodate the vagaries of wireless communication, or scenarios encompassing mobility. A simple and more reliable solution is to retransmit the bootstrap packet a pre-defined number B of times. We analyze the impact of the value of B on reliability and energy efficiency in §7.5.2.

Local acknowledgments (*L*-ACKs). Upon packet TX, nodes embed the initiator ID of their last heard packet in the Weaver header. When received at another node in the second RX slot, the one devoted to communication from upstream nodes, this ID indicates that the corresponding packet has already made progress towards the sink. Therefore, the original data packet doubles as a *L*-ACK for previous packets at downward nodes waiting to TX the same old data; these nodes can suppress this unnecessary packet TX and replace it with one for a new packet, if any, speeding up propagation of the latter and avoiding unnecessary contention due to the former. Nodes without new data listen during TX slots to hear same-hop neighbors and help them deliver their packets.

Global acknowledgments (*G*-ACKs). The *G*-ACKs sent by the sink contain a bitmap with one bit for each node in the system, signaling whether a packet from the corresponding node has been delivered at the sink during the epoch. *G*-ACKs are interwoven with data collection; they are received in the second RX slot from upward nodes and subsequently propagated downwards in the next TX slot. As with *L*-ACKs, nodes piggyback the *G*-ACK bitmap on data packets, if any, as part of the mandatory Weaver header. Nodes without data to send re-propagate the *G*-ACK as a no-payload packet only if it contains new bits, to reduce contention.

Linking the two ACKs: Suppression period *L*. The reception of a *L*-ACKs by a downward node suppresses the TX of the corresponding packet. Nevertheless, the



(a) $Y = 1$

(b) $Y > 1$

Figure 7.5: Determining the suppression period L.

latter must eventually be acknowledged by the sink via a *G*-ACK; if this does not happen, the packet never reached the sink and dissemination must be resumed. This combination of acknowledgments exploits spatial diversity and, as we verified experimentally, is more reliable than triggering retransmissions only upon a missed *L*-ACK, which is prone to packets remaining stuck in areas with weak links towards parents.

The availability of topology information passively gathered during epoch bootstrap enables an accurate estimation of the number L of slots expected to elapse between the RX of an *L*-ACK for a packet and the *G*-ACK (Figure 7.5a). Indeed, for a *G*-ACK to be sent, the corresponding packet must first be received; for an initiator at h hops from the sink, this requires $2(h-2)$ slots after RX of the *L*-ACK. At the sink, because of the 3-slot scheme, an additional slot elapses between packet reception, in the first RX slot, and the next TX. Finally, in the latter slot the sink disseminates the *G*-ACK, which travels back to the initiator, requiring additional h slots. Therefore, upon receiving a *L*-ACK, a node computes a *suppression period*

$$L = 2(h-2) + h + 1. \tag{7.3}$$

If the suppressed packet is not acknowledged by the sink after L slots, the node resumes its transmission.

Both types of ACK are not immune from packet loss due to collisions, potentially causing wasteful TX that nonetheless rarely affect reliability. A missed *L*-ACK prevents packet suppression. As for *G*-ACKs, when a node receives a packet already known to be acknowledged by the sink, it resumes the piggybacking of the *G*-ACK bitmap, to cater for nodes that may have missed it.

In summary, *i)* *L*-ACKs avoid wasteful retransmissions of packets, hampering the progress of others *ii)* *G*-ACKs achieve the same goal definitely and globally *iii)* to-

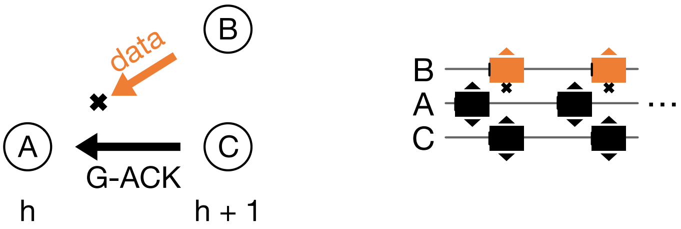


Figure 7.6: Example scenario in which *G*-ACKs block the propagation of data packets.

gether, as determined by L, they avoid that a packet stuck in a "dead end" area is lost and forgotten.

Tuning Weaver: Batching *G*-ACKs. Acknowledgments bring several benefits, but also cause their share of problems.

Consider the example in Figure 7.6. Nodes B and C are at the same hop distance from the sink; their schedule is aligned and their packets, whether containing data or ACKs, compete in the same TX slot. A problem arises if one node enjoys better link quality than the other(s) towards their upward node A, e.g., because C is physically closer to A. In this case, the *G*-ACKs re-broadcast by C are likely to suppress the data TX from B at A, preventing this latter from receiving new, informative packets in lieu of already known *G*-ACKs, and do so repeatedly due to the periodicity of schedules, until *G*-ACK propagation ends.

To counter situations like this, which do occur in practice, we introduce a *batching period* Y for *G*-ACKs at nodes other than the sink. Instead of immediately re-propagating a *G*-ACK upon RX, nodes send a cumulative *G*-ACK once every Y executions of the 3-slot TX-RX-RX pattern, i.e., every $3Y$ slots (Figure 7.5b). When the TX of a *G*-ACK occurs on a node, its bitmap is up-to-date w.r.t. *G*-ACKs received during this period. Therefore, the same information is delivered to the network, but $3(Y-1)$ slots are now free from data/ACK interference like the one in Figure 7.6. However, if data packets are transmitted in the meanwhile, the *G*-ACK bitmap is still piggybacked on them, as this does not cause problems.

As this technique changes the mechanics of *G*-ACK propagation, we revisit the earlier definition (7.3) of the suppression period L as

$$L = 2(h-2) + h + D \tag{7.4}$$

accounting for the additional D slots (Figure 7.5b) introduced by *G*-ACKs batching. As *G*-ACKs are issued with a predefined, globally-known period, nodes can autonomously determine the value of D upon receiving a *L*-ACK for a packet and before its next *G*-ACK.

Termination. Weaver targets fast, reliable, and energy-efficient data collection. This entails quickly turning off the network upon detecting absence of data packets while ensuring that key nodes do not leave before all packets have been delivered to the sink.

Every node terminates and enters low-power mode (sleep) after accumulating in a termination counter T a given number of inactive slots in which no new data is received. RX errors are considered an attempt from neighbors to transmit new information, and reset T. Similarly, the RX of G-ACKs informs a node that the sink is still active and whether it is aware of the data the node already transmitted; if it is not, the node postpones its termination.

The value of T depends on the protocol phase. During the epoch bootstrap, $T = 3H + 3B$, where B is the number of bootstrap packets sent and H the maximum hop distance of nodes from the sink. Indeed, *i)* H slots are required for the bootstrap packet to reach the farthest possible initiator and enable its packet TX; *ii)* $2H$ slots are required, due to the 3-slot scheme, for a packet from this worst-case initiator to reach the sink; however, *iii)* this packet competes with the B bootstrap packets rebroadcast by neighbors in consecutive TX slots; therefore, in the worst case where these are always received upstream instead of the packet, additional $3B$ slots must elapse.

If the sink does not receive any data packet T slots after sending the bootstrap packet, it enters sleep. Otherwise, an alternate counter is defined and reset every time a data packet is received. The sink waits $T = 3H + 3$ slots at the end of every G-ACK batching period. Similar to the above, *i)* H slots are required for the G-ACK to propagate downwards and enable the TX of a new packet, *ii)* $2H$ slots are required to collect it at the sink, plus *iii)* 3 slots to account for the worst case where the re-propagation of the G-ACK by a same-distance neighbor blocks the data TX. The suppression period L after a L-ACKs was defined precisely to allow transmissions to resume timely for packets that did not receive the G-ACK, giving them another chance to reach the sink before termination.

Once the sink decides to terminate, it floods a special packet to shutdown the entire network before entering sleep; there is no point in keeping nodes awake if the sink is not. However, nodes are also capable of entering sleep autonomously, as they maintain the same termination counter T as the sink; this serves as a fallback ensuring node termination when the shutdown packet is lost.

7.4 A Modular Implementation

WEAVER relies on the ability to individually manage CTX. This, however, involves low-level radio programming, time slot management, and synchronization, i.e., tedious and repetitive work that complicates and distracts from the high-level protocol logic and, worse, must be largely modified when the latter changes.

To simplify our iterative development and enable other researchers to build their own protocols atop fine-grained CTX (§7.6), fostering the widespread adoption of this technique, and in turn of UWB communication at large, we designed our prototype to sharply separate these two layers. We implement the low-level functionality necessary to CTX in a generic and reusable way, available to protocol designers via a simple yet expressive API (§7.4.1): the Time Slot Manager (TSM). The actual WEAVER protocol is implemented as a thin veneer atop it, easily replaceable and modifiable. The architecture of our prototype (Figure 7.7), implemented on Contiki [107], is com-

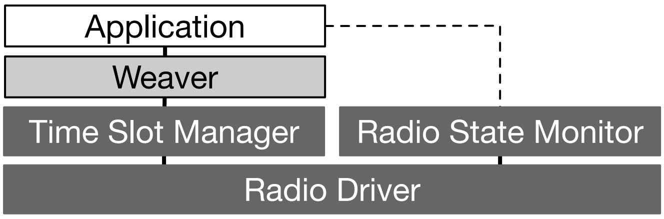


Figure 7.7: System architecture.

pleted by an optional module enabling accurate estimation of energy consumption (§7.4.2).

7.4.1 Time Slot Manager: A Flexible CTX Engine

Our goal is to avoid complexity when implementing simple things while giving fine-grained control to the higher layer, when needed.

Enabling factors. The decoupling of protocol logic is enabled by new capabilities of modern radio chips, allowing access to internal high-precision timers for timestamping radio RX events and scheduling TX/RX operations at specified times, as detailed in Chapter 5 for the DW1000 chip we rely on in this work.

Without these capabilities, meeting the strict timing requirements of CTX forced protocols to trigger the next action right within the handler of the previous radio event and keep the duration of event processing constant for all nodes, packets, and protocol states to guarantee that all nodes trigger the next operation at the same time. This approach limited code branching and therefore the complexity of the higher-level protocol logic. Instead, if the next operation is scheduled directly via the internal timer of the radio, autonomously from the MCU, the protocol logic can become more rich and dynamic, expanding through levels of abstraction and indirection. The only requirement is that event processing finishes within the predefined deadline, leaving enough time for the radio to initialize and perform the next operation at the scheduled time.

Basic principles. We observe that all CTX systems (§7.1) share the same time structure. They organise communication in rounds (epochs), placing the radio to sleep between them. Each round consists of multiple fixed-duration time slots associated with a TX or RX operation (sometimes neither) and related data processing; the number of slots per round may vary. The transmitted messages contain the current slot index within the round, enabling receivers to establish the round reference time (i.e., its beginning) from the start-of-frame delimiter (SFD) timestamp of the received packet.

We delegate to TSM all this common bookkeeping related to synchronization, i.e., computing the round reference time, executing radio operations at the right times, and updating the synchronization information in the header of TX packets to allow the reference time to propagate over multiple hops. Instead, we leave it to the higher layer (e.g., Weaver) to decide what action (TX, RX, or none) to perform in each slot, the data payload for each TX slot, and when to stop the current round and enter sleep until the next begins.

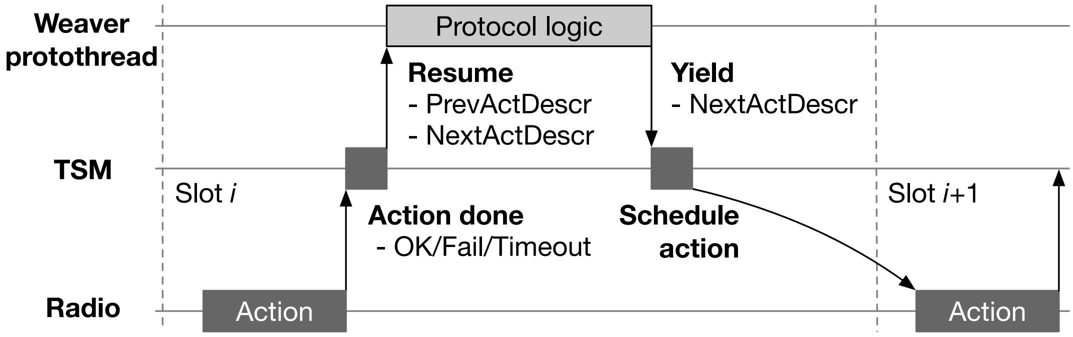


Figure 7.8: Flow of control.

```
while(1) {
  TSM_SCAN(buf); // scan until RX
  if (prevActDescr.status == SUCCESS) {
    // forward the packet in the next slot
    TSM_TX(buf, prevActDescr.data_len);
    TSM_RESET(ROUND_PERIOD); // skip to the next round
}}
```

Figure 7.9: Glossy forwarder logic ($N = 1$) atop TSM.

A node joins the network via the TSM_SCAN operation, instructing TSM to listen to the channel until a packet is received. When this happens, TSM automatically synchronizes with the network and starts the slotted operation. Optionally, the higher layer can instruct TSM to adjust its reference time upon any successful RX; it is wise to do so periodically, to counter clock drift. Unlike Glossy, TSM is agnostic of node roles, leaving it to the higher layer to determine which node(s) provide the authoritative time reference, but provides all the necessary time calculations, adjustments, and scheduling.

API and control flow. A protocol built atop TSM begins with a TSM_START call providing the desired slot duration and a pointer to the slot handler function. The control flow is then driven by TSM, which automatically calls the latter function before every slot (Figure 7.8), passing as a parameter a special read-only structure describing the operation performed in the *previous* slot, if any, including the code of the action performed, its status (success or error code), the RX payload data and size (if any), and the slot index.

Another structure describing the *next* slot action is passed by reference, to be filled by the slot handler function. TSM pre-configures most fields with default values based on settings and context; only few must be set by the protocol. This next-slot structure includes the action to perform (SCAN,RX,TX,RESTART,STOP), a pointer to the TX payload or RX buffer, and other fields described later. After the slot handler function ends, control returns to TSM which uses the values set in this structure to schedule the next action.

In principle, a conventional function can be used as slot handler. However, TSM was designed to take full advantage of Contiki protothreads by providing convenience calls (C macros) that combine the configuration of the next action with protothread interaction, effectively mimicking a conventional blocking function (Figure 7.8). These convenience calls yield control to the system, letting the MCU perform other tasks or go to a low-power mode while waiting for the requested action to complete; when this occurs, the protothread is resumed from the point where it requested the TSM action. This enables the description of a complex protocol logic as a sequential program with branching and loops, arguably more natural than the cumbersome event-driven style necessary with classic callbacks.

Figure 7.9 shows a naïve yet working implementation of the Glossy forwarder logic with $N = 1$, written atop TSM in only 6 lines of code. A full-blown Glossy re-implementation is outside the scope of this paper; the code is meant to illustrate the simplicity induced by TSM, fully exploited in our WEAVER prototype. Each loop iteration is a Glossy round. The forwarder, i.e., any node other than the initiator, begins each round by scanning the channel for incoming packets. When one arrives, TSM automatically uses it to resume the protothread and begin slotted operation; in case of successful RX the node retransmits the exact same payload in the next slot via the `TSM_TX` call. TSM advances the slot index automatically and updates the TX packet header accordingly. After packet TX is finished, the `TSM_RESET` call instructs TSM to finalize the current round and sleep for the rest of the specified `ROUND_PERIOD`.

This example shows how TSM keeps simple things simple, by hiding all operations related to timing and slot scheduling, and allowing the higher layer to concentrate on the protocol logic. On the other hand, the fact that WEAVER, a significantly more complex protocol, was implemented atop TSM confirms that the abstractions in TSM are not only simple but also expressive.

Delayed TX. As reported in [81], CTX perform significantly better in UWB if they are slightly de-synchronized, unlike in narrowband. TSM caters for this by allowing the definition of a small TX delay (ns to μs) on a per-slot basis. This requires adding the value of the delay used to the nominal slot reference in the TSM header, enabling receiving devices to compensate the delay when computing the round time reference. WEAVER exploits this feature by inserting a random delay before all TX, specified in the corresponding field of the next-action structure. Delay values are reported in §7.5.1.

RX timeouts and energy savings. Idle listening (preamble hunt) is the most energy-consuming operation of the DW1000 [54]. Therefore, we minimize the time the radio listens to the channel in RX slots. Since we expect nodes to be synchronized, the

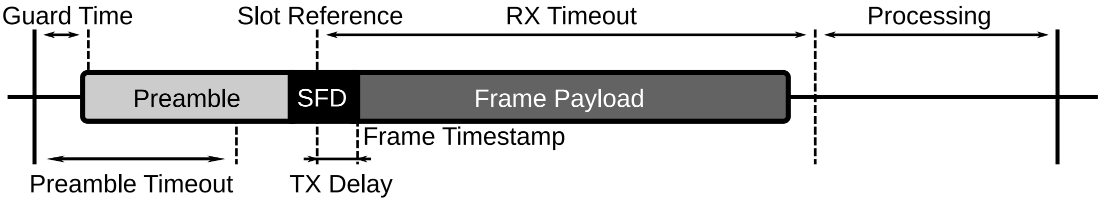


Figure 7.10: TSM slot structure.

radio can begin listening right before we expect the frame preamble to arrive, and stop shortly after if none is received. We achieve this by setting a preamble detection timeout equal to the sum of *i)* the initial guard time, *ii)* the maximum TX delay senders could use, and *iii)* half the preamble duration. If no preamble symbols are detected before timeout, the radio switches to idle mode automatically and triggers an event to TSM. Moreover, we set a frame RX timeout to guarantee that any RX operation leaves enough time within the slot for the protocol layer to run its logic and prepare for the next slot. Figure 7.10 shows the resulting structure of a RX slot. In case a timeout triggers, TSM reports a failed RX action to the higher layer.

7.4.2 Monitoring Energy Consumption

An accurate estimation of energy consumption is crucial to validate the performance of our prototype. Systems built atop Contiki for IEEE 802.15.4 narrowband can rely on the well-known Energest [174] component. Unfortunately, no equivalent exists for UWB.

Therefore, we designed our own component to estimate the energy spent during radio operations. Our Radio State Monitor module (Figure 7.7) brings the core concepts of Energest to the more complex state machine of the DW1000 radio. This entails supporting several key features not present in Energest, e.g., delayed operations and timeouts, and using the precise timer of the radio. As in Energest, our module maintains several counters aggregating the overall time spent by the radio in the various states. However, differently from it, our module tracks separately different portions of the frame RX and TX for more accurate estimation of the energy spent, as these consume very different amounts of energy on the DW1000. Finally, the current drawn in idle mode is also accounted for.

Overall, the Radio State Monitor is a valuable contribution per se, not tied to CTX, that can be exploited by other researchers working on UWB at large to assess the energy consumption of their systems.

7.5 Evaluation

We evaluate WEAVER in a UWB testbed at our premises, considering two network topologies with different characteristics (§7.5.1). We first provide an in-depth analysis of parameters B and Y, which control the reliability of the epoch bootstrap and the periodicity of G-ACK dissemination, and analyse their impact on performance (§7.5.2). We then compare WEAVER to the UWB implementation of CRYSTAL (Chapter 6, [2]) in the same conditions (§7.5.3). For both protocols we report *i)* the packet delivery rate (*PDR*) at the sink, *ii)* the per-epoch estimated energy consumption of non-sink nodes, and *iii)* the latency, defined as the time between the beginning of an epoch and the delivery of the *last* data packet at the sink. Finally, in §7.5.4 we experiment with mobile nodes to assess whether WEAVER is suitable for use in dynamic topologies.

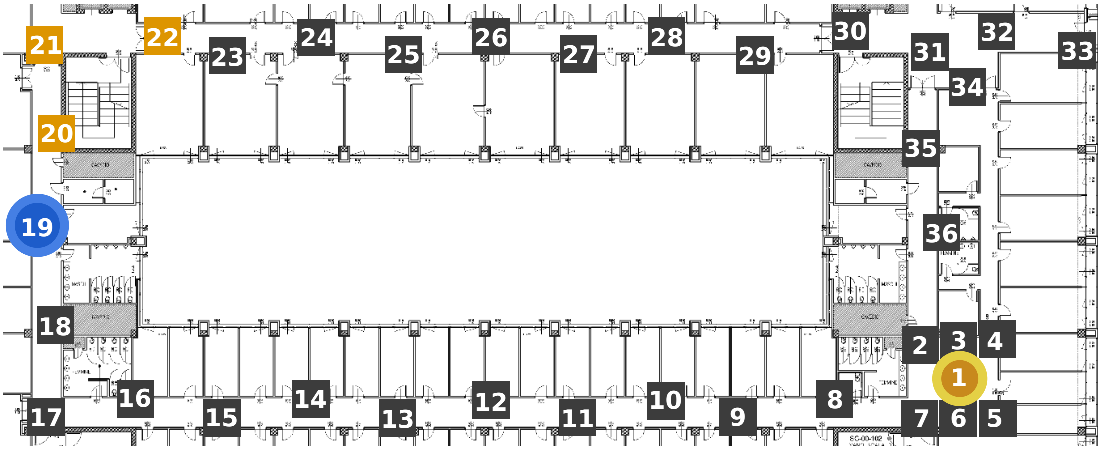


Figure 7.11: Testbed spanning 84 × 33m^2. In Floor, node 1 is the sink. Linear excludes node 20–22; node 19 is the sink.

7.5.1 Experimental Setup

Hardware and testbed. We report experiments from a 36-node testbed installed on the ceiling above the corridors of an office building, over a 84 × 33 m^2 area (Figure 7.11). Each node includes a Raspberry Pi 3 Model B+, a JTAG programmer, and a DecaWave EVB1000 board equipped with a DW1000 UWB radio and a STM32F105 ARM Cortex M3 MCU. A dedicated Ethernet infrastructure enables automated and remote control of experiments and collection of logs.

Network topologies. We consider two topologies, called Floor and Linear, with different characteristics. In Floor, node 1 is designated as the sink, all nodes are active, and the network spans 3 hops. Data can flow along two paths—clockwise and counter-clockwise—providing spatial diversity. The sink is deployed in a dense area where 10 neighbors have near-perfect link quality towards it, and most (nodes 2–7) are placed at similar distances from it. As reported in [81], a similar scenario can be challenging for CTX-based protocols, and therefore intriguing to analyze, since multiple signals with similar strengths and timing are likely to reach the sink, increasing collisions' probability, especially with several and different packets.

In Linear, node 19 acts as the sink and nodes 20–22 (top left corner) are disabled, preventing communication between the sink and node 23. This *i)* increases the maximum hop distance to 6 hops, and *ii)* forces all data flows to proceed along a single path, significantly reducing spatial diversity. Moreover, node 18 cannot communicate with any of nodes 8–16 on the bottom corridor; therefore, node 17 is the *only* connection between the sink and the remaining (large) part of the network. The absence of receiver redundancy, known to be detrimental for CTX-based protocols, and proven to affect UWB CTX performance in §6.3.3, makes this topology particularly challenging, yet realistic in indoor environments.

Radio configuration. We use channel 4 with 64 MHz pulse repetition frequency (*PRF*). To minimize energy consumption and reduce latency, we choose the highest 6.8 Mbps data rate on the DW1000 and the shortest ∼64 μs preamble, in line with the configuration adopted in the previous chapter. We set the TX power to the maximum

Table 7.1: Occurrence of failed bootstrap for any node, and average energy consumption vs. number B of bootstrap packet retransmissions. Data acquired over $10,000$ epochs in the two topologies with no initiator ($U = 0$).

	% of failed bootstraps			**Energy (mJ)**		
Topology	B=1	B=2	B=3	B=1	B=2	B=3
FLOOR	0.015	0.0003	0	2.49	3.00	3.35
LINEAR	0.0016	0.0006	0	3.19	3.46	3.73

recommended [55] for our channel and *PRF*. We exploit TSM to randomly delay all transmissions by up to 1 μs (i.e., roughly the duration of one preamble symbol) as this small de-synchronization significantly reduces the chance of collision in UWB [81].

Packet size and slot duration. Long packets are known to increase the chance of collision when transmitted concurrently [29, 168]; this holds even with UWB communication, as ascertained in Chapter 6. To assess how the packet size impacts reliability and energy consumption, we perform experiments with both short (2 bytes) and long (100 bytes) payloads. We set the duration of WEAVER slots to 813 μs, enough to accommodate the maximum IEEE 802.15.4 frame length.

7.5.2 Dissecting Weaver

We study the impact of parameters B and Y on the performance of WEAVER. The former impacts the reliability of epoch bootstrap, while the latter controls the trade-off between latency and energy consumption depending on the expected traffic patterns.

Reliability of epoch bootstrap. We explored $B \in \{1..3\}$. Table 7.1 reports the number of failed epoch bootstrap attempts across $10,000$ epochs, with no initiators ($U = 0$). In FLOOR, $B = 1$ yields 59 occurrences of a node missing the bootstrap packet (0.015%), while $B = 2$ yields only 1 occurrence (0.0003%). LINEAR is less prone to a failed bootstrap, with the same values of B yielding only 5 and 2 occurrences (0.0016% and 0.0006%), respectively.

The value $B = 3$ guarantees a correct bootstrap in all epochs for both topologies. However, this reliability comes at the cost of energy consumption (Table 7.1) whose increase is more evident without traffic ($U = 0$) as node termination directly depends on B (§7.3). In this case, consumption is 3.35 mJ and 3.73 mJ in FLOOR and LINEAR, a $+35\%$ and $+17\%$ increase w.r.t. $B = 1$. However, when traffic is present ($U > 0$), the influence of B is less marked as *i)* the network remains awake for longer to collect all data, and *ii)* collection starts immediately, in parallel with bootstrap. Hereafter, we set $B = 2$, the best compromise between reliability and energy efficiency.

Impact of G-ACK batching period. The period Y used to disseminate G-ACKs upon data reaching the sink is the main knob to control WEAVER, balancing timeliness in acknowledging packets via G-ACKs with their interference with data (§7.3). We analyze the impact of Y on the duration of the flood (Figure 7.12). For each combination of topology, packet size, $Y \in \{1..9\}$, and $U \in \{1, 10, 30\}$ the results are obtained by aggregating 1000 epochs.

The impact of Y on termination, while not high in relative terms, varies in function of the amount of traffic (Figure 7.12). In both topologies, WEAVER shows a similar response to the increase of Y, although the trend is more evident in FLOOR. Similarly, packet size does not have a substantial impact, with longer packets causing only a slight increase in latency. With sparse traffic, increasing Y does not yield benefits as G-ACKs rarely interfere with data floods, making the duration of the collection phase independent from Y. Thus, $Y = 1$ is the fastest and the most energy-efficient solution, as it minimizes the time a node waits in between the last packet RX and termination (§7.3). However, as the number U of initiators increases, a small value of Y becomes detrimental; with $Y = 1$, each packet reaching the sink triggers a new G-ACK, disseminated network-wide. This increases contention and the risk of interference between data and G-ACKs, slowing down the collection process. By increasing Y and therefore reducing the number of G-ACKs, we increase the chance to collect multiple packets in between two consecutive G-ACKs floods. The impact on the flood duration is clearly visible for $U = 30$. For instance, in FLOOR and with short packets a flood requires 174 slots to terminate with $Y = 1$, and only 143 with $Y = 4$ (18% reduction). On the other hand, increasing Y further does not pay off, as it forces the system to remain active for several slots after the last packet collected; this is very costly with sparse traffic and brings little to no improvement with a denser one.

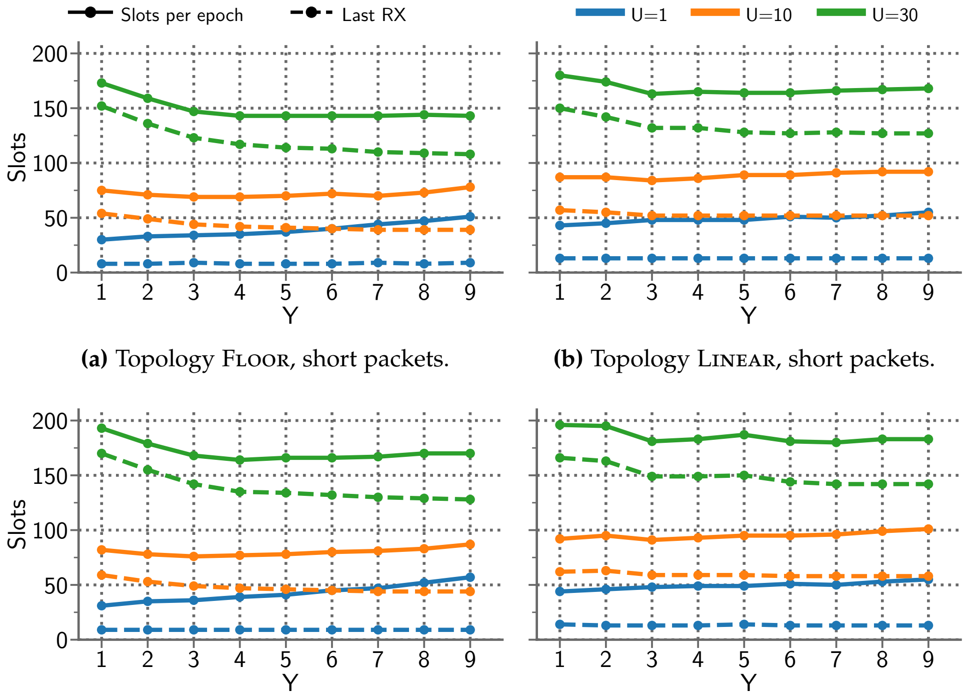
The best choice of Y ultimately depends on the behavior of initiators. If the traffic profile is known beforehand, users can tune the value of Y to further reduce the latency and energy consumption of WEAVER. Otherwise, Figure 7.12 shows that the impact is relatively limited anyway. In the rest of this section, we assume the application has *no a priori knowledge* of traffic and set $Y = 4$, as, in our case, this is a good balance across all dimensions.

Finally, WEAVER achieved $PDR \geq 99.9\%$ independently from the value of Y. A more thorough study of reliability is described next.

7.5.3 Weaver vs. Crystal

We compare against CRYSTAL [1, 31], a state-of-the-art data collection protocol, using its publicly-available implementation for UWB [60] that we presented in the previous chapter.

Protocol configurations. We configure WEAVER with $B = 2$ bootstrap packet retransmissions and a G-ACK batching period $Y = 4$, informed by our analysis in §7.5.2. Configuring CRYSTAL entails tuning the underlying Glossy for every phase, by defining the flood redundancy factor N and adapting the maximum phase duration W accordingly. Large values of N enhance flood reliability, by increasing the spatio-temporal redundancy of Glossy, but also increase energy consumption. In our analysis we consider $N \in \{1, 2\}$, exploring different trade-offs between reliability and energy efficiency. $N = 1$ is the most energy-savvy configuration possible, but also



(a) Topology Floor, short packets.

(b) Topology Linear, short packets.

(c) Topology Floor, long packets.

(d) Topology Linear, long packets.

Figure 7.12: *G*-ACK batching period Y vs. number of slots required for termination and last packet collected at sink.

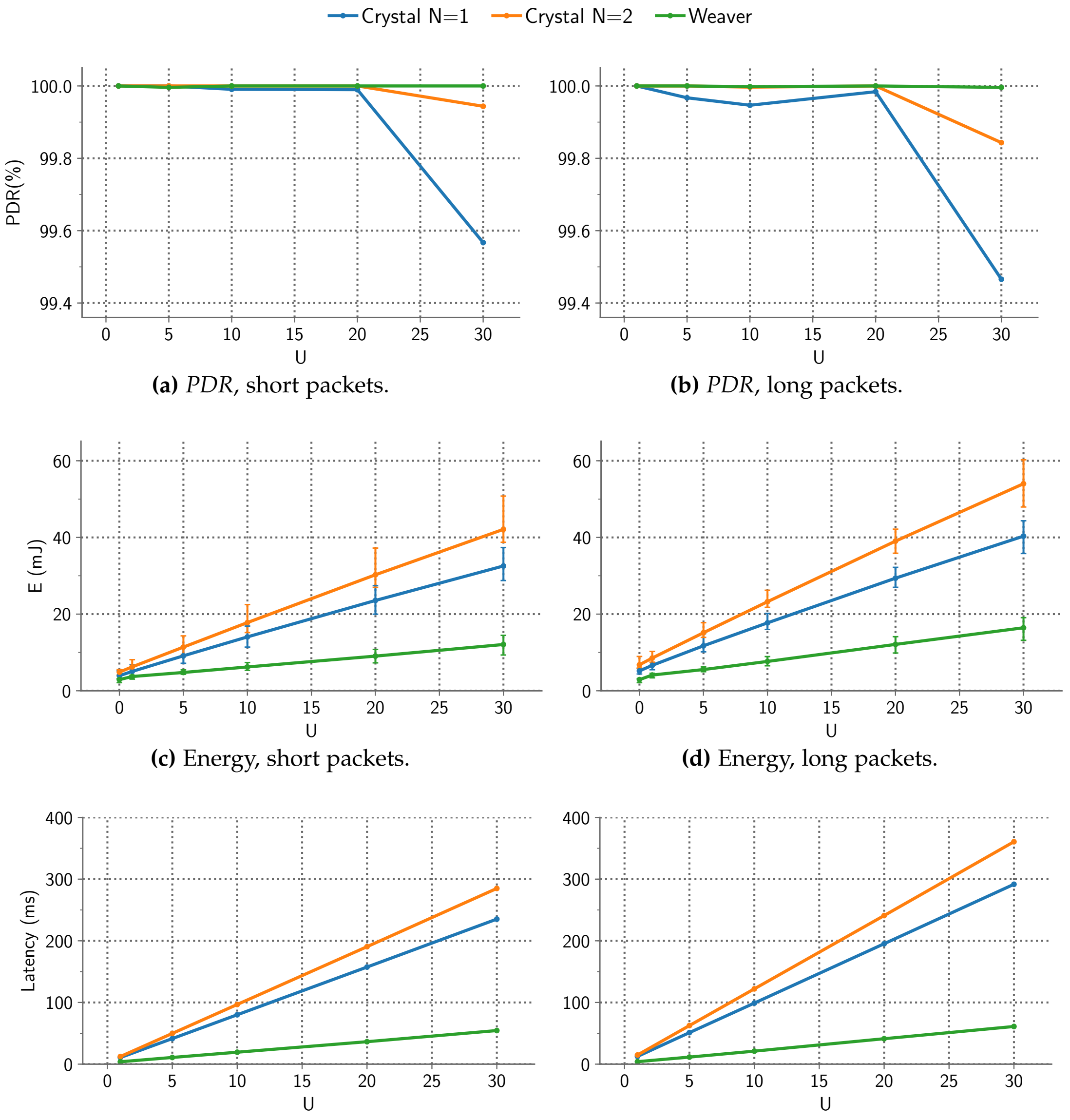
the most fragile. Table 7.2 reports a summary of the configurations. Following the methodology of [31], we dimension W for each phase to accommodate the maximum hop count H and the desired number of packet retransmissions N, plus a small slack to cope with possible flood delays due to collisions. Other Crystal parameters (e.g., number of empty TA pairs before termination) are unchanged w.r.t. [2, 31].

Results. For each combination of topology, number of initiators $U \in \{0, 1, 5, 10, 20, 30\}$, packet size, and protocol configuration, we collect execution traces of 5,000 epochs for both protocols.

In Floor, Weaver is largely unaffected by packet size, achieving near-perfect reliabil-

Table 7.2: Parameters used for the two configurations of Crystal considered. T_s and T_l are the duration of the T phase optimized for a short (2 bytes) and long (100 bytes) packet, respectively.

	N	***W* (ms)**			
Topology	S, T, A	S	A	T_s	T_l
Floor	1	2.7	2.8	2.8	4.5
	2	3.6	3.7	3.6	6.1
Linear	1	4.0	4.1	4.0	6.0
	2	4.8	5.0	4.9	8.4



(a) *PDR*, short packets.

(b) *PDR*, long packets.

(c) Energy, short packets.

(d) Energy, long packets.

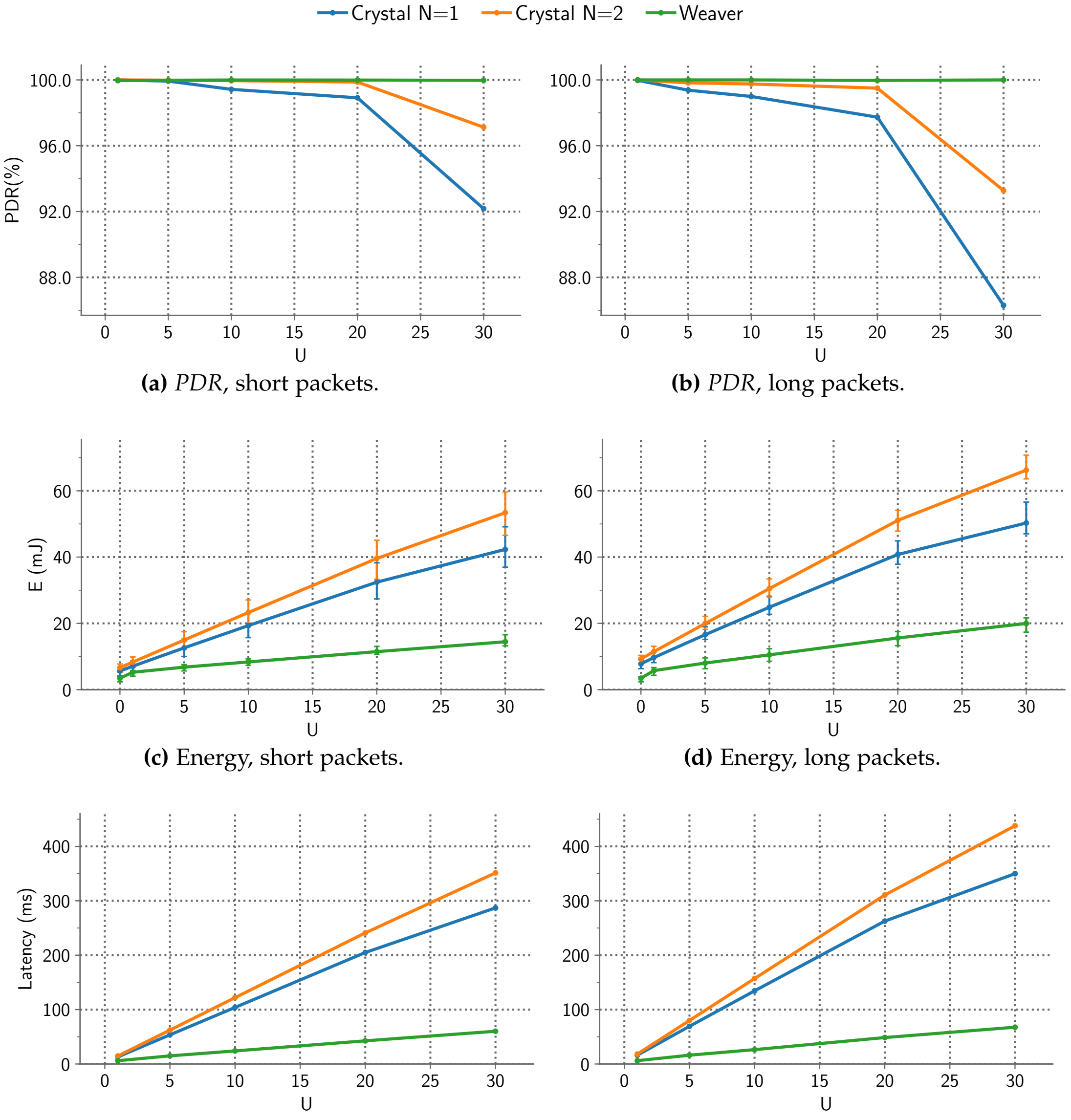
(e) Latency, short packets.

(f) Latency, long packets.

Figure 7.13: WEAVER vs. CRYSTAL in the FLOOR topology.

ity with both short and long ones (Figure 7.13a, 7.13b) even under heavy contention, with $PDR > 99.99\%$ when $U = 30$. The reliability of CRYSTAL—although very high—is significantly lower, especially with $N = 1$ and traffic bursts. Furthermore, the negative impact of long packets is clearly visible as U increases, due to the higher chance of collisions.

We actually found an increased rate of collisions for long packets also in WEAVER, by analyzing the RX error rate at the level of single slots. For instance, with $U = 30$ each node incurs in a RX error 5.83 times per epoch with short packets and 9.07 with long ones. However, WEAVER can tolerate more collisions, as the continuous flood grants each node many chances to retransmit. The number of retransmissions is not fixed beforehand, as in Glossy and therefore CRYSTAL, rather WEAVER's flood termination policy is adaptive to data traffic and RX errors (§7.3), enabling nodes to keep attempting to forward packets more times upon collisions. Moreover, thanks to the L-ACKs, WEAVER can promptly suppress transmissions before the arrival of



(a) *PDR*, short packets.

(b) *PDR*, long packets.

(c) Energy, short packets.

(d) Energy, long packets.

(e) Latency, short packets.

(f) Latency, long packets.

Figure 7.14: Weaver vs. Crystal in the Linear topology.

G-ACKs from the sink, quickly reducing contention.

High reliability often comes with extra energy consumption. This is not the case for Weaver, specifically designed to remove the inefficiencies of Glossy-based solutions. Indeed, the fast, reliable and contention-resilient operation of Weaver yields significant energy improvements w.r.t. Crystal (Figure 7.13c, 7.13d).

Without traffic ($U = 0$), Weaver consumes 40% and 57% less than Crystal with short and long packets, respectively. The benefits of fine-grained control over CTX increase with U, as Weaver fully unleashes its ability to parallelize collection floods, *reducing energy consumption by ~70% for $U = 30$ initiators regardless of packet size.*

In the Floor topology explored so far, multiple paths enable packets to reach the sink, itself surrounded by many relays. In Linear, instead, all data towards the sink must flow through the bottleneck of node 17; continued collisions at this node can lead to interruption of the flood and multiple packet losses. This is particularly true for CTX-based systems, as analysed in §6.3.3. CRYSTAL behaves quite poorly in these conditions; even with the more reliable $N = 2$, *PDR* decreases as U increases, down to ∼97% and ∼93% for $U = 30$ and short and long packets, respectively (Figure 7.14a, 7.14b). WEAVER is affected to a much smaller extent, achieving $PDR > 99.9\%$ in all conditions. Increasing N brings CRYSTAL *PDR* back to much higher values, as demonstrated in Chapter 3 and 6; we don't explore this opportunity here, as it unavoidably affects CRYSTAL latency and energy efficiency, both already worst than WEAVER as discussed next.

Energy consumption increases for both protocols in LINEAR, due to the larger diameter. However, WEAVER consumes a fraction of the energy required by CRYSTAL (Figure 7.14c, 7.14d) similar to what observed in FLOOR; with $U = 0$, WEAVER saves 40% and 63% with short and long packets, and ∼70% for both packet sizes with $U = 30$.

WEAVER is highly reliable and energy efficient in both our topologies, despite LINEAR being quite challenging. The question is whether it is also faster, as predicted by our model (§7.2.3). Many definitions of latency are possible. We report the time needed to complete data collection (i.e., RX of last packet) because *i)* the average latency incurred by a packet is roughly half the duration of collection, and *ii)* flood termination at the sink happens consistently a few slots after the RX of the last packet, making these two metrics redundant.

Our experiments confirm that WEAVER is significantly faster than Crystal. With long packets and $U = 30$, CRYSTAL receives the last packet after 361 ms in FLOOR, and 438 ms in LINEAR, while WEAVER does the same in only 109 ms and 121 ms, respectively (Figure 7.13f, 7.14f). Interestingly, switching from FLOOR to LINEAR causes a 21% latency increase for CRYSTAL, but only 11% for WEAVER. Even with a single packet ($U = 1$) WEAVER is faster at 7 ms and 11 ms, against the 15 ms and 18 ms of CRYSTAL, also thanks to the ability to begin packet TX concurrently with the initial bootstrap (§7.3). As for short packets, WEAVER is similarly faster than CRYSTAL in both topologies (Figure 7.13e, 7.14e). For $U = 30$, its latency is 97 ms and 107 ms, against 285 ms and 351 ms for CRYSTAL. For $U = 1$, WEAVER incurs a latency similar to long packets, while the one of CRYSTAL, reduced to 12 ms and 15 ms, remains higher than WEAVER.

Optimizing the fixed slot duration (§7.5.1) for packet size leads to even larger improvements. For 2 bytes packets, a shorter slot of 455 μs reduces latency and energy consumption respectively by 43% and 22% w.r.t. Figure 7.13–7.14, regardless of traffic and without affecting reliability.

7.5.4 Weaver and Mobility

Among the advantages of CTX-based protocols is that they are agnostic of the underlying network (Chapter 2); being resilient to topology changes they are suitable for scenarios with mobile nodes [30]. However, this is not entirely true for WEAVER. Nodes learn their hop distance during the epoch bootstrap, and leverage this infor-

Table 7.3: Performance of Weaver with 3 mobile nodes.

	PDR(%)		**Latency (ms)**		**Energy (mJ)**	
	U=10	U=30	U=10	U=30	U=10	U=30
Static sink	100	99.99	57.72	119.51	7.58	14.07
Mobile sink	100	99.95	61.79	123.58	8.22	15.24

mation to direct data and *G*-ACKs flows during collection; this potentially makes the protocol susceptible to topology changes.

Nevertheless, Weaver completes the collection of packets from 30 initiators over a 6-hop network in about 100 ms (Figure 7.14e). During this time, a person walking covers nearly 14 cm and a car traveling at 100 km/h moves of roughly 3 m. Therefore, a Weaver flood is so fast that even when nodes are moving the topology inside it remains essentially static.

We ascertain whether this is true, and the applicability of Weaver to mobile scenarios, through experiments in which 3 people, each carrying a node, walk at brisk pace in the testbed area for the entire duration of the test. In these experiments, all 39 nodes of the testbed are active. As mobile nodes traverse the testbed, their links to other nodes degrade or even interrupt abruptly due to obstacles.

Table 7.3 shows *PDR*, latency, and energy consumption with a static or mobile sink. In the first scenario, node 1 is the sink, as in Floor, and all mobile nodes are initiators; in the second, one of them serves as mobile sink. The latter scenario is particularly challenging, as sink movement *i)* alters the structure of the collection scheme, and *ii)* explores several topologies at once, including problematic ones like Linear. We run $2,000$ epochs for each $U \in \{10, 30\}$ and short packets, and observed no packet loss with $U = 10$ and $PDR > 99.9\%$ with $U = 30$, regardless of sink mobility.

Overall, the values in Table 7.3 are in line with those in §7.5.3; mobility appears to have little to no impact on performance. These results provide a preliminary confirmation of Weaver robustness under mobility, which we will assess through a more extensive evaluation as part of our future work.

7.6 Discussion and Outlook

We concisely elaborate on the potential impact of our work and how it could be extended and generalized by other researchers.

What did we accomplish? The evaluation we presented, along with the analytical model in §7.2.3, confirm that protocols based on fine-grained CTX rather than monolithic Glossy floods can unlock remarkable improvements over the already impressive performance achieved by the latter. The ability to weave and consolidate multiple floods into a single, coordinated one improves on latency, but also on reliability and energy consumption—i.e., all three metrics in which CTX excel. On the other hand, the very small latency also enables a novel way to exploit topology information, allowing protocol designers to treat the network as static even when it is not, as in scenarios encompassing mobility. We argue that these design principles are a contribution per se, which goes beyond the nonetheless remarkable performance of Weaver, and can inspire further research on the topic.

What about other radios? Although we focused on UWB, we argue that our contribution is not limited to it, as neither Weaver nor TSM rely on features specific to the PHY or radio chip we used.

The superior performance of Weaver is intrinsically determined by its use of fine-grained CTX, as shown quantitatively in §7.2.3. Indeed, the efficient organization of multiple data flows in Weaver builds solely on the assumption that receivers can successfully decode, with high probability, one among different packets transmitted concurrently. As discussed in Chapter 2, this assumption has been shown to hold for other popular PHY layers besides UWB. Therefore, we expect the principles of Weaver, if not the exact protocol, to find direct application for these other radio technologies.

However, the extent to which our quantitative findings can be transferred to other radios is yet to be established experimentally, for which TSM provides a handy framework. We argue that it is simple to port TSM to any platform that, like DW1000, supports timestamping and scheduling of packet TX and RX precisely enough to enable non-destructive interference of TX signals. A short-term item on our research agenda is to port TSM and Weaver to a modern IEEE 802.15.4 narrowband radio supporting these features, e.g., the CC2538 for which Contiki-based implementations of Glossy already exist [144], further simplifying the transfer of our results.

What about other traffic patterns? The role of TSM, however, is not limited to simplifying the transfer of our results to other platforms. On the contrary, our main motivation for its development was to sharply separate the *general* low-level mechanics of CTX from the *specific* higher-layer protocol exploiting them.

In this respect, Weaver is only one of the possibilities, geared towards data collection. We argue that the benefits unlocked by the key insight of Weaver, i.e., its fine-grained use of individual CTX instead of monolithic floods, can be reaped for other traffic patterns similar to what happened for Glossy, whose availability as a core communication primitive was exploited in many directions.

As discussed further in the next chapter, it is on top of our research plans to leverage the flexibility and power of individual CTX, underpinned by the capability of our TSM kernel to easily and efficiently orchestrate them, for co-designing novel networking stacks targeting industrial wireless control. Combining communication primitives based on fine-grained CTX with aperiodic control strategies could enhance the reactivity, dependability, and energy efficiency of WCSs that, while enjoying all the benefits concurrent transmissions provide, are free from the overhead and the rigidity of relying on predefined, fixed-structure Glossy floods.

What about ranging and localization? Our focus on UWB opens intriguing opportunities. For instance, the work in [165] has recently shown that CTX in UWB enable concurrent distance estimation (ranging) towards multiple nodes at once, inspiring several follow-up works [166, 175, 176, 177]. The concepts in Weaver, and the core building blocks in TSM, could therefore be exploited to rejoin the two perspectives of communication and localization provided by UWB under a single framework efficiently enabling both.

7.7 Conclusions

CTX have been studied for about a decade, but largely within the perimeter of what enabled by the popular Glossy system. In this chapter, we show that an alternate design mindset is possible; one where the protocol designer regains control over all degrees of freedom available once using individual CTX as a primary building blocks, significantly finer-grained than the monolithic one offered by Glossy. We offer analytical and experimental evidence that this alternate design paradigm brings remarkable advantages in the context of convergecast, and provide publicly-available, open-source software [60] enabling researchers to explore other ways to harvest its benefits.

8

Conclusions and Outlook

Societal and market demands are accelerating the need for low-power wireless solutions that are at the same time deterministic yet flexible, and satisfy stringent requirements in terms of reliability, responsiveness, and energy consumption. About two decades of joint efforts from academia and industry paved the way towards this accomplishment; nevertheless, much work is still required to meet with evolving communication demands, as these increasingly become stricter and more complex.

In this thesis, we proposed and evaluated novel techniques and protocols to enhance the dependability, latency, energy efficiency, and interference resilience of low-power wireless systems, pushing forward the applicability and performance of this technology. Our research endeavour has been mainly funnelled towards industrial wireless control, one of the most attractive applications of low-power wireless communications, which is gaining momentum within academia and has become a business priority for many organizations. We followed a system-centric approach: our research is informed, driven, and validated by experiments carried out in real-world testbeds, concretely demonstrating our results. We focused on two different PHY layers: IEEE 802.15.4 narrowband, the reference one in the industrial context, and ultra-wideband (UWB), which is rapidly reaching the forefront of IoT enabling technologies.

Leveraging concurrent transmissions (CTX), the latest and arguably one of the most promising advancements in low-power wireless networking, we put forth four main contributions. We started by analysing the ability of the popular Crystal system to support its target and challenging aperiodic, sparse traffic under strong radio-frequency noise, a precondition for its adoption in noise-prone environments like industrial settings, and devised techniques to further improve its interference resilience without impairing the ultra-low power consumption. Channel hopping and noise detection enable our enhanced version of Crystal to efficiently survive external interference by dynamically avoiding jammed frequencies and adapting the effort, and hence the energy consumption, put towards packet delivery depending on the noise conditions—an asset that can be similarly exploited by other CTX solutions. Our extensive evaluation demonstrates that these techniques, together, bring reliability to an unprecedented near-perfect level even under multiple sources of RF-noise, while preserving a per-mille radio duty cycle with aperiodic, sparse traffic. This performance was later confirmed in the last two editions of the EWSN Dependability Competition, where our modified Crystal system got the second and third place, respectively.

This achievement laid the ground for the exploitation of concurrent transmissions for aperiodic wireless control. The efficient, timely, and dependable support of non-predictable communications is indeed of paramount importance to unleash the full advantages provided by aperiodic sampling strategies, like event-triggered control, whose real-world impact has hitherto been hampered by the lack of appropriate networking layers. We filled this gap by co-designing the Wireless Control Bus (WCB), a full-fledged protocol stack leveraging carefully orchestrated floods of CTX to dynamically adapt the network operations to the ETC communication demands, unlocking and practically demonstrating for the first time the remarkable potential for energy savings that ETC entails.

We then directed our research attention towards UWB communications, a topic so far only marginally studied in the related literature. We provided reference implementations for UWB of Glossy, its TX-only variant, and Crystal—three representative solutions from the narrowband state of the art—analysing if, and to what extent, concurrent transmissions can be exploited for multi-hop UWB communication. The response is very positive: UWB CTX yield a reliability similar to narrowband both when the same and different packets are transmitted concurrently, whilst the higher UWB data rate and clock resolution enable substantial latency improvements, which become orders-of-magnitude ones for network-wide time synchronization. Furthermore, our study suggests that systems based on UWB CTX can achieve energy consumption akin, if not lower than, their narrowband counterparts, depending on the packet size. These results encourage the design and development of other CTX-based UWB protocols, which we foster by detailing the opportunities and challenges we faced while developing the above mentioned systems for the DecaWave DW1000 chip, serving as a guideline for other researches and practitioners.

With Weaver, our last contribution, we *i)* made an additional, significant step towards the widespread adoption of UWB technology for monitoring and control, providing a hitherto missing ultra-fast convergecast stack for UWB, and *ii)* proposed an alternate fine-grained approach to exploit concurrent transmissions, extending the perimeter of what CTX can achieve in low-power wireless far beyond what enabled by the Glossy system. Weaver design and performance are a witness to this. Leveraging individual CTX as primitive building block, Weaver disseminates towards a receiver different packets from multiple senders in a single, self-terminating network wide flood, abating collection latency and energy consumption w.r.t. state-of-the-art Glossy-based protocols, while achieving higher reliability.

Future directions. This thesis opens various research opportunities, briefly characterized in the following.

Our interference-resilient extension of Crystal, our first contribution in chronological order, already attracted considerable interest in the IEEE 802.15.4 community, with several works exploiting our protocol either as a term of comparison (e.g., [120, 122, 123, 173]), or as a stepping stone for novel developments [178, 179, 180].

The UWB research community can similarly benefit from the contributions presented in this dissertation, practically demonstrating for the first time the potential of UWB technology for communication, and providing readily-available implementations of CTX-based UWB stacks. A promising area where UWB communication protocols can make a difference is industrial wireless control. As exemplified in the second part of this thesis, UWB solutions yield unprecedented latency and synchronization performance, while minimising coexistence issues by operating outside the notoriously crowded and highly interfered 2.4 GHz ISM band; three key aspects that will tremendously benefit any wireless control system, especially aperiodic ones. Our immediate research plans include fostering the adoption of low-power wireless technology in the industrial context by co-designing novel control system architectures combining UWB CTX in synergy with aperiodic control strategies.

The individual CTX design paradigm presented in Chapter 7, likewise, opens new research perspectives in low-power wireless networking. By unlocking a fine-grained control of the nodes' operations, individual CTX raise the flexibility and expressiveness of protocol design, while preserving—or even enhancing—all the advantages concurrent transmissions provide. We foresee a clear potential in leveraging this asset for co-designing new WCSs, where a tighter synergy between the network and the control layer is exploited at run-time to improve the dependability and/or the energy efficiency of the overall system. In this respect, WCB paved the way towards a holistic WCS architecture that combines CTX with ETC. By taking advantage of all the degrees of freedom offered by individual CTX, we aim at extending the adaptability of our network stack to control demands—e.g., reducing the overhead and the rigidity that WCB's reliance on the Glossy flooding primitive entails—further enhancing the system performance and opening new co-design opportunities.

On a different path, the capillary control and accurate scheduling of radio operations provided by our TSM kernel unlock an additional, intriguing line of research; namely, to rejoin in a single protocol stack the communication and localization facets enabled by UWB radios. The intersection between UWB localization and networking research, hitherto largely unexplored in the literature, could finally unleash the full potential of UWB technology, fostering its adoption in several scenarios where we expect it to have an outbreak impact. For instance, mobile applications like coordination of robots and drones or swarm exploration could feature a single, cheap, energy efficient, and lightweight UWB chip both for precise localization and communication purposes, abating costs, weight, consumption, and complexity w.r.t. traditional solutions. Conventional or concurrent ranging techniques would enable devices to rapidly compute their relative positions, while CTX-based protocols would yield fast, dependable, and energy efficient multi-hop communications, e.g., to inform a coordinator about the current position of the nodes, disseminate new exploration commands, or schedule the operations of the localization layer.

We released all the techniques, protocols, and systems presented in this dissertation as open source, with the precise intent and hope to fuel these and many other opportunities of future work. Overall, we believe that this thesis already provides significant and novel stimuli to low-power wireless research, by offering concrete solutions to key problems hampering the use of low-power wireless networking stacks in industrial environments, and by showing the opportunities offered by recent technological developments.

Bibliography

[1] T. Istomin, M. Trobinger, A. L. Murphy, and G. P. Picco. "Interference-resilient Ultra-low Power Aperiodic Data Collection". In *Proc. of the* 17^{th} *ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN. 2018. DOI: 10.1109/IPSN.2018.00015.

[2] D. Lobba, M. Trobinger, D. Vecchia, T. Istomin, and G. P. Picco. "Concurrent Transmissions for Multi-hop Communication on Ultra-wideband Radios". In *Proc. of the* 17^{th} *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2020.

[3] M. Trobinger, T. Istomin, A. L. Murphy, and G. P. Picco. "Competition: CRYSTAL Clear: Making Interference Transparent". In *Proc. of the* 15^{th} *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2018.

[4] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. "Smart Dust: Communicating with a Cubic-millimeter Computer". In *Computer* 34.1 (2001), pp. 44–51.

[5] *Wireless Systems for Industrial Automation: Process Control and Related Applications, ISA-100.11a*. Standard. International Society of Automation, 2009.

[6] *HART Field Communication Protocol Specification*. Revision 7.0. HART Communication Foundation, 2007.

[7] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert. "IETF 6TiSCH: A Tutorial". In *IEEE Communications Surveys Tutorials* 22.1 (2020), pp. 595–615.

[8] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE. 2011.

[9] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli. "Low-Power Wireless for the Internet of Things: Standards and Applications". In *IEEE Access* 6 (2018), pp. 67893–67926.

[10] M. Zimmerling, L. Mottola, and S. Santini. "Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services". In *ACM Computing Surveys* 53.6 (2020).

[11] T. O'donovan et al. "The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences". In *ACM Transaction on Sensor Networks (TOSN)* 10.1 (2013).

[12] FieldComm Group. *WirelessHART User Case Studies*. Online: https://www.fieldcommgroup.org/technologies/hart/documents-and-downloads-hart/ [Accessed: 21-05-2021]. 2021.

[13] G. P. Picco, D. Molteni, A. L. Murphy, F. Ossi, F. Cagnacci, M. Corrà, and S. Nicoloso. "Geo-referenced Proximity Detection of Wildlife with WildScope: Design and Characterization". In *Proc. of the 14th International Conference on Information Processing in Sensor Networks (IPSN)*. 2015.

[14] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. T. Gună, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. "Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive lighting in Road Tunnels". In *Proc. of the International Conference on Information Processing in Sensor Networks (IPSN)*. 2011.

[15] Edward A. Lee. "Cyber Physical Systems: Design Challenges". In *Proc. of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. 2008.

[16] A. Biri, N. Jackson, L. Thiele, P. Pannuto, and P. Dutta. "SociTrack: Infrastructure-Free Interaction Tracking through Mobile Sensor Networks". In *Proc. of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. 2020.

[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2347–2376.

[18] 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.

[19] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, and S. Trimpe. "Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges (to appear)". In *Proceedings of the IEEE* (2020), pp. 1–27.

[20] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan. "A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks". In *IEEE Communications Surveys & Tutorials* 20.1 (2018), pp. 39–95.

[21] S. Jeschke, Christian Brecher, T. Meisen, D. Özdemir, and T. Eschert. "Industrial Internet of Things and Cyber Manufacturing Systems". In *Industrial Internet of TFhings*. Springer, 2017, pp. 3–19.

[22] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund. "Industrial Internet of Things: Challenges, Opportunities, and Directions". In *IEEE Transactions on Industrial Informatics* 14.11 (2018), pp. 4724–4734.

[23] S. Wang, J. Wan, D. Li, and C. Zhang. "Implementing Smart Factory of Industrie 4.0: An Outlook". In *International Journal of Distributed Sensor Networks* 12 (2016).

[24] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems". In *Proceedings of the IEEE* 104.5 (2016), pp. 1013–1024.

[25] J. Åkerberg, M. Gidlund, and M. Björkman. "Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation". In *Proc. of the 9th IEEE International Conference on Industrial Informatics*. 2011.

[26] N. Baccour, A. Koubâa, C. A. Boano, L. Mottola, H. Fotouhi, M. Alves, H. Youssef, M. A. Zúñiga, D. Puccinelli, T. Voigt, K. Römer, and C. Noda. *Radio Link Quality Estimation in Low-Power Wireless Networks*. SpringerBriefs in Electrical and Computer Engineering - Cooperating Objects. Springer International Publishing, 2013.

[27] A. Hithnawi, H. Shafagh, and S. Duquennoy. "Understanding the Impact of Cross Technology Interference on IEEE 802.15.4". In *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*. 2014.

[28] T. Chang, T. Watteyne, X. Vilajosana, and P. H. Gomes. "Constructive Interference in 802.15.4: A Tutorial". In *IEEE Communications Surveys Tutorials* 21.1 (2019), pp. 217–237.

[29] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. "Efficient Network Flooding and Time Synchronization with Glossy". In *Proc. of the International Conference on Information Processing in Sensor Networks*. IPSN. 2011.

[30] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. "Low-power Wireless Bus". In *Proc. of ACM Conference on Embedded Network Sensor Systems*. SenSys. 2012.

[31] T. Istomin, A. L. Murphy, G. P. Picco, and U. Raza. "Data Prediction + Synchronous Transmissions = Ultra-low Power Wireless Sensor Networks". In *Proc. of the ACM Conference on Embedded Network Sensor Systems*. SenSys. 2016.

[32] O. Landsiedel, F. Ferrari, and M. Zimmerling. "Chaos: Versatile and Efficient All-to-all Data Sharing and In-network Processing at Scale". In *Proc. of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2013.

[33] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme. "The Industrial Indoor Channel: Large-Scale and Temporal Fading at 900, 2400, and 5200 MHz". In *IEEE Transaction on Wireless Communications* 7.7 (2008), pp. 2740–2751.

[34] M. Paavola and K. Leiviska. "Wireless Sensor Networks in Industrial Automation". In *Factory automation*. IntechOpen, 2010.

[35] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga. "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation". In *Proc. of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2011.

[36] M. Schuß, C. A. Boano, M. Weber, M. Schulz, M. Hollick, and K. Römer. "JamLab-NG: Benchmarking Low-Power Wireless Protocols under Controllable and Repeatable Wi-Fi Interference". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks*. EWSN '19. 2019.

[37] F. Foukalas, P. Pop, F. Theoleyre, C. A. Boano, and C. Buratti. “Dependable Wireless Industrial IoT Networks: Recent Advances and Open Challenges”. In *Proc. of the 2019 IEEE European Test Symposium (ETS)*. 2019.

[38] Karl Johan Åström and Bo Bernhardsson. “Comparison of Periodic and Event Based Sampling for First-order Stochastic Systems”. In *IFAC Proceedings Volumes* 32.2 (1999), pp. 5006–5011.

[39] Karl-Erik Åarzén. “A simple Event-based PID Controller”. In *IFAC Proceedings Volumes* 32.2 (1999), pp. 8687–8692.

[40] W. Heemels, K. H. Johansson, and P. Tabuada. “An Introduction to Event-triggered and Self-triggered Control”. In *Proc. of the IEEE Conference on Decision and Control (CDC)*. 2012.

[41] M. Schuß, C. A. Boano, M. Weber, and K. Römer. “A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge”. In *Proc. of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2017.

[42] M. Trobinger, T. Istomin, A. L. Murphy, and G. P. Picco. “Competition: CRYSTAL”. In *Proc. of the* 16^{th} *International Conference on Embedded Wireless Systems and Networks*. EWSN. 2019.

[43] Paulo Tabuada. “Event-triggered Real-time Scheduling of Stabilizing Control Tasks”. In *IEEE Transactions on Automatic Control* 52.9 (2007), pp. 1680–1685.

[44] M. Mazo Jr. and P. Tabuada. “Decentralized Event-triggered Control over Wireless Sensor/Actuator Networks”. In *IEEE Transactions on Automatic Control* 56.10 (2011), pp. 2456–2461.

[45] W. Heemels, M. Donkers, and A. Teel. “Periodic Event-triggered Control for Linear Systems”. In *IEEE Transactions on Automatic Control* 58.4 (2013), pp. 847–861.

[46] M. Miskowicz. *Event-based Control and Signal Processing*. CRC Press, 2018.

[47] J. Araújo, M. Mazo Jr., A. Anta, P. Tabuada, and K. H. Johansson. “System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems”. In *IEEE Transactions on Industrial Informatics* 10.1 (2014), pp. 175–184.

[48] S. Kartakis, A. Fu, M. Mazo Jr., and J. A. McCann. “Communication Schemes for Centralized and Decentralized Event-Triggered Control Systems”. In *IEEE Transactions on Control Systems Technology* 26.6 (2018), pp. 2035–2048.

[49] Apple. *iPhone11*. Online: `https://support.apple.com/en-gb/guide/security/sec1e6108efd/web` [Accessed: 12-04-2021]. 2021.

[50] Samsung Newsroom. *Samsung Galaxy S21 Ultra*. Online: `https://news.samsung.com/us/samsung-galaxy-s21-ultra-unpacked-2021-ultimate-smartphone-experience-designed-epic/` [Accessed: 12-04-2021]. 2021.

[51] Samsung Newsroom. *Samsung SmartTag+*. Online: `https://news.samsung.com/us/introducing-the-new-galaxy-smarttag-plus/` [Accessed: 12-04-2021]. 2021.

[52] F. Zafari, A. Gkelias, and K. K. Leung. "A Survey of Indoor Localization Systems and Technologies". In *IEEE Communications Surveys Tutorials* 21.3 (2019), pp. 2568–2599.

[53] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa. "Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances". In *Sensors* 16.5 (2016).

[54] DecaWave Ltd. *DW1000 Data Sheet, Version 2.19*. 2017.

[55] DecaWave Ltd. *DW1000 User Manual, Version 2.18*. 2017.

[56] Advanticsys. *MTM-CM5000-MSP 802.15.4 TelosB mote Module*. Online: `https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html` [Accessed: 09-05-2021]. 2021.

[57] Zolertia Inc. *Zolertia Firefly*. Online: `https://zolertia.io/product/firefly` [Accessed: 18-12-2020]. 2020.

[58] DecaWave Ltd. *DecaWave ScenSor EVB1000 Evaluation Board*. 2013.

[59] `https://github.com/d3s-trento/crystal`.

[60] `https://github.com/d3s-trento/contiki-uwb`.

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

[62] J. Klaue, A. Corona, M. Kubisch, J. Garcia-Jimenez, and A. Escobar-Molero. "Competition: RedFixHop". In *Proc. of the 13th International Conference on Embedded Wireless Systems and Networks (EWNS)*. 2016.

[63] R. Lim, R. Da Forno, F. Sutton, and L. Thiele. "Competition: Robust Flooding Using Back-to-Back Synchronous Transmissions with Channel-Hopping". In *Proc. of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2017.

[64] P. Sommer and Y-A Pignolet. "Competition: Dependable Network Flooding Using Glossy with Channel-Hopping". In *Proc. of the 13th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2016.

[65] A. Escobar, F. Moreno, A. J. Cabrera, J. Garcia-Jimenez, F. J. Cruzand U. Ruiz, J. Klaue, A. Corona, D. Tati, and T. Meyerhoff. "Competition: BigBangBus". In *Proc. of the 15th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2018.

[66] X. Ma, P. Zhang, Y. Liu, X. Li, W. Tang, P. Tian, J. Wei, L. Shu, and O. Theel. "Competition: Using DeCoT+ to Collect Data under Interference". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2019.

[67] M. Baddeley, A. Stanoev, U. Raza, M. Sooriyabandara, and Y. Jin. "Competition: Adaptive Software Defined Scheduling of Low Power Wireless Networks". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2019.

[68] A. Escobar-Molero, J. Garcia-Jimenez, J. Klaue, F. Moreno-Cruz, B. Saez, F. J. Cruz, U. Ruiz, and A. Corona. "Competition: RedNodeBus, Stretching out the Preamble". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2019.

[69] M. König and R. Wattenhofer. "Maintaining Constructive Interference Using Well-Synchronized Sensor Nodes". In *Proc. of the 12th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2016.

[70] C. Liao, Y. Katsumata, M. Suzuki, and H. Morikawa. "Revisiting the So-Called Constructive Interference in Concurrent Transmission". In *IEEE 41st Conference on Local Computer Networks (LCN)*. 2016.

[71] M. Wilhelm, V. Lenders, and J. B. Schmitt. "On the Reception of Concurrent Transmissions in Wireless Sensor Networks". In *IEEE Transactions on Wireless Communications* 13.12 (2014), pp. 6756–6767.

[72] M. Baddeley, C. A. Boano, A. Escobar-Molero, Y. Liu, X. Ma, U. Raza, K. Römer, M. Schuß, and A. Stanoev. "The Impact of the Physical Layer on the Performance of Concurrent Transmissions". In *IEEE 28th International Conference on Network Protocols (ICNP)*. 2020.

[73] V. S. Rao, M. Koppal, R. Venkatesha Prasad, T.V. Prabhakar, C. Sarkar, and I. Niemegeers. "Murphy Loves CI: Unfolding and Improving Constructive Interference in WSNs". In *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*. 2016.

[74] Antonio Escobar-Molero. "Improving Reliability and Latency of Wireless Sensor Networks using Concurrent Transmissions". In *at - Automatisierungstechnik* 67 (2019), pp. 42–50.

[75] M. Trobinger, G. A. Gleizer, T. Istomin, M. Mazo Jr., A. M. Murphy, and G. P. Picco. "The Wireless Control Bus: Enabling Efficient Multi-hop Event-Triggered Control with Concurrent Transmissions". In (2021). arXiv: `2101.10961 [eess.SY]`.

[76] Michael Baddeley. "Software Defined Networking for the Industrial Internet of Things". Ph.D. Dissertation. Dept. of Electrical and Electronic Engineering, University Bristol, UK, 2020.

[77] B. Al Nahas, S. Duquennoy, and O. Landsiedel. "Concurrent Transmissions for Multi-Hop Bluetooth 5". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2019.

[78] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso. "Do LoRa Low-Power Wide-Area Networks Scale?" In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '16)*. 2016.

[79] C. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa. "Multi-Hop LoRa Networks Enabled by Concurrent Transmission". In *IEEE Access* 5 (2017), pp. 21430–21446.

[80] J. Lee, W. Kim, S. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. "An Experimental Study on the Capture Effect in 802.11a Networks". In *Proc. of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH '07)*. 2007.

[81] D. Vecchia, P. Corbalan, T. Istomin, and G. P. Picco. "Playing with Fire: Exploring Concurrent Transmissions in Ultra-wideband Radios". In *Proc. of the 18th IEEE International Conference on Sensing, Communication and Networking (SECON)*. 2019.

[82] C. Gezer, C. Buratti, and R. Verdone. "Capture Effect in IEEE 802.15.4 Networks: Modelling and Experimentation". In *Proceedings of the 5th IEEE International Conference on Wireless Pervasive Computing (ISWPC)*. 2010.

[83] J. Lu and K. Whitehouse. "Flash Flooding: Exploiting the Capture Effect for Rapid Flooding in Wireless Sensor Networks". In *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*. 2009.

[84] B. Al Nahas, S. Duquennoy, and O. Landsiedel. "Network-wide Consensus Utilizing the Capture Effect in Low-power Wireless Networks". In *Proc. of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys)*. 2017.

[85] C. Herrmann, F. Mager, and M. Zimmerling. "Mixer: Efficient Many-to-all Broadcast in Dynamic Wireless Mesh Networks". In *Proc. of SenSys*. 2018.

[86] P. Dutta, Răzvan Musăloiu-e, I. Stoica, and A. Terzis. "Wireless ACK Collisions Not Considered Harmful". In *In Proc. of the 7th ACM Workshop on Hot Topics in Networks (HotNets)*. 2008.

[87] P. Dutta, S. Dawson-Haggerty, Y. Chen, C-J. M. Liang, and A. Terzis. "A-MAC: A Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless". In *ACM Transaction on Sensor Networks (TOSN)* 8.4 (2012).

[88] M. Zimmerling, F. Ferrari, L. Mottola, and L. Thiele. "On Modeling Low-Power Wireless Protocols Based on Synchronous Packet Transmissions". In *Proc. of International Symp. on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 2013.

[89] K. Khakpour and M. H. Shenassa. "Industrial Control using Wireless Sensor Networks". In *Proc. of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications*. 2008.

[90] A. Ahlén et al. "Toward Wireless Control in Industrial Process Automation: A Case Study at a Paper Mill". In *IEEE Control Systems Magazine* 39.5 (2019), pp. 36–57.

[91] Y. Ma and C. Lu. "Efficient Holistic Control over Industrial Wireless Sensor-Actuator Networks". In *Proc. of IEEE International Conference on Industrial Internet (ICII)*. 2018.

[92] D. Baumann, F. Mager, R. Jacob, L. Thiele, M. Zimmerling, and S. Trimpe. "Fast Feedback Control over Multi-Hop Wireless Networks with Mode Changes and Stability Guarantees". In *ACM Transaction on Cyber-Physical Systems* 4.2 (2019).

[93] D. Gunatilaka and C. Lu. “REACT: an Agile Control Plane for Industrial Wireless Sensor-Actuator Networks”. In *Proc. of the Fifth IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 2020.

[94] Y. Ma, D. Gunatilaka, B. Li, H. Gonzalez, and C. Lu. “Holistic Cyber-Physical Management for Dependable Wireless Control Systems”. In *ACM Transaction on Cyber-Physical Systems* 3.1 (2018).

[95] 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 Tutorials* 16.3 (2014), pp. 1391–1412.

[96] D. Yuan, M. Riecker, and M. Hollick. “Making ’Glossy’ Networks Sparkle: Exploiting Concurrent Transmissions for Energy Efficient, Reliable, Ultra-Low Latency Communication in Wireless Control Networks”. In *Proc. of the 11th European Conference on Wireless Sensor Networks (EWSN)*. 2014.

[97] S. Duquennoy, O. Landsiedel, and T. Voigt. “Let the Tree Bloom: Scalable Opportunistic Routing with ORPL”. In *Proc. of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2013.

[98] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt. “Low-Power Listening Goes Multi-Channel”. In *Proc. of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems*. DCOSS ’14. 2014.

[99] M. Mohammad, X. Guo, and M. C. Chan. “Oppcast: Exploiting Spatial and Channel Diversity for Robust Data Collection in Urban Environments”. In *Proc. of the 15th International Conference on Information Processing in Sensor Networks (IPSN)*. 2016.

[100] P. Zhang, O. Landsiedel, and O. Theel. “MOR: Multichannel Opportunistic Routing for Wireless Sensor Networks”. In *Proc. of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2017.

[101] B. Al Nahas and O. Landsiedel. “Competition: Towards Low-Power Wireless Networking that Survives Interference with Minimal Latency”. In *Proc. of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2017.

[102] M. Suzuki, Y. Yamashita, and H. Morikawa. “Low-Power, End-to-End Reliable Collection Using Glossy for Wireless Sensor Networks”. In *Proc. of the 77th IEEE Vehicular Technology Conference (VTC Spring)*. 2013.

[103] N.Q.V. Hung, H. Jeung, and K. Aberer. “An Evaluation of Model-Based Approaches to Sensor Data Compression”. In *IEEE Transaction on Knowledge and Data Engineering (TKDE)* 25.11 (2013).

[104] U. Raza, A. Camerra, A.L. Murphy, T. Palpanas, and G.P. Picco. “Practical Data Prediction for Real-World Wireless Sensor Networks”. In *IEEE Transaction on Knowledge and Data Engineering (TKDE)* 27.8 (2015), pp. 2231–2244.

[105] Intel Research Lab. `http://db.csail.mit.edu/labdata/labdata.html`.

[106] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda. “Indriya: A low-cost, 3D wireless sensor network testbed”. In *Proc. of the International Conference on Testbeds and Research Infrastructures (TridentCom)*. 2011.

[107] A. Dunkels, B. Gronvall, and T. Voigt. “Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors”. In *Proc. of the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*. 2004.

[108] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. “Collection Tree Protocol”. In *Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2009.

[109] Adam Dunkels. *The ContikiMAC Radio Duty Cycling Protocol*. Technical Report T2011:13. SICS, 2011.

[110] Texas Instruments. *CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Online: `http://www.ti.com/product/CC2420/technicaldocuments[Accessed: 05-04-2021]`. 2013.

[111] D. Han and O. Gnawali. “Performance of RPL Under Wireless Interference”. In *IEEE Communications Magazine* 51.12 (2013), pp. 137–143.

[112] T. Watteyne, A. Mehta, and K. Pister. “Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense”. In *Proc. of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. PE-WASUN ’09. 2009.

[113] A. Gonga, O. Landsiedel, P. Soldati, and M. Johansson. “Revisiting Multichannel Communication to Mitigate Interference and Link Dynamics in Wireless Sensor Networks”. In *Proc. of the 8th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2012.

[114] A Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou. “A Survey on Jamming Attacks and Countermeasures in WSNs”. In *IEEE Communications Surveys Tutorials* 11.4 (2009), pp. 42–56.

[115] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. “FlockLab: A Testbed for Distributed, Synchronized Tracing and Profiling of Wireless Embedded Systems”. In *Proc. of the 12th International Conference on Information Processing in Sensor Networks (IPSN)*. 2013.

[116] *IEEE802.15.4-2015: IEEE Standard for Low-Rate Wireless Networks*. Standard. W-PAN Working Group, 2015.

[117] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne. “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH”. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2015.

[118] M. Doddavenkatappa, M. C. Chan, and B. Leong. “Splash: Fast Data Dissemination with Constructive Interference in Wireless Sensor Networks”. In *Proc. of the 10th USENIX Conference on Networked Systems Design and Implementation*. 2013.

[119] W. Du, J. C. Liando, H. Zhang, and M. Li. "When Pipelines Meet Fountain: Fast Data Dissemination in Wireless Sensor Networks". In *Proc. of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2015.

[120] X. Ma, P. Zhang, X. Li, W. Tang, J. Wei, and O. Theel. "DeCoT: A Dependable Concurrent Transmission-Based Protocol for Wireless Sensor Networks". In *IEEE Access* 6 (2018), pp. 73130–73146.

[121] M. Baddeley, U. Raza, A. Stanoev, G. Oikonomou, R. Nejabati, M. Sooriyabandara, and D. Simeonidou. "Atomic-SDN: Is Synchronous Flooding the Solution to Software-Defined Networking in IoT?" In *IEEE Access* 7 (2019), pp. 96019–96034.

[122] X. Ma, P. Zhang, Y. Liu, C. A. Boano, H. S. Kim, J. Wei, and J. Huang. "Harmony: Saving Concurrent Transmissions from Harsh RF Interference". In *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*. 2020.

[123] V. Poirot and O. Landsiedel. "Dimmer: Self-Adaptive Network-Wide Flooding with Reinforcement Learning". In *Proc. of the International Conference on Distributed Computing Systems (ICDCS), to appear*. 2021.

[124] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling. "Feedback Control Goes Wireless: Guaranteed Stability over Low-power Multihop Networks". In *Proc. of the International Conference on Cyber-Physical Systems*. ICCPS. 2019.

[125] K. J. Åström and B. Bernhardsson. "Comparison of Riemann and Lebesgue Sampling for First Order Stochastic Systems". In *Proc. of the 41st IEEE Conference on Decision and Control (CDC)*. 2002.

[126] Hassan K. Khalil. *Nonlinear Systems*. 3rd ed. Pearson, 2001.

[127] X. Wang and M. D. Lemmon. "Event Design in Event-triggered Feedback Control Systems". In *Proc. of the 47th IEEE Conference on Decision and Control (CDC)*. 2008.

[128] Antoine Girard. "Dynamic Triggering Mechanisms for Event-triggered Control". In *IEEE Transactions on Automatic Control* 60.7 (2014), pp. 1992–1997.

[129] B. van Eekelen, N. Rao, B. A. Khashooei, D. J. Antunes, and W. P. M. H. Heemels. "Experimental Validation of an Event-triggered Policy for Remote Sensing and Control with Performance Guarantees". In *Proc. of the Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*. 2016.

[130] C. Ramesh, H. Sandberg, and K. H. Johansson. "Design of State-Based Schedulers for a Network of Control Loops". In *IEEE Transactions on Automatic Control* 58.8 (2013), pp. 1962–1975.

[131] V. Dolk, J. Ploeg, and W. Heemels. "Event-Triggered Control for String-Stable Vehicle Platooning". In *IEEE Transactions on Intelligent Transportation Systems* 18.12 (2017), pp. 3486–3500.

[132] B. Al Nahas, A. Escobar-Molero, J. Klaue, S. Duquennoy, and O. Landsiedel. "BlueFlood: Concurrent Transmissions for Multi-Hop Bluetooth 5 — Modeling and Evaluation (to appear)". In *ACM Transactions on Internet of Things (TIOT)* (2021).

[133] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta. "SurePoint: Exploiting Ultra Wideband Flooding and Diversity to Provide Robust, Scalable, High-Fidelity Indoor Localization". In *Proc. of the 14th ACM Conference on Embedded Network Sensor Systems (SenSys)*. 2016.

[134] M. Trobinger, D. Vecchia, D. Lobba, T. Istomin, and G.P. Picco. "One Flood to Route Them All: Ultra-fast Convergecast of Concurrent Flows over UWB". In *Proc. of the* 18^{th} *ACM Conference on Embedded Networked Sensor Systems*. SenSys. 2020. DOI: `10.1145/3384419.3430715`.

[135] D. Baumann, F. Mager, M. Zimmerling, and S. Trimpe. "Control-Guided Communication: Efficient Resource Arbitration and Allocation in Multi-Hop Wireless Control Systems". In *IEEE Control Systems Letters* 4.1 (2020), pp. 127–132.

[136] Y. Ma, C. Lu, and Y. Wang. "Efficient Holistic Control: Self-Awareness across Controllers and Wireless Networks". In *ACM Transaction in Cyber-Physical Systems* 4.4 (2020).

[137] M. Cantoni, E. Weyerand Y. Li, S. K. Ooi, I. Mareels, and M. Ryan. "Control of Large-scale Irrigation Networks". In *Proceedings of the IEEE* 95.1 (2007), pp. 75–91.

[138] J. Malmborg and J. Eker. "Hybrid Control of a Double Tank System". In *Proc. of International Conference on Control Applications*. 1997.

[139] J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson. "Self-triggered Control over Wireless Sensor and Actuator Networks". In *Proc. of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. 2011.

[140] Y. Li and B. de Schutter. "Stability and Performance Analysis of an Irrigation Channel with Distributed Control". In *Control Engineering Practice* 19.10 (2011), pp. 1147–1156.

[141] Y. Li and M. Cantoni. "Distributed Controller Design for Open Water Channels". In *IFAC Proceedings Volumes* 41.2 (2008), pp. 10033–10038.

[142] Rubicon Water. *FlumeGate – Flow control and Measurement Gate*. [Accessed 25-11-2019]. 2019. URL: `https://www.rubiconwater.com/catalogue/flumegate` (visited on 11/25/2019).

[143] Erik Weyer. "System Identification of an Open Water Channel". In *Control engineering practice* 9.12 (2001), pp. 1289–1299.

[144] K. C. Hewage, S. Raza, and T. Voigt. "Protecting Glossy-Based Wireless Networks from Packet Injection Attacks". In *Proc. of the International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2017.

[145] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe. "Evaluating Low-Power Wireless Cyber-Physical Systems". In *Proc. of the IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*. 2018.

[146] M. Zimmerling, L. Mottola, P. Kumar, F. Ferrari, and L. Thiele. "Adaptive Real-Time Communication for Wireless Cyber-Physical Systems". In *ACM Transaction on Cyber-Physical Systems* 1.2 (2017).

[147] F. Sutton, R. Da Forno, D. Gschwend, T. Gsell, R. Lim, J. Beutel, and L. Thiele. "The Design of a Responsive and Energy-Efficient Event-Triggered Wireless Sensing System". In *Proc. of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2017.

[148] A. S. Kolarijani and M. Mazo Jr. "A Formal Traffic Characterization of LTI Event-triggered Control Systems". In *IEEE Transactions on Control of Network Systems* 5.1 (2016), pp. 274–283.

[149] G. A. Gleizer and M. Mazo Jr. "Scalable Traffic Models for Scheduling of Linear Periodic Event-Triggered Controllers". In *21st IFAC World Congress (accepted)* (2020). `https://arxiv.org/abs/2003.07642`.

[150] IEEE. *IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN)*. Version IEEE 802.15.4-2003. 2003.

[151] Time Domain. *Time Domain PulsON 400 RCM Data Sheet*. 2011.

[152] *UWB Alliance*. Online: `https://uwballiance.org` [Accessed: 05-04-2021]. 2021.

[153] *FiRa Consortium*. Online: `https://www.firaconsortium.org` [Accessed: 05-04-2021]. 2021.

[154] Nitin Dahad (embedded). *Lenovo Adds Novelda UWB Sensor for Human Presence Detection in ThinkPad*. Online: `https://www.embedded.com/lenovo-adds-novelda-uwb-sensor-for-human-presence-detection-in-thinkpad/` [Accessed: 12-04-2021]. 2021.

[155] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco. "TALLA: Large-scale TDoA Localization with Ultra-wideband Radios". In *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2019.

[156] J. F. Schmidt, D. Neuhold, C. Bettstetter, J. Klaue, and D. Schupke. "Wireless Connectivity in Airplanes: Challenges and the Case for UWB". In *IEEE Access* (2021).

[157] J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney. "UWB Systems for Wireless Sensor Networks". In *Proceedings of the IEEE* 97.2 (2009), pp. 313–331.

[158] J. F. Schmidt, D. Neuhold, J. Klaue, D. Schupke, and C. Bettstetter. "Experimental Study of UWB Connectivity in Industrial Environments". In *Proc. of the 24th European Wireless Conference*. 2018.

[159] J. F. Schmidt, D. Chernov, and C. Bettstetter. "Towards Industrial Ultra-Wideband Networks: Experiments for Machine Vibration Monitoring". In *IEEE Access* 8 (2020), pp. 42576–42583.

[160] D. Neuhold, J. F. Schmidt, J. Klaue, D. Schupke, and C. Bettstetter. "Experimental Study of Packet Loss in a UWB Sensor Network for Aircraft". In *Proc. of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. 2017.

[161] L. Yang and G. B. Giannakis. "Ultra-wideband Communications: An Idea Whose Time Has Come". In *IEEE Signal Processing Magazine* 21.6 (2004), pp. 26–54.

[162] Texas Instruments. *CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee Applications*. 2015.

[163] P. Corbalán, T. Istomin, and G. P. Picco. "Poster: Enabling Contiki on Ultra-Wideband Radios". In *Proc. of the 15th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2018.

[164] M. Mohammad and M. C. Chan. "Codecast: Supporting Data Driven in-Network Processing for Low-Power Wireless Sensor Networks". In *Proc. of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2018.

[165] P. Corbalán and G. P. Picco. "Concurrent Ranging in Ultra-Wideband Radios: Experimental Evidence, Challenges, and Opportunities". In *Proc. of the 2018 International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2018.

[166] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer. "Concurrent Ranging with Ultra-Wideband Radios: From Experimental Evidence to a Practical Solution". In *Proc. of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 2018. DOI: `10.1109/ICDCS.2018.00149`.

[167] Texas Instruments. *MSP430F1611 datasheet*. Online: `https://www.ti.com/product/MSP430F1611[Accessed:20-04-2021]`. 2009.

[168] M. Mohammad, M. Doddavenkatappa, and M. C. Chan. "Improving Performance of Synchronous Transmission-Based Protocols Using Capture Effect over Multichannels". In *ACM Transaction on Sensor Networks (TOSN)* 13.2 (2017).

[169] SPARK. *SPARK SR1000 UWB Wireless Transceiver Family*. Online: `https://www.sparkmicro.com/products/` [Accessed: 21-04-2021]. 2021.

[170] DecaWave Ltd. (Qorvo). *DW3000 Data Sheet, Version 1.1*. 2020.

[171] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer. "Enabling Runtime Adaptation of Physical Layer Settings for Dependable UWB Communications". In *Proc. of the IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 2018. DOI: `10.1109/WoWMoM.2018.8449776`.

[172] M. Brachmann, O. Landsiedel, and S. Santini. "Concurrent Transmissions for Communication Protocols in the Internet of Things". In *Proc. of the 41st IEEE Conference on Local Computer Networks (LCN)*. 2016.

[173] R. Jacob, J. Baechli, R. Da Forno, and L. Thiele. "Synchronous Transmissions Made Easy: Design Your Network Stack with Baloo". In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN)*. 2019.

[174] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. "Software-Based on-Line Energy Estimation for Sensor Nodes". In *Proc. of the 4th Workshop on Embedded Networked Sensors (EmNets)*. 2007.

[175] P. Corbalán, G. P. Picco, and S. Palipana. "Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets". In *Proc. of the 18th International Conference on Information Processing in Sensor Networks (IPSN)*. 2019.

[176] B. Großwindhager, M. Stocker, M. Rath, C. A. Boano, and K. Römer. "SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags". In *Proc. of the 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2019.

[177] T. Wang, H. Zhao, and Y. Shen. "An Efficient Single-Anchor Localization Method Using Ultra-Wide Bandwidth Systems". In *Applied Sciences* (2020).

[178] C. Stylianopoulos, M. Almgren, O. Landsiedel, and M. Papatriantafilou. "Continuous Monitoring meets Synchronous Transmissions and In-Network Aggregation". In *Proc.of the 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2019.

[179] J. Mueller, A-B. Schaper, R. Jacob, and R. Da Forno. "Competition: Keep It Simple, Let Flooding Shine." In *Proc. of the 16th International Conference on Embedded Wireless Systems and Networks*. EWSN '19. 2019.

[180] M. Brachmann, O. Landsiedel, D. Göhringer, and S. Santini. "Whisper: Fast Flooding for Low-Power Wireless Networks". In *ACM Transaction on Sensor Networks (TOSN)* 15.4 (2019).